Analysis of Massive Heterogeneous Temporal-Spatial Data with 3D Self-Organizing Map and Time Vector

Yu Ding

Huazhong University of Science and Technology
Luoyu Road 1037, Wuhan, China
dingy@hust.edu.cn

Abstract

Self-organizing map (SOM) have been widely applied in clustering, this paper focused on centroids of clusters and what they reveal. When the input vectors consists of time, latitude and longitude, the map can be strongly linked to physical world, providing valuable information. Beyond basic clustering, a novel approach to address the temporal element is developed, enabling 3D SOM to track behaviors in multiple periods concurrently. Combined with adaptations targeting to process heterogeneous data relating to distribution in time and space, the paper offers a fresh scope for business and services based on temporal-spatial pattern.

Keywords: Self-Organizing Map, Multi-Period Pattern, Heterogeneous Data

1. Introduction

1.1. Background of Research

With the development of information gathering technology, people can access to tremendous amount of real-time occurrence data consisting of coordinates both in time and space, such as the e-commerce orders, Uber requests[1], crime incident reports[2], and vehicle collisions[3]. The massiveness conceals patterns requiring feasible a tool to identify. Following research is tightly related to their features listed below:

1) Extremely dense.

Density in both time and space makes it impossible to track every input, therefore, determining centroids that represent the cluster they belong is essential for
providing service based on time point and location, such as optimized warehouse site for delivery, effective patrol schedule and so forth.

2) Multi-Periodically Structured.
Data repeats in days, weeks and months, demanding us to analysis under multiple period. For example, vehicle collision’s distribution fluctuates within 24 hours and days in one week.

3) Heterogeneous.
Inputs include both numeric and categorical data. The crime incidents data not only covers time and geographic variables, but also classification of felonies. How to incorporate heterogeneous data is challenging yet useful.

The research was conducted in a progressive manner, beginning with basic clustering on time, latitude and longitude, aiming to solve the problem 1. Since not much prior experiment has been done to this type of data, making it necessary to test with different configurations and parameters to attain better accuracy. Next, the highlight of this passage, time vector is introduced to handle problem 2. Finally, adaptations on SOM itself are made to overcome the difficulties mentioned in 3.

1.2. Introduction on algorithm
Self-organizing map[4] is an unsupervised and efficient algorithm for clustering, which not only allows people to divide the data into sectors, but also to understand their topographic relation. Following terminology is introduced with potential application.

- Nodes, building up self-organizing map’s grid, the centroids of clusters.
- Neighborhood, a sector within certain radius centering around a selected nodes.
- Codebook Vector, indicating where a node is situated in coordinate’s of input data. Its value offers detailed information about the nodes, like when and where is most representative for a cluster of crimes.
- Best Matching Unit(BMU), the node with least distance to a chosen input. BMU can be regarded as location with least cost. For instance, a driver at BMU has the least distance to a customer.
• Hits, inputs that belongs to a specific cluster, like the recipients that a warehouse served.

2. Clustering with 3D Self-Organizing Map

2.1. Construction of SOM

The framework of 3D grid consists of layers covering latitude and longitude axes, while different layers array along time axis. Nodes in one layer are marked with same color.

![3D SOM Grid](image)

Figure 1: A $7 \times 6 \times 6$ 3D SOM Grid

2.1.1. Significance of Pre-Processing Input

According to previous research, normalization will promote the quality of map[5]. This paper compared two types of normalization, rescaling to $[0,1]$ and standardization with z-score, both of which improves map’s quality by drastically reducing errors introduced later. Detailed results is available in Appendix A1.

2.1.2. Initialization

In the research it’s found that in order to achieve a well-organized map, grid must be initialized linearly and the side that parallels to time axis should be initialized with
greatest number of nodes, in other words, the number of layers should be larger than
each side of layers. Moreover, initializing the map with different sizes yields an interest-
ing yet telling phenomenon that if lengths of grid’s sides are not selected properly, the map will rotate during reiterations, ending up with poor quality. Explanation is discussed in [Appendix A2].

