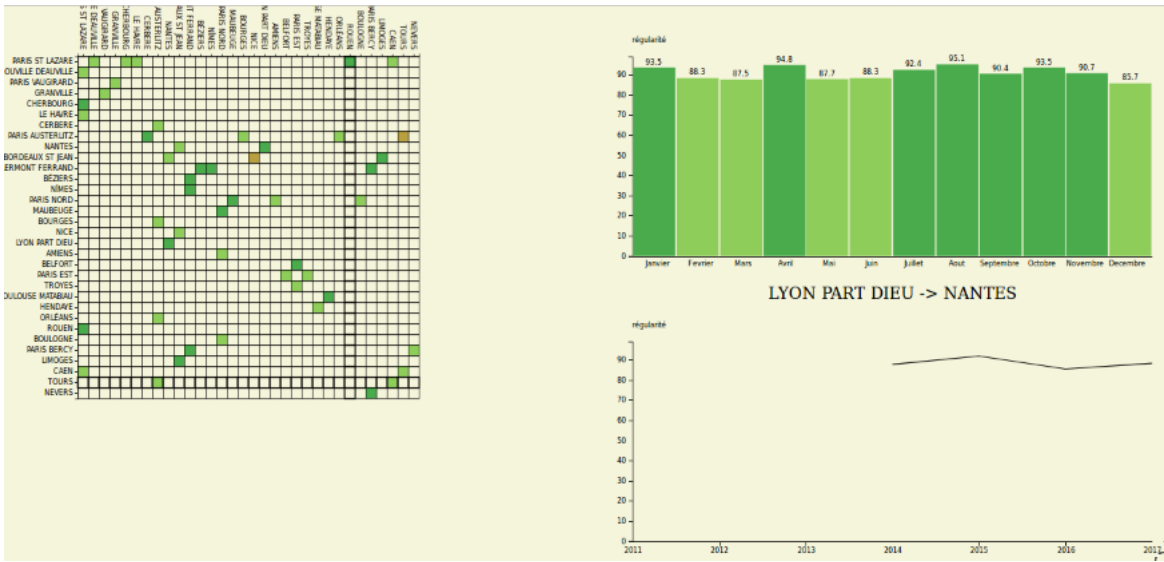


Are SNCF's TGV trains always late?

Isabelle Flores
Univ Claude Bernard Lyon 1

Enzo Lebrun
Univ Claude Bernard Lyon 1

Romain Candy
Univ Claude Bernard Lyon 1



1 ABSTRACT

Representation of network data traffic into matrix of adjacency with other juxtaposed visualization:

The representation of networks is mainly done by graphs but this kind of representation can be messy when it's deals with heavy graphs such as traffic flow because links between each nodes often overlap to each others. Moreover, the problem we are going to enlighten with our representation deals with geographical aspect and we will show how to get rid of this dimension in order to involve some others facts.

The adjacency matrix, is a way to represent graphs by a matrix, each line and columns represent a node of a graph and a square of a matrix represent a link between the two corresponding nodes. It is an underestimate way to deal with heavy graph, so we are going to expose how to use this method for this kind of problem.

Moreover the juxtaposition is a way to add an other visualization next to the matrix of adjacency, the problem is to not lose the user with too much information, so we are going to expose how juxtaposition can guide the user onto a very complex problematic.

2 INTRODUCTION

A graph is a way to represent dependencies between entities, each dependencies is represented by an edge between two (or more) vertices. It is a very convenient way to represent many problems in computer science because most of the time we get rid of any combinatorics issues such as exponential complexity and so on. It is also a way to represent real things such as traffic and evolution of things. The main issue with graph is that they are mainly adapted for computers than for human especially when they are very heavy.

The visualization of network problematic is now a hot topic in many fields of research such as traffic, cloud computing, scheduling and so on. The Big issue with networks visualization is that we do not visualize a big network the way we visualize a small one, we do not visualize a thick network the way we visualize a light one, or a directed one. There is such a high variety of graphs and there is no ready to wear only made-to-measure. But this article will expose some good methods that we can use for graph representation in order to solve a problematic. In this article we will lean on a question to solve, and try to use the best graph representation methods in order to answer this question. The question is "Are SNCF trains always late?". This problematic is very relevant because it concerns everyone and then we will be able to see if the representation we will use is adapt to a non-computer-scientist public. The SNCF train networks is a very thick graph so the representation will be adapted to this problematic, this article will show you why the adjacency matrix fits our problematic, why it is very interesting on directed graphs, and how we will deduce some other information from it.

