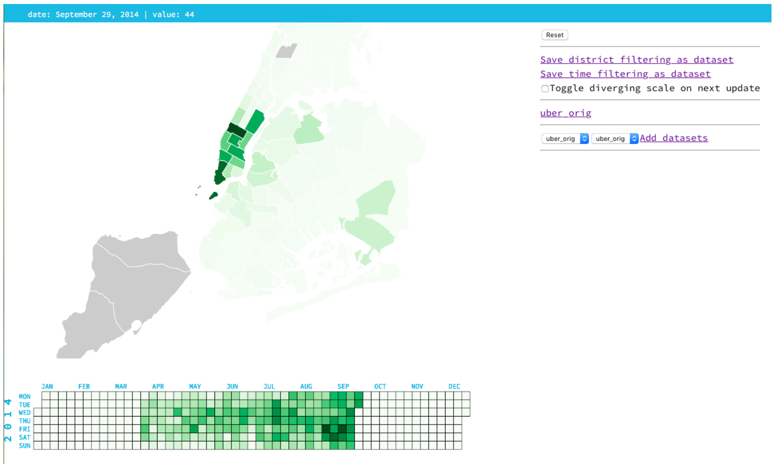
Density heat map visualization of spatially and temporally localized events

Pierre Couy[*]   Louis Manhès[†]   Hugo Fauvet[‡]

Universit Claude Bernard Lyon 1



date: September 29, 2014 | value: 44

□Toggle diverging scale on next update

Figure 1: Home view when initializing visualization

Abstract

The visualization of density data is very often done using simple heatmap. Even if the visual rendering seems attractive, the interpretation of these data is not necessarily the most accurate. Indeed, some bias may be caused by lack of correlation with external information or the presence of artifacts (outliers) from too small datasets. By taking these biases into account, we propose here to solve these problems by proposing an interactive visualization, allowing to put in relation different datasets between them, or even different subsets of the same dataset. In this work, we used the well-known Bayesian Surprise as a measure of comparison between two sets of data so as to firstly, limit the number of bias but also to create new information. Using the kaggle's Uber pickups in New York City, we demonstrate how Bayesian surprise overcome some limitations of traditional event maps and create useful additional information.

1 Introduction

Cross-referencing of data sources is very important in order to extract the least biased information. This information may be spatial, temporal or a combination of both. The bias is either due to a lack of data, which can result in irregularities or outliers in the visualization and corrupt the representation. But this bias may also be due to misinterpretation by the user. Indeed, a poorly chosen color map or event density information provided without normalization by an external knowledge, for instance standardizing human density events relative to the population density in a specific location, may be the cause for misinterpreting data by thinking an important pattern exists, when noise or bad sampling are better explanations.

In order to tackle these problems we introduced the well-known Bayesian-Surprise metric [5] as a measure of comparison between different datasets or even multiple subsets. Surprise quantifies how data affects a natural or artificial observer, by measuring the difference between posterior and prior beliefs of the observer. In other words, the most surprising events are those that induce large updates in beliefs about the model of expected event distributions.

We have tried this method on transportation data, more precisely the Uber pickups in New York during 2014. Thus, our target audience is made up of public transport committees, private companies from the transport sector or even individuals who are curious to know more about the hidden mechanisms of the use of transport facilities.

In this work, we contribute to a visualization technique based on a dual-component namely heatmap and calendar, that allows users to compare efficiently different datasets or multiple subsets of them by the use of Bayesian Surprise as a comparative tool. The benefits are two fold, firstly it greatly reduces the bias caused by multiple factors such as unbalanced data or artefacts of increased importance due to a dataset that is too small. But it also makes it possible to generate new information, which could not be present without linking different datasets.

In order to give further explanations about the use of Thematic maps based on Bayesian surprise, we will discuss similar work in the related work section. Then a precise report of the technical and design choices that constitutes the visualization is given in the next section. Finally last sections will be used to discuss project limitations and possible future work.