2.1.3. Training Progress

After initializing, training of SOM is summarized as following:

1) Find BMU for every input under Euclidean metric;
2) Update the map in batch mode;
3) Reiterate from step one;

Since this research engages with heavy calculation, batch training is more preferable than sequential one, and using voronoi set can accelerate updating progress further[6] with following function.

\[
m_i = \frac{\sum_j n_j h_{c,i} x(t)}{\sum_j n_j h_{c,i}}
\]  

On the left, \(m_i\) is a codebook vector, and \(n_j\) is number of hits for node \(j\), \(h_{c,i}\) measures influence or weigh of node \(c\) to node \(i\), determined by Gaussian neighborhood function

\[
h_{c,i} = e^{-\frac{d_m(m_c,m_i)^2}{\sigma^2}}
\]

2.2. Measurement of Quality

2.2.1. Quantization Error

Quantization error(QE) measures average distance between inputs to their BMUs.

\[
QE = \frac{\sum_{i=1}^{n} d(x_i, m_j)}{n}
\]  

d\((x_i, m_j)\) is the distance of between input \(x_i\) and its best matching unit \(m_j\) and \(n\) is the total number of input. QE is important in application, for instance, selecting warehouse according to map with small QE can be economical for shipping industry.
2.2.2. Topographic Error

Topographic error (TE) measures the portion of input whose BMU and second BMU connected directly in grid.

\[ TE = \frac{\sum_{i=1}^{n} \delta(x_i)}{n} \]  

(3)

The value of \( \delta(x_i) \) is determined by whether the nodes has second minimum distance to input \( x_i \) is a neighbor to BMU. If so, \( \delta(x_i) = 0 \), otherwise it will be set as 1. If a map exhibits low TE, while the BMU’s site is not available in application like police station, second BMU might be good substitute.

2.3. Reliability Estimation

2.3.1. Generating Correlation Matrix

In order to fully evaluate map’s soundness in application, we need to ascertain whether centroids and their hits distribute correspondingly to density of input. It can be achieved by converting input’s regional density into frequencies as following steps:

1) Slicing the input space into cubics;

2) Generating a matrix contains numbers of inputs in each cubic. Precision can be set via choosing different pieces of slices;

3) Counting the centroids and hits in every cubic, obtaining other two matrices with same dimension;

4) Measuring their correlations to input frequency matrix.

If a map is representative, correlation coefficient (COR) should approximate to 1.

\[ COR(X,Y) = \frac{\sum_{j=1}^{m} \sum_{i=1}^{n} (x_{ij} - \bar{X})(y_{ij} - \bar{Y})}{\sqrt{\sum_{j=1}^{m} \sum_{i=1}^{n} (x_{ij} - \bar{X})^2 \sum_{j=1}^{m} \sum_{i=1}^{n} (y_{ij} - \bar{Y})^2}} \]

where \( \bar{X} \) and \( \bar{Y} \) are mean of matrices \( X,Y \).
2.3.2. *Projection*

Projecting high-dimensional into subspaces makes interpretation of interior relations more accessible. In clustering input with time, latitude and longitude, under the notion of projection, estimation can be divided into two parts:

- Spatial consistency, in fixed durations, measuring how distribution of nodes and their hits match with input. To be more concrete, it measures frequency matrix slices along the time axis.

- Temporal coherence, measuring if spatial consistency remain stable in different periods, by setting the durations to other periods.

In further analysis, projections is essential for assessing performance. While probating multiple period patterns, projecting data into different temporal dimensions allows us to observe variance under periods of varied durations. When input covers numeric and categorical variables, we need to evaluate accuracy of both temporal-spatial clustering and category classification.

3. Exploring the Potential of 3D SOM

Apart from identify centroids in time-latitude-longitude space, basic model is insufficient for demands in real world. First, only distribution in 24 hours is considered, whereas weekly or monthly fluctuation is ignored. In addition, input data comprises numeric elements and categorical ones, like name of purchased item, felony classifications, among which distance cannot be reckoned via Euclidean norm. Further analysis will investigate two possible solution to overcome these limitations.