Then we will use some bar chart representation in order to show the evolution of a specific node in the graph. The user graphical interface will be very user friendly, the user will have to click on any box of the matrix in order to see some information about the traffic of this specific line.

We got two inputs in the JavaScript, the different stations of the graph and also all the links of the graph [source, target, weight, year, month, comment] so we built the $N \times N$ matrix, for the matrix ordering, it is related to the order of the names on the input so we can apply a sort algorithm bases on the links, with this we decided to order the matrix based on the mean of the weight for a specific station. The results are not really relevant, this must need some improvements. The visualization start in 2011 so the we get all the data gathered during 2011 and we color the matrix in function of the mean of the year. Then the algorithm is waiting for the user to click on a square. Once the user click on a square, we get all the data coming from this specific train line for this specific year and we draw the bar chart month by month. Now the visualization is waiting for the user to pass his mouse over the months of the bar chart in order to display a comment of the problems that the SNCF encountered during this date and also to draw the line chart of the evolution of the traffic quality during this month for every year

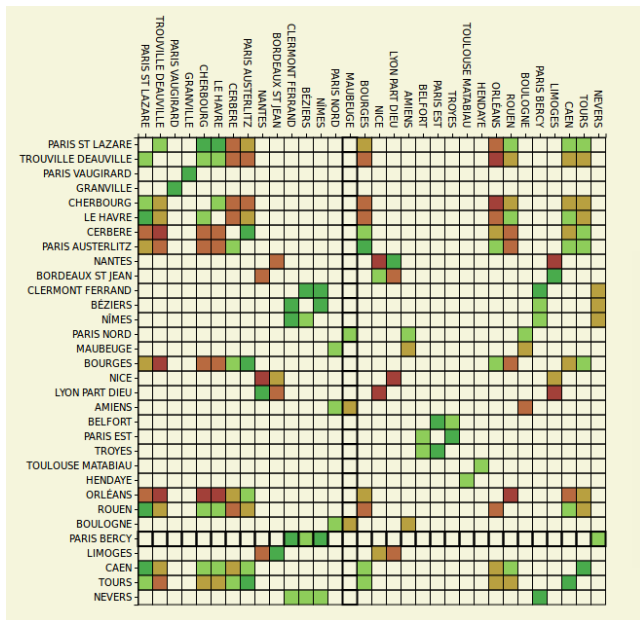


Figure 2: multiple

As you can see with the figure above, we inferred on the data by adding all possible path between the stations, as mentioned before, those estimation of quality traffic is quite naive and severe.

concerned by the visualization.

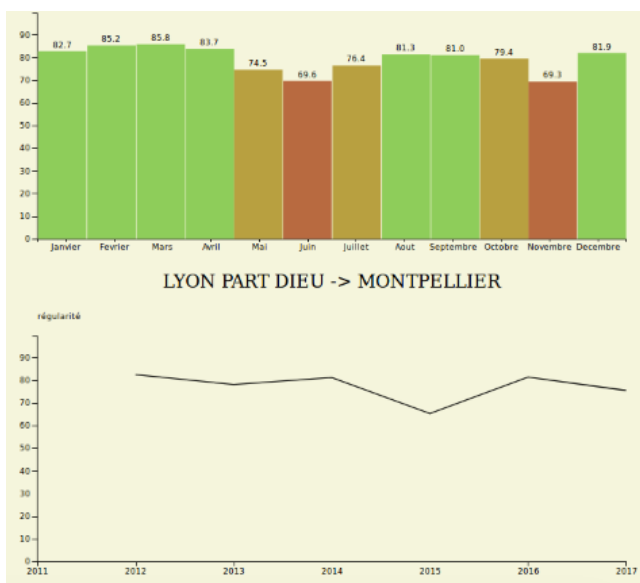


Figure 3: linechart

The mentioned linechart displaying the evolution of this month over the years

In terms of performances, the multiple train line matrix is quite heavy for the visualization and it might also be a good point of improvements for the next generation of this algorithm.

