Time-Varying Data Visualization Using Clustered Heatmap and Dual Scatterplots

Satsuki Kumatani[1], Takayuki Itoh[1], Yousuke Motohashi[2], Keisuke Umezu[2], Masahiro Takatsuka[3]
[1]Ochanomizu University, [2]NEC Corporation, [3]The University of Sydney
{[1]satsuki@itolab.is.ocha.ac, [1]itot@is.ocha.ac.jp, [2]y-motohashi@bk.jp.nec.com,
[2]k-umezu@ak.jp.nec.com, masa.takatsuka@sydoneyu.edu.au}

Abstract

Heatmap is one of the effective representations for time-varying data visualization. It may require large display spaces when an input dataset contains large number of data items or time steps. 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. We can intuitively understand how clusters of data items or time steps are constructed, by looking at the scatterplots while operating the sliders.

Keywords--- **Time-varying data visualization, Heatmap, Scatterplot, Clustering**.

1. Introduction

Recent information technology has brought huge and complex time-varying data. Visualization techniques for time-varying data are useful to subjectively and interactively explore such data and discover fruitful knowledge. For example, business management data can be treated as time-varying data containing multiple numeric values, such as revenue and/or profit of multiple products or services. Relationships and correlations among such time-varying values are keys to discover important knowledge: for example, it is fruitful to find particular time periods which particular sets of values well correlate each other. This kind of knowledge can be often hints to improve the business of companies. Time-varying data visualization is effective to discover such relationships and correlations in the datasets.

Polyline chart and heatmap are the most popular approaches to represent the time-varying data. This paper focuses on heatmap-based representation of time-varying data. Here, we suppose a two dimensional orthogonal coordinate system where time is assigned to the horizontal axis and data items (corresponding to the variables) are arranged along the vertical axis. The heatmap-based visualization technique divides a display space into a grid along the coordinate system, and paints the small rectangles in the grid according to the values at each time step. A rectangle corresponding to one value at a particular time step would be very small when the number of variables and time steps are very large, while the display space is limited; this problem may be often a bottleneck to visually recognize the trend of the time-varying datasets. Therefore, it is also often effective to filter non-important data items or noisy time steps to adjust the size of displayed portion of the datasets and focus on observation of well-correlated groups of data items. Interactive mechanisms to filter non-important portions of the datasets are effective technical components to realize effective heatmap-based time-varying data visualization.

This paper presents a heatmap-based time-varying data visualization technique featuring a user interface to easily filter non-important data items and time steps which corresponds to an operation to control the sizes of displayed portions of the datasets. The technique calculates distances between arbitrary pairs of data items treating them as multidimensional vectors, and constructs a dendrogram consisting all the data items. Similarly, it calculates distances between arbitrary pairs of time steps treating them as multidimensional vectors. The technique applies a clustering algorithm to the data items to identify groups of well-related data items, and also, to the time steps to filter the outlier time steps. Finally, it displays the identified data items and time steps as a heatmap. Users can interactively control the thresholds of the two clustering processes to obtain the comprehensive size of the heatmap.

Figure 1 shows a window capture of our implementation. The left side of the drawing widget displays the heatmap. The right side displays two scatterplots: one represents the distribution of distances

among the data items, and the other represents the distribution of distances among the time steps. Two sliders indicated by a red circle are featured to interactively control the thresholds of the clustering processes. Users can intuitively understand how clusters are generated, by operating the sliders and observing the two scatterplots.

Figure 1: Snapshot of the presented technique. In addition to displaying a heatmap in the left side of the drawing region, it displays two scatterplots to represent the distribution of distances among data items and time steps respectively. The technique applies clustering algorithms to data items and time steps so that users can filter non-important data items and time steps. Users can intuitively observe how clusters are generated by looking at the scatterplots. Thresholds of the clustering processes can be controlled by using two slider widgets indicated by a red circle.

2. Related work

2.1. Heatmap-based Time-Varying Data Visualization

Time-varying data visualization is one of the active topics in the visualization community. Polyline chart is the most common representation for time-varying data in our daily life. However, polyline-based representation has drawbacks including.