3.1. *Broadening Timeline with Time Vector*

If time is studies as one dimension line while every moment is a zero-dimensional point, details in shorter spans are compressed when studying long term patterns. To circumvent this paradox, time point can be converted into time vector:

\[
t(\frac{n_i}{P_i} \times \cdots \times \frac{n_j}{P_j}) \Rightarrow t^* < \frac{n_i}{P_i}, \cdots, \frac{n_j}{P_j}> \tag{4}
\]
On the left side, time $t$ is expressed as a product of fractions where the denominators $P_i$ are different periods, for day, week, and month are $24 \times 60$, $7$ and $12$ respectively. Numerators $n_i$ are sequences in period $P_i$. The right side is a vector consisting of multipliers on the left. For example, 8:30 on Tuesday in March would be $\left( \frac{510}{1440} \times \frac{2}{7} \times \frac{3}{12} \right) \rightarrow < \frac{510}{1440}, \frac{2}{7}, \frac{3}{12} >$.

In this passage, 2D time vector is used, turning input into 4D vectors (day-week-latitude-longitude). Combined with 3D SOM, this technique empowers us to inspect multiple periods’ behaviors simultaneously. Furthermore, it could reveal influence between temporal periods, for instance, as days elapse in a week, when peaks occur in one day will also change.

3.2. Mixed with Categorical Data

The major difficulty in analysis of mixed data is to find a method determines the distance between variables, and update the map. In this paper, straightforward approach is employed to deal with heterogeneous data, introduced below:

1) Assigning the name strings with ID numbers arbitrarily, since we do not relay on their quantitative meaning, then input $x_i$ becomes a 4D vectors (time-latitude-longitude-ID);

2) Transforming $x_i$’s ID number $j$ into a binary vector $C_{x_i}(j)$, index of column with 1 indicate ID number it holds, $x_i$ now has two parts, 3D numeric vector $N_{x_i}$ and kD category vector $C_{x_i}$:

$$C_{x_i}(j) = < 0, \cdots, \hat{j}, \cdots, 0 >_k$$ (5)

$k$ is number of categories.

3) Total distance $D(x_i, m_j)$ between input $x_i$ and $m_j$ is sum of numeric variables $Dn(N_{x_i}, N_{m_j})$ calculated under Euclidean norm and categorical part $Dc(C_{x_i}, C_{m_j})$ deduced via logic operation $\text{AND}$, if $x_i$, $m_j$ share same ID number, set it to 0, if not, to 1:

$$D(x_i, m_j) = Dn(N_{x_i}, N_{m_j}) + \alpha \times Dc(C_{x_i}, C_{m_j})$$ (6)

$$Dc(C_{x_i}, C_{m_j}) = C_{x_i} \& C_{m_j}$$ (7)
$\alpha$ is a parameter to offset scale dominance of either $Dn(x_{ik}, m_{jk})$ or $Dc(x_{ik}, m_{jk})$ in next step;

4) Searching for BMUs, working out a $m \times k$ weight matrix $W$, $W_{ij}$ represent how category $j$ weighs in nodes $i$. For details on $W$, please view [Appendix B1](#).

5) Applying winner-take-all strategy on $W$, codebook vector $m_i$’s ID variable is assigned with column $j$ that weighs most in row $W_i$.

$$m_i = \text{argmax}_j (W_i(j))$$ (8)

Before utilize it, we should be aware that binary coding process neglects inner structure of categorical variables, which is not suitable category classifications have affiliations[7].

4. Experimental Results

Running the algorithms on two data sets, crime incidents and vehicle collisions, entering data in a single month at one time. In general, performances didn’t vary much in input from different months, and crime incidents in Jan.2015 is selected for illustration, containing 7816 crime reports with date, time, latitude, longitude values and felony names. The CORs are affected by number of cubics dividing original space, for the impartiality of results, CORs are reckoned under a fixed division. Followings are result of $13 \times 8 \times 7$ map, measured under $8 \times 5 \times 5$ cubics in time-latitude-longitude space.

4.1. Performance on Basic Clustering

In different months, overall CORs between frequency matrices of nodes and input fluctuates around 0.82, for CORs of hits and input hold steady above 0.95.