2 Related Work

This part depict the state of the art. Several Bayesian surprise studies can be found which depicts the usefulness of such a metric based on the Human attention. In addition to that we have collected existing techniques for visualizing spatially and temporally embedded event data. Finally we add some extensive studies on the choose of colormaps because we believe it is a crucial factor to take into account

[*] e-mail:pierre.couy@etu.univ-lyon1.fr
[†] e-mail:louis.manhes@etu.univ-lyon1.fr
[‡] e-mail:hugo.fauvet@etu.univ-lyon1.fr

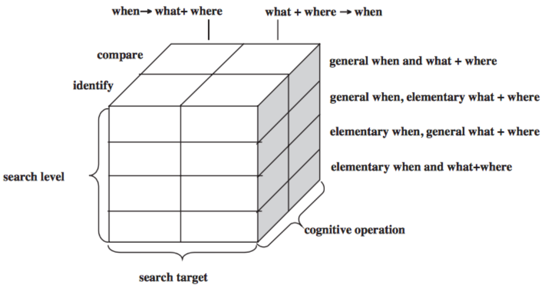

when→ what+ where
what + where → when
compare
identify
general when and what + where
general when, elementary what + where
elementary when, general what + where
elementary when and what+where
search level
cognitive operation
search target

Figure 2: Operational task typology used in [1]

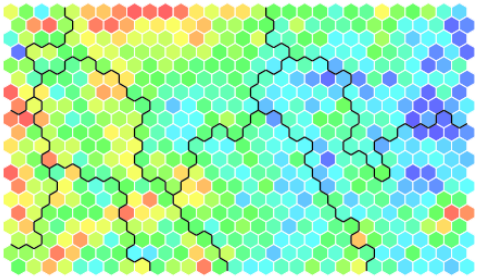


Figure 3: Example of an hexbinmap

when someone is using heatmap or thematic maps.

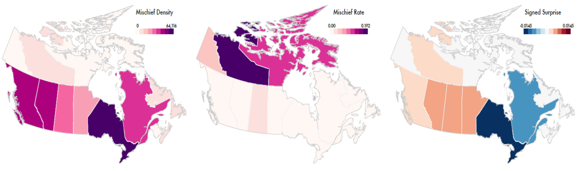
2.1 Event visualization

In order to use Bayesian surprise metric, one has to find a way of displaying some event data so that its interpretation is as clear as possible. Here we survey the related work and possible problems encountered when visualizing spatio-temporal data. In the next section we review Bayesian statistics and surprise measures in the context of displaying unbiased event data.

[1] systematized the existing methods for representing spatio-temporal information in traditional maps. The resulting catalogue of this study provides guidelines for selection of appropriate cartographic symbology depending on the data to be mapped. They introduced an analytical procedure based on answering the following three questions and their combinations : *what*, *when* and *where*. For example it could be : where are most events located? (*where*) or when do events reach a maximal density in this region? (*when*) or when does this maximal density, previously observed, reoccur in a different region? (*what*/*when*/*where*). 2 shows the operational task typology used in [1].

One approach to event visualization is to visualize individual streams of event density [2] as a 1D streams of event data. Other approaches seek to directly mapping events to points in 3D space [8]. The later lacks of simplicity when one need to discover patterns of interest. Indeed it requires interaction and 3D spatial reasoning in order to discover these patterns because of projection and occlusion issues. Finally, an other approach is to use thematic maps like choropleth or heat maps. The integration of animated flux or explicit differencing can be use to compare temporal patterns.

When one has a huge dataset of events, it may be infeasible to use a discrete glyph for each event without lacking clarity in the visualization and possibly missing interesting patterns. Some techniques are useful to tackle this kind of size problems : histograms using rectangular or hexagonal bins 3 or even sub-sampling techniques as depicted in [6]. But as we said earlier it is possible that these techniques induces some unexpected bias in the visualization, as we will see in the next section.



(a) The **Event Density** of "mischief" in Canada.

(b) The per-capita **Event Rate** of mischief.

(c) The **Surprise Map** of mischief.

Figure 4: Choropleth maps from [4] of (a) event density, (b) per-capita event rates, and (c) Bayesian surprise for mischief (a class of property crime) in Canada.