- Cluttering among large number of polylines in a single display space.
- Less display space utilization when the numeric distribution is unbalanced.

While several improved polyline-based techniques [6] are developed to solve the above problems, other representations including heatmaps have been applied to many time-varying data visualization techniques.

Heatmap-based representation has been applied to various time-varying data visualization techniques, as well as the technique presented in this paper. Heatmap has advantages over other representations from the standpoint of cluttering reduction and display space utilization for overviews. Imoto et al. presented a technique that extracts interesting portions of time-varying data on a heatmap [3]. Ziegler et al. [9] also presented a heatmap-based technique applying Pixel Bar Charts. These techniques did not explicitly group similar data items. Suematsu et al. [4] presented a heatmap-based time-varying multi-variate data visualization technique applying a non-hierarchical clustering to the mixture of numeric and categorical variables. However, the technique does not support interactive mechanism to adjust the clustering results.

Heatmap-based time-varying data visualization is also useful for development of visual analytics tools. WireVis [1] is a coordinated-view system featuring heatmap, polyline charts, pie charts, and search result display widgets. Hayashi et al. [2] presented a similar system featuring a variable recommendation algorithm so that users can easily find interesting trends in particular variables.

2.2. Dimension Analysis with Scatterplots

Dimension analysis for multi-dimensional datasets has been an active issue for development of visualization techniques and quite relevant to the visualization technique presented in this paper. There have been many popular multi-dimensional data visualization techniques including parallel coordinate plots (PCP) and scatterplots. When a multi-dimensional dataset contains a very large number of dimensions, such existing visualization techniques may need very large display spaces to represent them completely. This problem can be solved by dividing the high-dimensional data space into smaller subsets. To interactively and intuitively determine how to select the smaller subsets of dimensions, several recent studies have applied scatterplots for the representation of dimension spaces in which each dot in the scatterplot represents a one dimension in the space. Turkay et al. [5] presented a dual scatterplot model to visualize both the items and dimensions spaces. Similarly, Yuan et al. [7] presented an interactive mechanism to select low-dimensional subspaces on the scatterplot display in which each dot corresponds to a different dimension. Another technique recently proposed by Zhang et al. [8] displays a graph for representation of dimension relationships. The presented technique also applies a similar representation; however, it differs from these existing techniques since the technique features two scatterplots. One represents relationship among data items, while the other represents relationship among time steps.

3. PRESENTED VISUALIZATION TOOL

3.1. Data Definition and Processing Flow

We formalize the time-varying datasets visualized by our technique as follows. The dataset has m items, which contain n time steps of real values. The dataset D and the i-th item a_i are described as:

$$D = \{a_1, ..., a_m\}$$
$$a_i = \{v_{i1}, v_{i2}, ..., v_{in}\}$$

where v_{ij} denotes the real value at the j-th time step of the i-th item.

We treat the real values of the items as vectors, and calculates the distances between arbitrary pairs of items. At the same time, we also treat the real values of all items at a particular time step as a vector, defined as following:

$$T = \{t_1, ..., t_n\}$$
$$t_i = \{v_{1i}, v_{2i}, ..., v_{mi}\}$$

Figure 2(1) illustrates the data structure.