Table 1 shows CORs during eight sections in 24 hours of sample mentioned above.

| Table 1: Coefficients between Input and Map |
|-------------------------------------------|
| Input & Nodes    | 0.94 | 0.43 | 0.84 | 0.72 | 0.96 | 0.63 | 0.95 | 0.64 |
| Input & Hits     | 0.99 | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 | 0.97 |

8
While CORs offer a percentage-like assessment, table 2 provides a quantitative reflection of consistency between map and input. It sums up numbers in different sections.

| Input | 976 | 540 | 729 | 1041 | 1239 | 1380 | 1244 | 667 | $R^2$ |
|-------|-----|-----|-----|------|------|------|------|-----|-------|
| Nodes | 89  | 72  | 73  | 103  | 106  | 121  | 111  | 43  | 0.81  |
| Nits  | 966 | 577 | 663 | 1121 | 1201 | 1387 | 1280 | 621 | 0.98  |

The last column contains the $R^2$ value deduced from linear regression to measure map’s consistence to input. In order to observe in a perceptive manner on how map represent the input, heat map of input’s density is painted with nodes in same section.

![Daily Crime Incident Map in Jan.2015](image)

In Figure 2, two distinguishable intervals are selected to show how nodes match with the density of input.
4.2. Performance on Multi-Period Analysis

2D time vector $<\frac{\text{time}}{24\times 60}, \frac{\text{day}}{7}>$ is used in studying daily and weekly behaviors of crime incidents. Assessing the reliability with projections into subspaces of day-latitude-longitude and week-latitude-longitude. Extra test was conducted on vehicle collisions, result is supplied in Appendix C1.

In study the daily pattern, Table 3 represents CORs in eight sections of 24 hours.

| Input & Nodes | 0.74 | 0.11 | 0.89 | 0.61 | 0.76 | 0.96 | 0.68 | 0.53 |
|---------------|------|------|------|------|------|------|------|------|
| Inputs & Hits  | 0.97 | 0.97 | 0.95 | 0.98 | 0.99 | 0.96 | 0.97 | 0.98 |

Sum up numbers in each section and apply linear regression to obtain $R^2$ value.

| Input | 976 | 540 | 729 | 1041 | 1239 | 1380 | 1244 | 667 | $R^2$ |
|-------|-----|-----|-----|------|------|------|------|-----|-------|
| Nodes | 85  | 62  | 68  | 102  | 102  | 135  | 111  | 63  | 0.92  |
| Hits  | 942 | 582 | 696 | 1100 | 1162 | 1456 | 1183 | 695 | 0.96  |

In week-latitude-longitude subspace, CORs are measured day by day.

| Day in Week | Mon. | Tue. | Wed. | Thur. | Fri. | Sat. | Sun. |
|-------------|------|------|------|-------|------|------|------|
| Input & Nodes | 0.97 | 0.91 | 0.92 | 0.67  | 0.93 | 0.54 | 0.94 |
| Inputs & Hits | 0.99 | 0.99 | 0.99 | 0.98  | 0.99 | 0.99 | 0.99 |

Then sum the numbers in each day of week, and calculate weekly $R^2$ value.
Table 6: Weekly Sum of Input and Map

| Day in Week | Mon. | Tue. | Wed. | Thur. | Fri. | Sat. | Sun. | $R^2$ |
|-------------|------|------|------|-------|------|------|------|-------|
| Input       | 1412 | 1383 | 1184 | 921   | 1064 | 913  | 939  | 0.89  |
| Nodes       | 111  | 112  | 105  | 101   | 105  | 98   | 96   | 0.99  |
| Hits        | 1417 | 1380 | 1181 | 918   | 1068 | 890  | 962  |       |

Distribution on Tuesday and Thursday are selected to show how the map trace weekly variance.

As mentioned, projections into day-week plane suggest temporal elements are interrelated, that behavior in 24 hours is affected by which day it is in week, where COR between input and map reaches 0.64. Figure 4 is heat map of incidents in day-week plane.
4.3. Performance on Heterogeneous Input

The modified 3D SOM is tested with crime incidents data including felony types. Likewise, performance is gauged by calculate CORs under controlled felony IDs or fixed temporal intervals. To avoid monotony in representation, only results on clustering different IDs are provided here. Felony classification ID and temporal-spatial clustering of all input is supplied in Appendix C2.

Table 7: CORs of Different Felonies

| ID Number | 1     | 2     | 3     | 4     | 5     | 6     | 7     |
|-----------|-------|-------|-------|-------|-------|-------|-------|
| Nodes     | -0.22 | 0.94  | 0.54  | 0.96  | 0.80  | 0.17  | None  |
| Hits      | 0.06  | 0.97  | 0.98  | 0.99  | 0.98  | 0.92  | None  |
In same manner, sum up numbers in each type of crime and calculate $R^2$.