2.2 Bayesian Surprise

In order to counteract the biases of existing event visualization methods, we choose to rely on Bayesian modeling and Bayesian Suprise as depicted in [4].

The authors defines Bayesian surprise as "a measure of changes in belief by comparing the prior and posterior probability distributions. It captures the notion that large changes in belief are salient, and may characterize the importance of the data that caused these changes".

The basis of this measure stems from [5] in which the authors used Bayesian surprise as a technique to model human attention. They reported decent results in the field of computer vision and perceptual psychology.

As depicted in 4, we can see that Ontario and Quebec have crime rates lower than expected given their population. This is a typical application of what we wanted to achieve. The use of the Bayesian map provides additional informations that are essential for the true comprehension of the input data.

2.3 Color Scheme

The choice of a specific color map should be taken seriously into consideration and it should not be based on one's preferences. Indeed, a color map carries information that could lead to misinterpretations if not used correctly.

A wide variety of heat maps are used to display all kinds of data. These heat maps often use a color map ranging from blue or green for lower values to red for higher values. However, this is not always the best solution : depending on the type of data displayed, other color schemes may provide a more intuitive visualization.

In [7], Kenneth Moreland deals with the issue of rainbow color maps. According to him, and numerous other researchers in the field, this kind of color map is inefficient at meaningfully displaying scalar fields, while for instance giving the illusion of a gradient where there is none, or highlighting low interest regions of the visualization. To address this problem, the author discusses the use of diverging color maps.

Another point of view is to avoid as much as possible the use of colors in heat maps. In [3], the authors advise the use of gray-scale for heat maps for several reasons :

- Color maps lack the natural and intuitive ordering of gray scale
- Uncontrolled changes in luminance cause a loss of information when converting to gray-scale or printing
- Color maps can lead to the perception of a non-existent gradient

3 Project Description

3.1 Data acquisition

We chose to work on a method to visualize and analyze punctual events located in time and space. The dataset we decided to handle contains every Uber pickups in New York City from April to September 2014. These datas were released following a Freedom of

Information Law request submitted by the NYC Taxi & Limousine Commission in July 2015.

These raw datas have one entry for every Uber pickup recording the date and time as well as the longitude and latitude of the pickup. These informations allow us to consider each entry as an event that we can use in our visualization.

In order to compare effectively our main dataset with other datasets we implemented a function allowing us to gather every event happening in a same space region. For instance, to compare this dataset to the felony rate in New York City, implementing a police district repartition of Uber pickups is needed. This repartition was made possible due to the function coordsToPolygon that we developed. A geoJson is a file containing the drawing of borders used to define areas. Our function takes a geoJson as an input to create a version of the dataset where events are gathered by areas.

To keep the abstractness of our functions, it was necesseray to establish a standard in our data format. We consider each dataset as an array of events which are objects composed of four attributes :

- `polygon_id` representing the belonging of the event to a specific area according to the previously achieved slicing
- `time_begin` a time reference for the beginning of this event
- `time_end` a time reference for the end of this event
- `value` corresponding to the number of occurences of this specific event in the time span

We need to convert all of our data to this format before being able to display it. For instance, if a dataset describes the population of the different geographic areas defined in the geoJson by years, we get a data point for each polygon and each year in the dataset. The `time_begin` and `time_end` attributes are the timestamps respectively for the beginning and the end of the year.

For data with precise geographic locations, such as the Uber pickup dataset, the preprocessing mainly consists of determining which polygon contains the point at the given longitude and latitude.

Once standardized, our Uber pickups dataset present two specificities :

- its `time_begin` and `time_end` are the same
- its `value` is 1

These specificities are caused by the nature of the datas : one entry corresponding to one and only one pickup, it weights one unit, and pickups being punctual events, it is a single point in time.