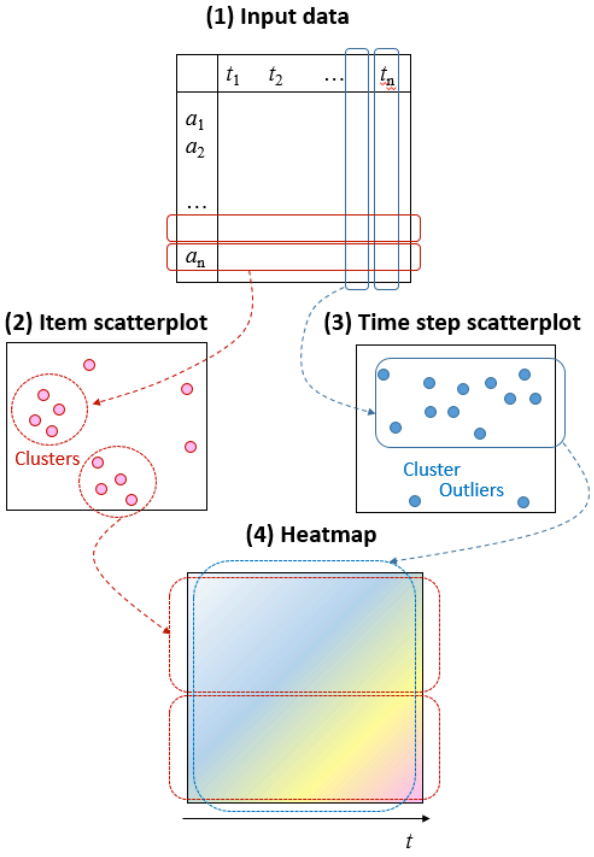


Figure 2: Processing flow of the presented visualization technique.

The technique then calculates the following distances:

- $dsta_{ij}$: distances between all possible pairs of items a_i *and* a_j.
- $dstt_{ij}$: distances between all possible pairs of time step t_i and t_j.

We display two scatterplots which represents the distance distributions of the set of items D and the set of time steps T respectively, as illustrated in Figure 2(2)(3). We also apply a hierarchical clustering technique to generate clusters of data items and time steps independently. Finally, we construct a heatmap reflecting the clustering results, as illustrated in Figure 2(4). As a result, we can easily observe clusters of similarly varying or correlated data items. Also, our implementation can selectively remove outlier time steps from the heatmap based on the clustering result.

3.2. Distance Calculation and Scatterplot Generation

It is important to understand relationships among variables in many time-varying data analysis applications. Thus we provide a scheme to select sets of dimensions in which similar or highly correlated variables are closely displayed in a heatmap.

For the set of items D, we first calculate the distances between all possible pairs. Our current implementation defines the distance between the j-th and the k-th items as:

$$dsta_{ik} = 1.0 - |f_c(j,k)| \quad (1)$$

where $f_c(j,k)$ denotes Spearman's rank correlation coefficients. This definition means that positively or negatively correlated numeric dimensions have similar distances. Other definitions (e.g. Euclidian distance or Cosine distance) can be also applied to this implementation. The technique then calculate the positions of a_i based on the distances $dsta_{jk}$ by applying MDS (multi-dimensional scaling)[1].

Also, the presented technique applies a hierarchical clustering algorithm with minimum or maximum distance methods to construct a dendrogram of the set of items. Given a threshold $dsta_{select}$, our implementation constructs clusters of items using the dendrogram. Large clusters are easily generated applying minimum distance method, while balanced set of clusters are often generated applying maximum distance method. Finally, we reorder the items in each of the clusters so that well correlated items are adjacently placed in the heatmap.

Our implementation similarly applies the above process to the set of time steps T, specifying a threshold $dstt_{select}$. We suppose to apply minimum distance method for the construction of the dendrogram, because it has a preferable property to generate large clusters and eliminate outliers from the clusters.

3.3 Heatmap Design

Figure 3 shows the mechanism of our heatmap design. Our implementation extracts and displays clusters of data items, and also draws gray borders between two of the clusters. It fixes the arrangement of items in the heatmap as ordered by the dendrogram so that users can keep their mental maps. Meanwhile, the technique applies the

[1] We apply the classical MDS implemented in MDSJ: Java Library for Multidimensional Scaling (Version 0.2). Available at http://www.inf.unikonstanz.de/algo/software/mdsj/. University of Konstanz, 2009.

similar clustering process to time steps, and extracts the time steps from meaningful sizes of clusters. It then arranges the extracted time steps in the temporal order and places along the horizontal axis of the display space. We can define the minimum number of items and time steps, $na_{\min}$ and $nt_{\min}$ in the extracted clusters. Our implementation sets $na_{\min} = 2$ to eliminate isolated items and focus on groups of similar or correlated items. Also, it sets $nt_{\min} = 2$ to eliminate outlier time steps.