Table 8: Numbers in Each ID

| ID Number | 1   | 2   | 3   | 4   | 5   | 6   | 7 | $R^2$ |
|-----------|-----|-----|-----|-----|-----|-----|---|-------|
| Input Data| 94  | 1174| 1374| 3188| 1438| 510 | 38| 0.98  |
| Layer Nodes| 5   | 125 | 139 | 312 | 104 | 43  | 0 | 0.98  |
| SOM Hits  | 94  | 1182| 1383| 3204| 1441| 512 | 0 | 0.99  |

Picture below shows heat maps of two selected types of crime and distribution of nodes with same ID.

![Figure 5: Distribution of Type 4 and 6 Felony](image)

5. Discussion

Generally, we could observe that CORs between nodes and input fluctuate more conspicuously, compared to that of hits and input. It can be explained by the fact that natural positions of layers along time axis are not always consistent to fixed equidistant division, sometimes it gave low CORs misleadingly, which is a limitation of this...
reliability measurement. Broader spans are set to reduce such side effect.

However, regardless of such drawback, 3D SOM performs well in basic clustering, for the CORs between hits and input remain at a high level, which are not troubled by such limitation. And the number of nodes matches with number of input in each layer. And according to Figure 2 and 3, nodes’ distribution is consistent to input’s density.

In multi-period analysis, results confirmed that 3D SOM could trace both daily and weekly pattern effectively. Beyond that, when projected into day-week plane, map discovers the correlation between two periods themselves. From Figure 4, it can be seen when crime incidents peaked changed as day passing by and nodes of map moved accordingly. It provides a new basis for multi-period analysis, that if split the nesting interval into matrix, much more information can be gathered together without loss of short terms details.

When it comes to categorical data, if we focus how the performance in reflecting each type, COR for each ID is strongly influenced by how many input belongs to that ID, for major types map fit well with input. However, because of the winner-take-all strategy, hardly any node is allocated to infrequent crimes with ID 1, 7. Then I tried updating method based on the probability of each ID in weight matrix W, yet it fails to solve the problem and lead to decreased reliability. Detail on this method is supplied in Appendix B2. Linear regression analysis shows that it’s almost inevitable due to contrast between sparsity of nodes and density of input.

While the results are promising in basic clustering, as the dimension increases, it’s troublesome to judge whether a centroid locate at a namely right position. To be more concrete, when studying crime incidents, should the nodes move closer to input with same type of crime, or to data with less geographic distance? When adding ID variables into clustering, CORs in temporal-spatial subspace decreased. Moreover, there are some unsolved theoretical problem in self-organizing map itself, prohibiting us to employ an universal standard to evaluate results.

Besides, this paper concentrates on patterns and relationship within the data, it’s possible that temporal-spatial distributions are caused by factors not included, such as population, police force and so forth, leaving room for research to find potential causality.
Conclusion

Three-dimensional self organizing map is competent and versatile in clustering and identifying behavioral patterns. Framework of grid reflects topographic feature of input, while codebook vectors store detailed information. Facilitated with an innovative methodology that expands timeline into matrix, 3D SOM unravels pattern and inter-relation masked in multiple periods. Besides, a tailored 3D SOM clarifies complex relations in heterogeneous data, wisely avoiding calculation without analytical meaning.

In brief, what this paper seeks to address is not one or two specific clusterings of data contains time and geographic elements, but interpretation of interior relationships between variables based on temporal-spatial distribution. Self-organizing map is constructive for comprehension of high dimensional structures, providing an operable tool in functional utilization.

Appendix

Appendix A

Appendix A1

Significance of Pre-Processing Input

Results shows that rescaling process can significantly reduce both the quantization and topographic error under the same training epochs. Their definitions are listed:

Rescaling the data to $[0,1]$.

$$x_i^* = \frac{x_i - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

(9)

Standardization with z-score.

$$x_i^* = \frac{x_i - \mu}{\sigma}$$

(10)

$\mu$ is the mean of $x$, and $\sigma$ represents standard deviation.

