Irregular Trend Finder: Visualization tool for analyzing time-series big data

Shinnosuke Takeda∗  Aimi Kobayashi∗  Hiroaki Kobayashi∗  Saori Okubo∗  Kazuo Misue†

University of Tsukuba, Japan

ABSTRACT
We created a visualization tool called Irregular Trend Finder (ITF) for the VAST Challenge 2012 Mini Challenge 1. ITF is an interactive tool designed for analyzing large amounts of data with timestamp and hierarchy structure in mind, so that a user can see the overview first, and then obtain more detailed information. We discovered the answers for the Challenge, and confirmed that overview first and then looking in detail is an efficient method to seek particular information in case of large amounts of data.

Keywords: Information Visualisation, Anomaly Finder, Time-series Data, Big Data

Index Terms: H.5.2 [Information Systems]: Information Interfaces and Presentation—User Interfaces

1 INTRODUCTION

Figure 1: A View with Mosaic-Mode at 14:00 on February 2nd

We attempted VAST Challenge 2012 by using an original visualization tool called Irregular Trend Finder, or ITF. In this challenge, participants were given a set of large data in an imaginary world network with each machine’s health status. The mission was to find "anomalies" within the network using some kind of analysis tool. ITF’s goal was set to obtain overview and see detailed information about the anomalies with speed in mind.

2 IRREGULAR TREND FINDER

2.1 Approach
According to Schneiderman’s visual information-seeking mantra [3]: “Overview first, zoom and filter, then details on demand”, we designed ITF for the user to be able to find anomalies in such a way. The amount of given data was so large that it was hard to show all the data at the same time. Therefore, we focused on percentage.

For grasping the situation roughly with regards to large data, it is enough to show stochastic status. If the user notices that there are some kinds of anomalies, the tool provides a way to see more detailed information about them. Thus our ITF has 3 views: first view, where the user can see the percentage of status of all machines in the company (Bank of Money), second View in Region or DataCenter, and third View in Branch or Headquarters, telling each individual machine’s status.

Also, we previously arranged and calculated the data for speedy output and interactive usage. The time-series records about the same machine are gathered from the given data (CSV). To facilitate an interactive analysis for the user, we made data of machines that belong to each Branch or Region to come fast from the server at the user’s request. The stored information and time-series data about each machine are a kind of Key-Value Store: KyotoCabinet1 so that the server can reply to the request. The program, written in Ruby 1.9 as a CGI, extracts proper information from data in this KVS, processes quickly, and gives a response to the client.

2.2 Overview
ITF was developed by Processing with JSON library. It has a hierarchical structure and it consists of 3 views, from top to bottom: All-View (AView), Region-View (RView), and Branch-View (BView). The user can see more detailed information with deeper View. Once the user takes a look at a certain View, (s)he is able to see another view by switching the tab on the upper area of the window. Also the user can clear the view by clicking the current tab (except AView). In every View, we are able to select and see the change over time of PolicyStatus or ActivityFlag.

2.3 AView
In AView, a rectangle means a Region or a DataCenter. The size of the rectangle depends on the number of machines in the area. This View has 56 domains which consist of 1 Headquarters/headquarter, 5 Headquarters/DataCenter and 50 Regions. Each rectangle shows the visualization of the percentage of PolicyStatus or ActivityFlag of all machines in the area. This manner of visualization was inspired by [1, 2, 4].

The View provides 4 Modes: Mosaic-Mode, Linechart-Mode, Four-Hours-Linechart-Mode and Histogram-Mode. In each mode, the user is able to use a filter that exaggerates small percentages. A common color representation rule governs all Modes. The values from 1 to 5 of PolicyStatus or ActivityFlag are mapped as follows: 1 to pale blue, 2 to light green, 3 to orange, 4 to red, 5 to purple and no value (i.e. machine off) to white.

For grasping the situation roughly with regards to large data, it is enough to show stochastic status. If the user notices that there are some kinds of anomalies, the tool provides a way to see more detailed information about them. Thus our ITF has 3 views: first view, where the user can see the percentage of status of all machines in the company (Bank of Money), second View in Region or DataCenter, and third View in Branch or Headquarters, telling each individual machine’s status.

Also, we previously arranged and calculated the data for speedy output and interactive usage. The time-series records about the same machine are gathered from the given data (CSV). To facilitate an interactive analysis for the user, we made data of machines that belong to each Branch or Region to come fast from the server at the user’s request. The stored information and time-series data about each machine are a kind of Key-Value Store: KyotoCabinet1 so that the server can reply to the request. The program, written in Ruby 1.9 as a CGI, extracts proper information from data in this KVS, processes quickly, and gives a response to the client.

2.2 Overview
ITF was developed by Processing with JSON library. It has a hierarchical structure and it consists of 3 views, from top to bottom: All-View (AView), Region-View (RView), and Branch-View (BView). The user can see more detailed information with deeper View. Once the user takes a look at a certain View, (s)he is able to see another view by switching the tab on the upper area of the window. Also the user can clear the view by clicking the current tab (except AView). In every View, we are able to select and see the change over time of PolicyStatus or ActivityFlag.