After our datas are preprocessed, we need to aggregate it twice : once relative to polygons, and once relative to days. These data aggregations are performed in the functions which plot the map and the calendar. They both operate in a similar way by taking our array of events as an input and bringing together these events by district (respectively by day). This gathering consists on a sum of every value of the events with the same `polygon_id`. For day aggregation, we sum for each day the values of data entries for which the day is included between `time_begin` and `time_end`.

3.2 Scenario

Our visualization tool can be decomposed into three parts :

- a geographic map displaying the information
- a calendar heatmap allowing the user to select temporal point of interests
- a control panel

When the user accesses the visualization, there is already a single dataset preloaded : the Uber pickups in New York.

Hence the heatmap automatically displays the density of pickups for each district with a sequential green color map. The user can select one ore more districts by clicking on them while holding the *control* key, which results in a change in the calendar that will represent the temporal distribution of the event density during the year for these specific districts.

On the other hand, the calendar heatmap, bellow the map, displays the temporal side of the data : each square inside the calendar represents one day of the data and the intensity of the color corresponds to the average density of events during that day. The user can select days by clicking on them in order to filter a specific time frame which updates the map.

Finally, the user menu, located in the right side of the visualization, serves multiple purposes:

- allows the user to upload his own datasets
- allows the user to create new datasets based on the spatio-temporal filtering
- allows the user to compare datasets with the Bayesian Surprise

3.3 Description of the main dataset

In order to have some predefined transportation data to make our heat map visualization, we downloaded the Uber pickups in New York City Dataset from the web plateform Kaggle. This directory contains data from over 4.5 million Uber pickups in New York City from April to September 2014, and 14.3 million more Uber pickups from January to June 2015. The files are separated by month and each has the following columns :

- `Date/Time` : The date and time of the Uber pickup
- `Lat` : The latitude of the Uber pickup
- `Lon` : The longitude of the Uber pickup
- `Base` : The TLC base company code affiliated with the Uber pickup

We dismissed the Base attribute because such an information is not useful for the purpose of our work.

We discuss here the *Uber pickups* dataset, but a core aspect of our project is its ability to display any data provided in the CSV format with at least 3 columns : one for date and time, and two for latitude and longitude. However, even temporally and spatially localized data can be unfit for this visualization.

3.4 Visualization enrichment

To fulfill our objective of comparing and debiasing data, we let the user apply modifications on the base visualization : we offer a temporal and spatial filtering that can be saved into further subdatasets used for comparison. The two next sections depict respectively the spatial and temporal filtering.

3.4.1 District filtering

The spatial filtering is allowed by letting the opportunity to the user to select one or several districts to show the information for the selected districts only. The user can save this visualization for further treatments. 5 is a snapshot of the result of a spatial filter operation. By selecting some districts on the map, the calendar will automatically update its content in order to provide a temporal distribution of the density of events for all of the selected districts.

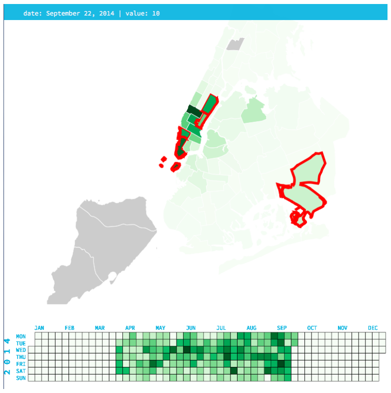
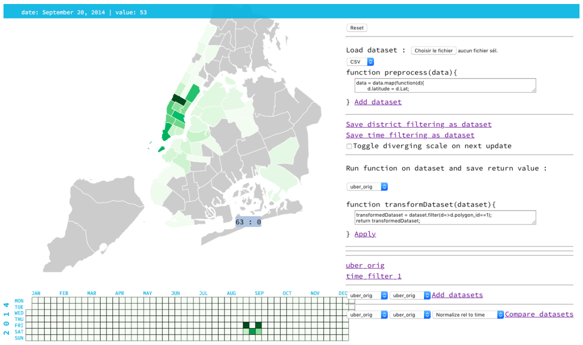


Figure 5: The user selected some districts on the map which result in the creation of a new dataset and an update of the calendar, which represents now the temporal distribution of these specific districts