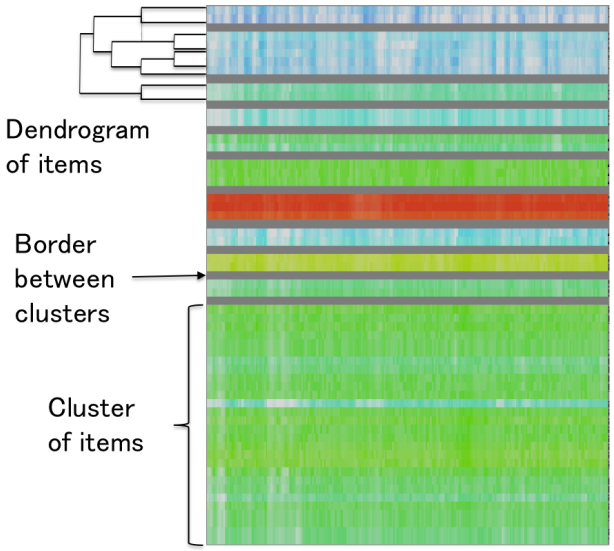


Figure 3: Heatmap design. Our current implementation displays clusters of items as ordered by the dendrogram.

We prepare two types of color maps applying the HSB color system as follows. Here, $D = \{a_1,...,a_m\}$ is the normalized value of v_{ij} by the following equation,

$$v'_{ij} = (v_{ij} - v_{\min}) / (v_{\max} - v_{\min})$$

where $v_{\min}$ and $v_{\max}$ are the minimum and maximum values.

Color map 1: Simply vary hues, as huge number of visualization software has already applied, as following equations,

$$H = hue(v'_{ij}), S = const_S, B = const_B$$

Color map 2: Calculate hues from average values, and saturation from relative values, as following equations,

$$H = hue(v'_{ij}), S = saturation(v''_{ij}), B = const_B$$

where *hue*() and *saturation*() are functions to calculate hue and saturation, $\overline{v'_{ij}}$ is the average values of an item, and v''_{ij} is the relative value of v'_{ij} calculated as $v''_{ij} = v'_{ij} - \overline{v'_{ij}}$.

3.4. User Interface Design

Our implementation displays a heatmap at the left side, and two scatterplots at the right side of the drawing space. Two sliders are featured to adjust the thresholds $dsta_{select}$ and $dstt_{select}$, as indicated by a red circle in Figure 1. Scatterplots draw connection segments between two dots if their distance is smaller than the thresholds. Users can interactively observe how the data items or time steps form clusters by observing the connection of dots in the scatterplots. The implementation invokes the clustering process when the mouse button operating the sliders is released.

4. Examples

This section shows examples of the heatmap-based visualization results. We implemented the technique with Java Development Kit (JDK) 1.8.0 and Java binding OpenGL (JOGL) 2.2, and executed on Apple MacBookAir with Mac OS 10.8. We applied Japanese weather data recorded by AMeDAS (Automated Meteorological Data Acquisition System) to the presented technique. We extracted time-varying temperature data observed in every 3 hours. This section shows heatmaps applying color map 2.

Figure 4 shows scatterplots of data items while adjusting the $dsta_{select}$ value. Dots enclosed by red circles are connected in this figure. These three scatterplots represent that larger number of smaller clusters were constructed when $dsta_{select} = 0{,}19$, while smaller number of larger clusters were constructed with larger $dsta_{select}$ values. Users can intuitively understand how clusters merge or split while looking at the scatterplot and operating the slider widget interactively to adjust the $dsta_{select}$ value.

Figure 5 shows heatmaps while adjusting the threshold $data_{select}$. Here, terms in Japan map in Figure 5 denote the names of regions, and red segments in the map depict mountain ranges. Only five clusters appeared in the heatmap when $dsta_{select} = 0{,}15$, where the displayed data items correspond to temperature in Tohoku and Kyusyu regions. Adjusting as $dsta_{select} = 0{,}19$, more clusters appeared in the heatmap corresponding to other regions including Kanto, Tokai, and Chugoku. Selecting larger $dsta_{select}$ values, temperature in Okinawa and Hokkaido are finally displayed. This is a reasonable visualization result because observation points are sparsely placed in Hokkaido and Okinawa regions relatively, and therefore variation of temperature was not very similar in these regions.