There is three distinct visualizations, each one showing one aspect of the data that we thought would be interesting. You can chose the year thank to a slider and which data you want to show thanks to radio buttons. The whole visualization follow a color gradient for the regularity.

We have five distinct colors :

- deep green : 90% - 100%
- light green : 80% - 90%
- yellow : 70% - 80%
- light red : 60% - 70%
- deep red : 50% - 60%

We made that color choice because there is very little data that go under 50% and we thought that 75% was already pretty bad. For the regularity that is less than 20%, we observed a dull version of the last color, but we're not sure why. The first visualization is an adjacency matrix that show the mean regularity for a year that can be chosen by a slider. To draw the matrix, we need to make a dictionary that take the source combined with the target as identified and the regularity, name weight, as value. Each square of the matrix take the information of this dictionary. A tool-tip show the regularity percentile of the square when we are over it and you can click on it to show more information on the right side. On the right side of the matrix, there is the second and third visualization. The second visualization is a histogram that show the regularity by month of the selections month and year. A tool tip show the commentary made about a specific month, when it exists, in order to help understanding why one specific month got a really bad result. The regularity percentile is show above the bars, it gives a more accurate information if we're interested and it also helps to differentiate a 0% from a missing value (and yes, we have at least one value at 0% in the intercity data). When you go over the bar of the histogram, the third and last visualization get to be used: It's a line chart that show the summary of the regularity over the years for those specific month and chosen train ride.

In terms of implementation we also worked on the matrix ordering in order to extract some clusters or some patterns. The main plan was to order the matrix by regularity rate of stations, thus we create an algorithm that sort the matrix in function of the mean of the regularity of each station, but this ordering methods was not very relevant, because most of the line are Paris centered, so there is no real cluster possible, it is more due to the data type than the ideas that sats within. Moreover there must be some better methods to cluster the matrix with some datamining algorithm but this is way beyond the limits of the possible for this specific project.

4.3 Scenario

The visualization is available on our web page, first of all the user have to chose the type of train he wants to visualize and if he wants to include non-direct paths, those options are made for the sake of simplicity, the multiple stations variation is more like an experiment of the limits of our visualization model, so if the user wants to have a crystal clear representation of the data which goes to the essentials it is better to chose the simple representation. We could have gathered the TGV and the intercity matrix together but because they are not connected by any station, it is better to keep them separated. For the sake of our scenario he has chosen TGV and simple paths, at this stage the chosen visualization will appear as a matrix, now the user can chose on the left the departure station and the arrival one on the top, he will have to click on the corresponding square on the matrix.

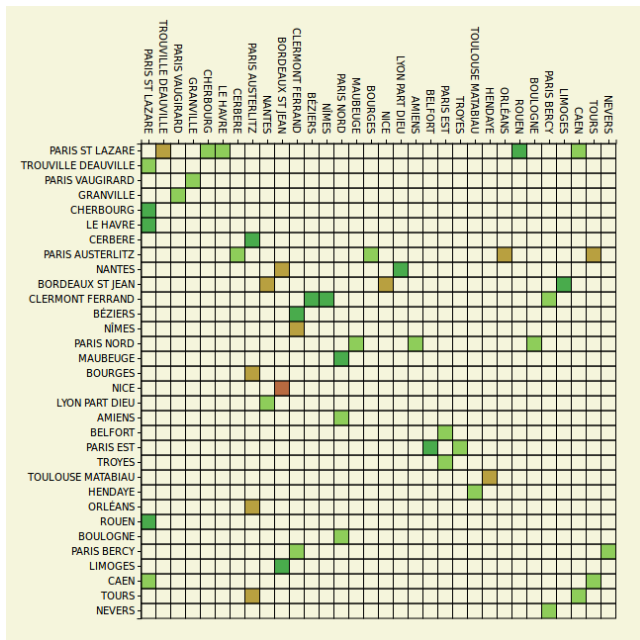


Figure 4: simple

As mentioned above, each colored square represents a rail line between two stations, the user can click on it in order to get more precise information.

For example our user as chose the line: MARSEILLE ST CHARLES ->LILLE, now on the top, the regularity bar chart will appear for this specific line, each bar represents a month, the color variate from green to red regarding the quality of traffic during this specific months for this specific year. When the user pass over a bar with his mouse, two things will be noticed:

A comment of the mains reasons for those delayed train, and on the bottom right corner a line of the variation of this specific month all over the years.