2.3 AView
In AView, a rectangle means a Region or a DataCenter. The size of the rectangle depends on the number of machines in the area. This View has 56 domains which consist of 1 Headquarters/headquarter, 5 Headquarters/DataCenter and 50 Regions. Each rectangle shows the visualization of the percentage of PolicyStatus or ActivityFlag of all machines in the area. This manner of visualization was inspired by [1, 2, 4].

The View provides 4 Modes: Mosaic-Mode, Linechart-Mode, Four-Hours-Linechart-Mode and Histogram-Mode. In each mode, the user is able to use a filter that exaggerates small percentages. A common color representation rule governs all Modes. The values from 1 to 5 of PolicyStatus or ActivityFlag are mapped as follows: 1 to pale blue, 2 to light green, 3 to orange, 4 to red, 5 to purple and no value (i.e. machine off) to white.

Figure 2: A part of AView with Mosaic-Mode at 14:00 on February 2nd

1http://fallabs.com/kyotocabinet/
Mosaic-Mode
In this Mode, each domain is subdivided, and filled with colors corresponding to each status. The rate of color existence is decided by the percentage of each status. For example, the facility which is the majority of Status 1 looks like pale blue, and the majority of Status 2 looks like light green.

Histogram-Mode
The user can see histogram of the percentage of each status in this Mode. The top of each domain means 100%, while the bottom means 0%. From left to right, each bar stands for Status 1, 2, 3, 4, 5 and shutdown, and is filled with each assigned color. In addition, each percentage is shown as text on each rectangle.

Linechart-Mode & Four-Hours-Linechart-Mode
These modes literally draw using the line chart. As with Histogram-Mode, the top of each domain means 100%, and the bottom means 0%. However, these are different from the other modes, because they are able to see the transition of the percentage over time. The user can see the time series percentage from left to right.

2.4 RView
AView and RView are similar and each have the same 4 Modes. However, in RView, a rectangle means a Branch or Headquarters. The user can use this View to focus on a certain region and see detailed information about headquarters and Branches in the same Region. AView and RView show the percentage of each status which was compiled by each Region, Branch, headquarters, DataCenter.

2.5 BView
With BView, the user can see each machine’s PolicyStatus or ActivityFlag, and the Number of Connections.

If you refer to Fig.4, a figure like a gear (we called this “Time Series Gear”) on the left hand side stands for each machine, and when a Gear is clicked, the detailed view appears on the right hand side. At the top of this area, there is a bigger Gear that provides more details and longer past information than that of smaller Gear. Also, at the bottom, the detailed status of the selected machine is given.

Every Gear’s color depends on the machine class. If the machine class is “server”, the color is cyan, in the same way, “workstation” is purple, and “atm” is light green. The taller the “tooth” and deeper the color of the Gear, the higher PolicyStatus or ActivityFlag the machine has at that time. In the case of no value, the tooth has no height and its color becomes black.

In addition, the red line represents the Number of Connections of the machine. With both Gear and red line, a whole circle means a day, and the latest status is shown at top of the circle, while the oldest is at the center of the circle.

3 ANALYSIS PROCESS
In Mini Challenge 1.1, the participants should find an anomaly at 2:00 p.m. on February 2nd. Primarily we found that there were many white rectangles in Region-25, meaning that a lot of machines in Region-25 are shutdown (see Fig.1 and Fig.2). Therefore we watched this Region with Linechart-Mode which indicates the change of the status along time. It appeared that, as time passes, white lines (i.e. percentage of turned off machines) went up gradually from 4:00 and reached the highest point at 14:00.

We inspected further in the same way in AView to grasp the status of each Branch in region in RView (see Fig.3). As a result, we found that machines in Branch-33 and 39 showed anomalous status at first. As time went by, each Headquarter or Branch became healthy after 14:00. These discoveries were made with Four-Hours-Linechart-Mode. It was also clear that the percentage of turned off machines goes down by 19:00 in all Branches.

Moreover, in Region-26/Branch-30 at 7:30 on February 2nd, we noticed that on the left hand side there was one gear with a deep purple pie (see Fig.4). We chose this gear (IP = 172.41.188.35), checked further in time, and discovered that this terminal has had PolicyStatus 5 all the time from 7:30 on February 2nd.

4 CONCLUSION
By using ITF, we could notice anomalies along Schneiderman’s scenario. We discovered the answers for the Challenge using ITF which is specialized for analyzing large time-series data. ITF has a hierarchy structure so that the user can see the overview of the data first, and then obtain more detailed information. We also found that the Schneiderman’s mantra is useful to find particular information, not only in case of regular size data, but also in case of large size data. As future work, we want to improve ITF that can be used with other types of large-scale data sets.

ACKNOWLEDGEMENTS
The authors would like to thank Dr. Simona Vasilache for her valuable comments.

REFERENCES