Figure 6 shows an example of specific time step extraction. Figure 6(Left) shows a heatmap representing all the time steps, and Figure 6(Center) shows a smaller heatmap after adjusting the threshold $dstt_{select}$. Figure 6(Right) shows a time step scatterplot, where dots enclosed by red circles correspond to the time steps

represented in the heatmap shown in Figure 6(Center). Users can interactively select well-correlated time steps and eliminate outlier time steps just by adjusting a threshold with a slider widget.

5. Conclusion and Future work

This paper presented a heatmap-based time-varying data visualization technique. This technique applies clustering algorithms to data times and time steps respectively, so that we can filter non-important data items and noisy time steps, and display appropriate size of heatmaps. Our implementation displays two scatterplots as well as a heatmap, where the scatterplots represent the distribution of distances among data items and time steps respectively. We can intuitively understand how data items and time steps are clustered by looking at the scatterplots while interactively adjusting the thresholds of the clustering processes.

Our study in this paper is still in the early stage and therefore we have many future issues. We would like to reconsider and extend our implementation as follows.

Firstly, we would like to apply other definitions of distance calculation among data items and time steps. Also, we would like to apply various schemes for hierarchical clustering in addition to the minimum distance method which has been applied to our implementation. We would like to evaluate the variation of distances and clustering implementation to realize better visualization results.

Our implementation addresses on representation of similar or correlated groups of data items, and removal of noisy time steps. On the other hand, it is also meaningful in many cases to focus on visualization of outlier data items or time steps. We would like to extend our implementation to selectively visualize such outlier factors to discover other kinds of knowledge.

Finally, we would like to apply various datasets including real business datasets, and conduct user experiences to evaluate the usability.

References

[1] R. Chang, M. Ghoniem, R. Kosara, W. Ribarsky, J. Yang, E. Suma, C. Ziemkiewicz, D. Keim, A. Sudjianto, WireVis: Visualization of Categorical, Time-Varying Data from Financial Transactions, IEEE Symposium on Visual Analytic Science and Technology, 155-162, 2007.

[2] A. Hayashi, T. Itoh, S. Nakamura, A Visual Analytics Tool for System Logs Adopting Variable Recommendation and Feature-Based Filtering, 17th International Conference on Information Visualisation (IV2013), 1-10, 2013.

[3] M. Imoto, T. Itoh, A 3D Visualization Technique for Large Scale Time-Varying Data, 14th International Conference on Information Visualization (IV2010), 17-22, 2010.

[4] H. Suematsu, S. Yagi, T. Itoh, Y. Motohashi, K. Aoki, S. Morinaga, A Heatmap-Based Time-Varying Multi-Variate Data Visualization Unifying Numeric and Categorical Variables, 18th International Conference on Information Visualization (IV2014), 84-87, 2014.

[5] C. Turkay, A. Lundervoid, H. Hauser, Representative factor generation for the interactive visual analysis of high-dimensional data IEEE Transactions on Visualization and Computer Graphics, 18(12), 2621-2630, 2012.

[6] S. Yagi, Y. Uchida, T. Itoh, A Polyline-Based Visualization Technique for Tagged Time-Varying Data, 16th International Conference on Information Visualization (IV2012), 106-111, 2012.

[7] X. Yuan, D. Ren, Z. Wang, C. Guo, Dimension Projection Ma- trix/Tree: Interactive Subspace Visual Exploration and Analysis of High Dimensional Data, IEEE Transactions on Visualization and Computer Graphics, 19(12), 2625-2633, 2013.

[8] Z. Zhang, K. T. McDonnel, E. Zadak, K. Muller, Visual Correlation Analysis of Numerical and Categorical Data on the Correlation Map, IEEE Transactions on Visualization and Computer Graphics, 21(2), 289-303, 2015.