Load dataset :

```
function preprocess(data){
    data = data.map(function(d){
        d.latitude = d.Lat;
} Add dataset
```

□Toggle diverging scale on next update

Run function on dataset and save return value :

```
function transformDataset(dataset){
    transformedDataset = dataset.filter(d=>d.polygon_id==1);
    return transformedDataset;
} Apply
```

Figure 6: The user selected 6 days on the calendar which result in the creation of a new dataset and an update of the heatmap

3.4.2 Calendar heatmap and time filtering

The temporal filtering can be found using our calendar heatmap. This visualization gives an easy access to the dates that are rich in information by bringing out the time points of interest from the dataset. The user can save these datasets just as in the previous filter. 6 is a snapshot of the result of a time filter operation. The user created a new dataset *timefilter1* once he selected a few days on the calendar.

3.4.3 Surprise visualization

In order to debias datasets, we implemented a function allowing the user to compare two datasets previously saved using our surprise management. To use this functionality, the user select two datasets to compare and normalize them in relation to time or space. Once the both data are normalized, the surprise is calculated as an evaluation of how accurate can be our prediction of the first dataset values using the second one.

For visualizing surprise, we recommend the use of a diverging color-map, which can be activated by checking the corresponding checkbox. The use of a diverging color map is justified by the fact that surprise values are centered on 0, positive surprise values corresponding to higher than expected values in the dataset, and negative values to lower than expected values.

4 Evaluation

Firstly, the major issue is the lack of a user-friendly interface. Indeed, without reading this report or the description page that is hosted on the project's directory, the user will have difficulties to take charge of all the project's functionalities. This represents a serious limitation for any new user that is not trained to use this tool.

We are also experimenting some latency when comparing large datasets due to some nested loops in the surprise evaluation, this is the reason why we took a subsample of the full 4.5 millions rows of the Uber pickup's dataset.

Finally, an interesting feature will be to give the user the possibility of downloading the datasets he created with the tool through multiple filtering and Bayesian Surprise.

5 Discussion

By developing this tool, we aims to give the user the power of having an insightful look at his data. The quite human Bayesian estimation of the surprise is in our opinion a powerful metric that can be used in a wider range of applications.

Plus we wanted to create a generic tool that could be used for any application outside the transportation field.

6 Conclusion

In this article, we give a detailed description of a visualization tool which handle spatio-temporal event data. Using the bayesian surprise, it is able to compare efficiently synthetic or real datasets in order to provide the user a comprehensive thematic map.

References

[1] N. Andrienko, G. Andrienko, and P. Gatalsky. Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages and Computing*, 14(6):503 – 541, 2003. Visual Data Mining. doi: 10.1016/S1045-926X(03)00046-6

[2] K. Beard, H. Deese, and N. R. Pettigrew. A framework for visualization and exploration of events. *Information Visualization*, 7(2):133–151, 2008. doi: 10.1057/palgrave.ivs.9500165

[3] D. Borland and R. Taylor. Rainbow Color Map (Still) Considered Harmful. *IEEE Computer Graphics and Applications*, 2007. doi: 10.1109/MCG.2007.323435

[4] M. Correll and J. Heer. Surprise! bayesian weighting for de-biasing thematic maps. 23:1–1, 01 2016.

[5] L. Itti and P. Baldi. Bayesian surprise attracts human attention. *Vision Research*, 49(10):1295 – 1306, 2009. Visual Attention: Psychophysics, electrophysiology and neuroimaging. doi: 10.1016/j.visres.2008.09.007

[6] A. Mayorga and M. Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Computer Graphics*, 19(9):1526–1538, Sept. 2013. doi: 10.1109/TVCG.2013.65

[7] K. Moreland. Diverging Color Maps for Scientific Visualization. *Proceedings of the 5th International Symposium on Visual Computing*, 2009. doi: 10.1007/978-3-642-10520-3_9

[8] C. Tominski, P. Schulze-wollgast, and H. Schumann. 3d information visualization for time dependent data on maps. pp. 6–8, 2005.