5 DISCUSSION

We did not made any measure of understanding time over a population of people so it's quite hard to talk about the quality of our specific visualization, In terms of data that we can deduce from the visualization, obviously the trains are not always late but some information just jump right to the eyes, as instance, the lines that

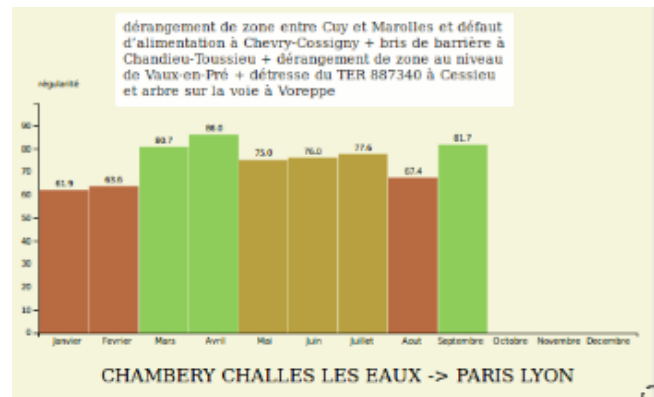


Figure 5: commentary

As you can see on the figure above, a bar chart is showing the evolution of quality traffic over a year, and when your mouse pass over a bar a comment will be displayed explaining the reason of the delayed train.

are not related to paris such as Lille -> Marseille have a low rate of regularity. Even if the multiple station matrix is quite messy, we can easily see that long train ride are more likely to be affected by a delayed train.

We also see that the year 2015 were pretty bad in terms of regularity, and the end (June to September) of the year 2017 is disastrous due to some natural disaster and other plethora of incidents.

6 POSSIBLE IMPROVEMENTS

There is many ways of improvements in our visualization one of them is the implementation of an ordering algorithm for the matrix, we now when to plug it on our code and we already tried one as mentioned before but this specific problematic suffer of a lack of knowledge from our side, there is some articles about this specific subject which deals with some common algorithm bases on column and line swapping. [2] We also didn't lean too much on those problematic because they were no good way to order the TGV matrix in a good clustered way, because most of the lines are centered on Paris, so this might also be a way of improvements, int the data acquisition we can add some other stations such as intercity with the TGV and then each region of france will represent a cluster of intercity trains and those clusters are connected together by the TGV. This part is really challenging because it needs a lot of data gathering and maybe the matrix would have been too big so we would have been forced to use the nodeTrix method mentioned in the article of Mr Fekete. Now we have a good knowledge of adjacency matrix, we will be able to add some new features such as other data like deleted trains instead of delayed ones, or maybe some prediction of the future of the traffic quality with the same visualization. In the multiple path problematic, the dijkstra algorithm is very naive because nobody take more than one TGV during a train travel, but this idea is good to keep and to improve by adding intercity trains in the TGV graph in order to have an accurate visualization. Moreover for multiple station lines, we considered that delayed trains are independents to one another, but this is very naive, as instance during strikes or when a train delayed all the other one in the line.

7 CONCLUSION

To conclude, there are several ways to represent flows in traffic network, we have chosen one that is free from any geographical representations, this choice was made because we didn't wanted the user to focus on the distance or the position but only on traffic quality, everything one the visualization is made for it, as instance, the juxtaposition methods is a way to represent data over time, once the user chose a square in the matrix everything is made for giving as much information as possible about the quality of traffic on this line over the year, with comments about the reasons of some delayed or problems. We think this web page is a real success and with further works on ordering and data gathering, it can be use by any companies or people that deals with problems on thick traffic graph representation over the time.

REFERENCES

- [1] Les misérables co-occurrence. <https://bost.ocks.org/mike/miserables/>. Accessed: 2017-11-25.
- [2] Françoise Bahoken, Laurent Beauguitte, and Serge Lhomme. La visualisation des réseaux. principes, enjeux et perspectives. 06 2013.
- [3] Jean-Daniel Fekete. Visualizing networks using adjacency matrices: Progresses and challenges. pages 636 – 638, 09 2009.