[9] H. Ziegler, M. Jenny, T. Gruse, D. A. Keim, Visual Market Sector Analysis for Financial Time Series Data, IEEE Symposium on Visual Analytics Science and Technology, 83-90, 2010.

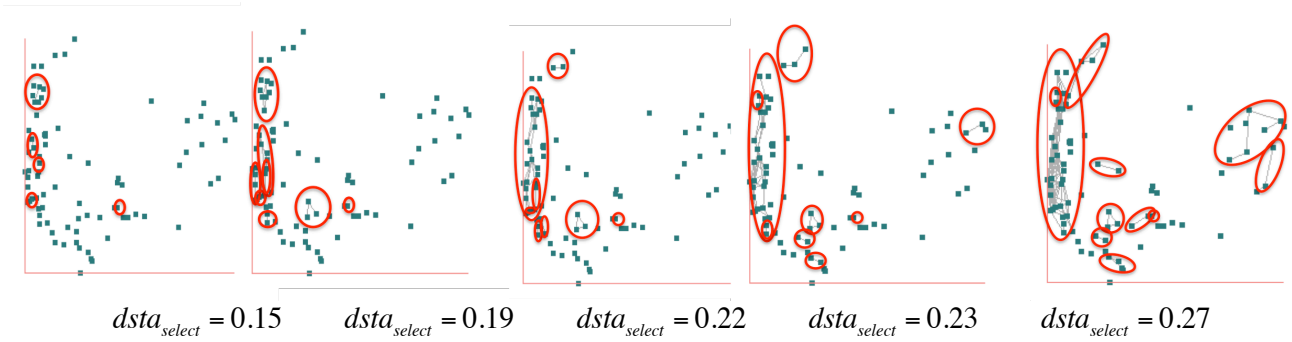
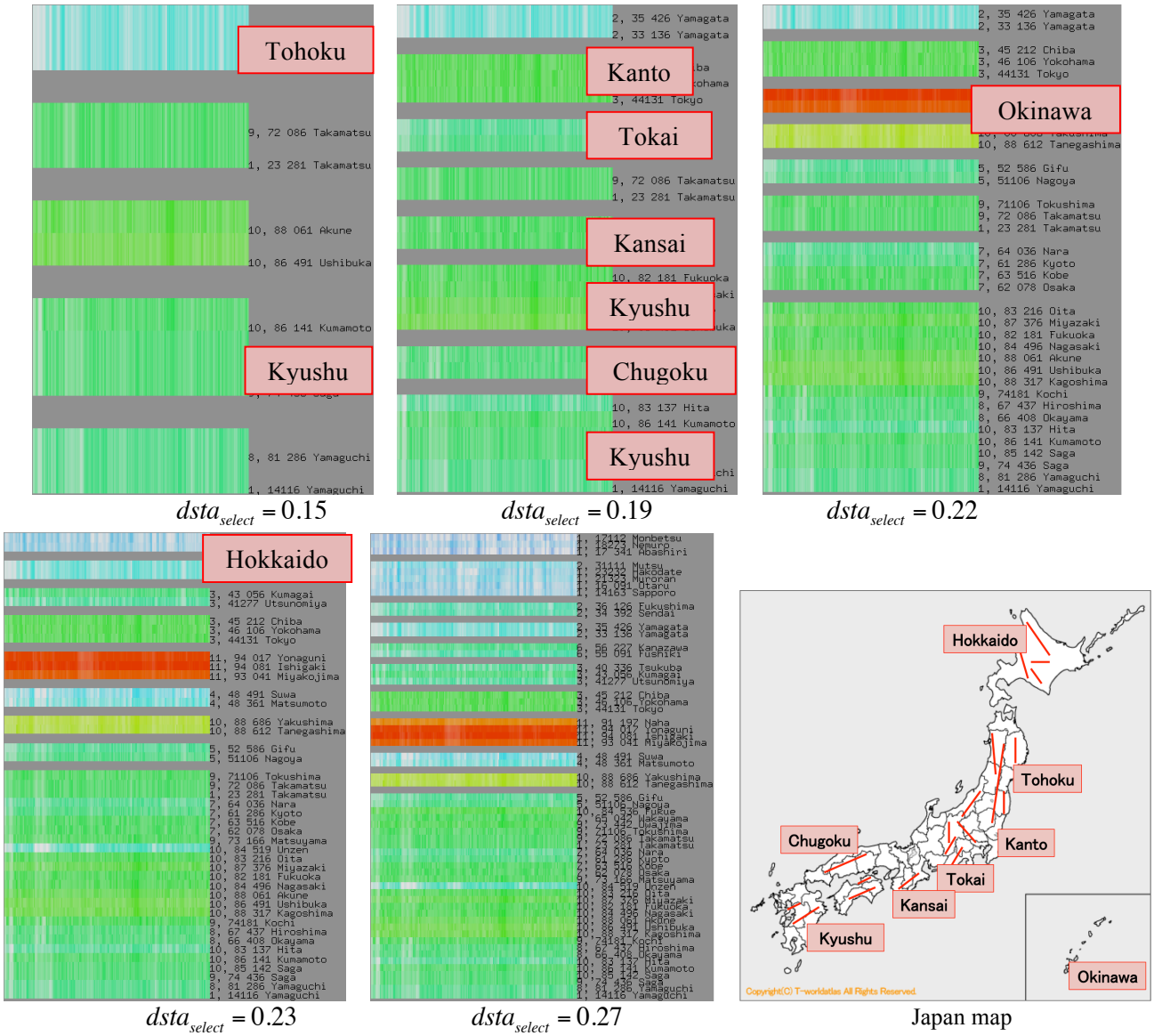