Following table contains QE and TE for data used in basic clustering trained with $10 \times 6 \times 6$ map after 100 epochs.
Table 9: QE and TE of different types of normalization

| Error Type | Raw Input | Rescaling | Z-score |
|------------|-----------|-----------|---------|
| QE         | 0.0235    | 0.0251    | 0.0273  |
| TE         | 0.4271    | 0.3595    | 0.3898  |

Judging from the map plotted with different color representing each layer, normalized map meets less overlaps. Together with quantitative measurement above, we can conclude that normalization contributes to quality enhancement.

Figure 6: Influence of Normalization

Appendix A2
 Initialization of Map

Initialization process can be classified into random and linear initialization. While 2D SOM is known for being insensitive to different initialization, however for 3D SOM, if the codebook vector is initialized randomly, its topographic feature will remain chaotic with high error rate, which might be explained by metastability [10].
Set map’s size as $l \times m \times n$, where $l$ is the number of layers along time axis and $m \times n$ is nodes in each layer. If the $\max(l, m, n) \neq l$, map will rotate in the training, eventually the largest side of the 3D grid approximately parallels to time axis. The expected layered structure is lost, accompanied with high QE and TE. This results is consistent with the behavior of 2D SOM, where the two-dimensional grid extend itself in time-latitude or time-longitude plane, instead of latitude-longitude. It can be explained by Principle Component Analysis (PCA). It’s found that time factor contributed the most to principle components. Therefore, in order to obtain less distortion and well organized layers, the size of grid should be chosen according to PCA, set the largest length to axis with greatest explaining ability.

Appendix B

Appendix B1

Algorithms for Categorical Variable

The updating of categorical variable is fully explained by following equation, inspired matrix calculations in SOM Toolbox\footnote{11}.

$$W_{m \times k} = M_{m \times m} \times (F_{m \times n} \times C_{n \times k})$$

The $m, n, k$ represent the number of nodes, inputs, and categories respectively.

- Weight matrix $W$, $w_{i,j}$ indicate the weight of category $j$ in node $i$.

- Neighborhood matrix $H$ measures the influence of node $i$ to node $j$. $H_{i,j}$ is calculated through Gaussian function $h(i, j)$.

- Filter matrix $F$, indicating BMUs of input. If 1 appears in row $i$, column $j$, it means input $i$’s BMU is node $j$.

- Input ID matrix $C$, coding the categorical data into binaries, for row $i$, if column $j$ is 1, then ID variable in input $i$ is $j$.

- The product of two matrices in parentheses indicates node $i$ has $(F \times C)_{i,j}$ hits with category $j$. 

17
Appendix B2

Choose ID Variable in Probabilistic Manner

In probability based method for updating $m_i$’s ID variable, any ID values can be selected according to their probabilities.

\[
P(m_i = j) = P_{W_i}(j) = \frac{\sum_{t} h_c n_j}{\sum_{t} h_c}
\]

While it seems to be more considerate, results tells another story, that probability based method is affected by the negative impact of random numbers. Other research advices a combination of two method [12], setting a threshold on minimum proportion for winner-take-all strategy, if it’s not reached, selecting ID randomly.

Appendix C

Appendix C1

Additional Test on Multi-Period Analysis

Following is result of clustering 13717 vehicle collision reports with a $13 \times 8 \times 7$ map trained after 100 epochs. First, performance in one day is measured.

|CORs between Input and Map in Days|
|---|
|Input & Nodes | 0.73 | 0.86 | 0.93 | 0.92 | 0.97 | 0.94 | 0.87 | 0.23 |
|Inputs & Hits | 0.94 | 0.92 | 0.99 | 0.99 | 0.99 | 0.98 | 0.98 | 0.93 |

Then, using same measurement on performance in one week.

---

\[1\text{Jan.2015}\]
Table 11: Daily Sum of Input and Map

|       | 731  | 510  | 2060 | 2481 | 2812 | 2650 | 1776 | 693  |
|-------|------|------|------|------|------|------|------|------|
| Input |      |      |      |      |      |      |      |      |
| Hits  | 738  | 480  | 2193 | 2463 | 2754 | 2670 | 1717 | 698  |
| Nodes | 45   | 35   | 109  | 127  | 143  | 137  | 92   | 40   |