Figure 4: Example of scatterplots of data items.



$dsta_{select} = 0.15$ $dsta_{select} = 0.19$ $dsta_{select} = 0.22$

$dsta_{select} = 0.23$ $dsta_{select} = 0.27$ Japan map

(Authors added captions on the Japan map [World Map | SEKAICHIZU]. http://www.sekaichizu.jp/atlas/japan/p500_japan.html, 2016/02/12)

Figure 5: Example of heatmaps applying time-varying temperature data in Japan.

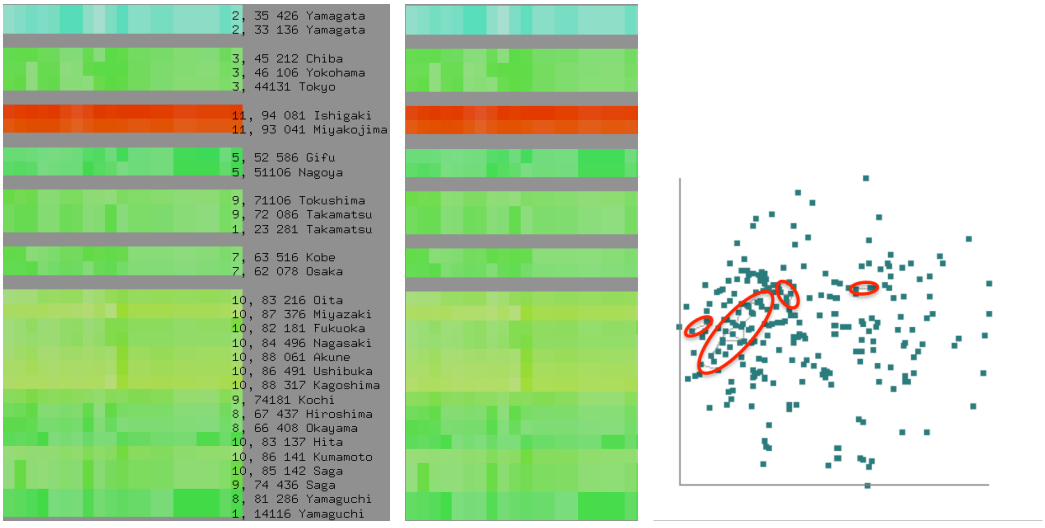


Figure 6: Example of heatmaps with specific time step extraction.