Table 12: Coefficients between Input and Map in Week

| Day in Week | Mon. | Tue. | Wed. | Thur. | Fri. | Sat. | Sun. |
|-------------|------|------|------|-------|------|------|------|
| Nodes       | 0.72 | 0.91 | 0.87 | 0.86  | 0.86 | 0.69 | 0.78 |
| Hits        | 0.98 | 0.98 | 0.98 | 0.98  | 0.98 | 0.97 | 0.98 |

Table 13: Weekly Sum of Input and Map

| Day in Week | Mon. | Tue. | Wed. | Thur. | Fri. | Sat. | Sun. |
|-------------|------|------|------|-------|------|------|------|
| Input Data  | 1739 | 1743 | 1713 | 2178  | 2555 | 2039 | 1746 |
| SOM Hits    | 1731 | 1749 | 1726 | 2176  | 2546 | 1998 | 1787 |
| SOM Nodes   | 96   | 105  | 105  | 106   | 113  | 114  | 89   |

Appendix C2

Details in Performance on Heterogeneous Input

Following table is classification of crimes with IDs.

Table 14: ID Numbers of Felonies

| ID Number | Offense Type               |
|-----------|----------------------------|
| 1         | Rape                       |
| 2         | Burglary                   |
| 3         | Felony Assault             |
| 4         | Grand Larceny              |
| 5         | Robbery                    |
| 6         | Grand Larceny of Motor Vehicle |
| 7         | Murder Non-Negl.Manslaughter |

Table 15 and 16 supplied clustering of temporal-spatial distribution of all crimes.
Table 15: Coefficients between Input and Map in Day

| Input & Nodes | -0.14 | 0.87 | 0.73 | 0.73 | 0.96 | 0.93 | 0.86 | 0.47 |
| Inputs & Hits | 0.98  | 0.95 | 0.90 | 0.97 | 0.98 | 0.99 | 0.97 | 0.94 |

Table 16: Daily Sum of Input and Map

| Input  | 976  | 540  | 729  | 1041 | 1239 | 1380 | 1244 | 667  |
| Nodes  | 64   | 54   | 89   | 109  | 124  | 136  | 110  | 42   |
| Hits   | 917  | 552  | 851  | 1064 | 1380 | 1365 | 1257 | 641  |
| R²     | 0.77 | 0.95 | 0.90 | 0.97 | 0.98 | 0.99 | 0.97 | 0.94 |

References

[1] NYC-TIC, Uber request, [https://github.com/fivethirtyeight/uber-tlc-foil-response](https://github.com/fivethirtyeight/uber-tlc-foil-response) (2015).

[2] NYC-OpenData, Nypd 7 major felony incident map, [https://data.cityofnewyork.us/Public-Safety/NYPD-7-Major-Felony-Incident-Map](https://data.cityofnewyork.us/Public-Safety/NYPD-7-Major-Felony-Incident-Map) (2016).

[3] NYC-OpenData, Nypd motor vehicle collisions, [https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95](https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95) (2015).

[4] T. Kohonen, The self-organizing map, Proceedings of the IEEE 78 (9) (1990) 1464–1480.

[5] P. Demartines, F. Blayo, Kohonen self-organizing maps: Is the normalization necessary?, Complex Systems 6 (2) (1992) 105–123.
[6] T. Kohonen, Self-organizing maps, springer series in information sciences 30 (2001a) 138–140.

[7] C.-C. Hsu, S.-H. Lin, Visualized analysis of mixed numeric and categorical data via extended self-organizing map, Neural Networks and Learning Systems, IEEE Transactions on 23 (1) (2012) 72–86.

[8] J. Rynksiewicz, Self-organizing map algorithm and distortion measure, Neural networks 19 (6) (2006) 830–837.

[9] T. Kohonen, Self-organizing maps, springer series in information sciences 30 (2001b) 146–148.

[10] E. Erwin, K. Obermayer, K. Schulten, Self-organizing maps: Stationary states, metastability and convergence rate, Biological Cybernetics 67 (1) (1992) 35–45.

[11] J. Vesanto, J. Himberg, E. Alhoniemi, J. Parhankangas, et al., Self-organizing map in matlab: the som toolbox, in: Proceedings of the Matlab DSP conference, Vol. 99, 1999, pp. 16–17.

[12] N. Chen, N. C. Marques, An extension of self-organizing maps to categorical data, in: Progress in Artificial Intelligence, Springer, 2005, pp. 304–313.