Chapter 6

Data Science and Data Visualization

Michalis Xyntarakis* and Constantinos Antoniou†
*Cambridge Systematics, Medford, MA, United States, †Department of Civil, Geo and Environmental Engineering, Technical University of Munich, Munich, Germany

Chapter Outline

1 INTRODUCTION

Data analysis and data visualization are very important tools for engineers, analysts, policy-makers, and decision makers. Developed originally for "small data," these techniques have been met with varied success in the past centuries and decades. There are a few famous visualizations that are very effective in capturing the essence of the data, and of course there are many infamous negative examples of poor visualizations. Even well-known publications, and researchers, often provide visualizations with considerable faults. Usually, a successful figure requires a lot of work, customization, attention to detail, and refinements.

This leads naturally to the question of visualization quality. What is a good visualization? Many will say an eye-catching one, a colorful one. Others will say one that is accessible to color-blind or sight-impaired individuals. Others

Mobility Patterns, Big Data and Transport Analytics. https://doi.org/10.1016/B978-0-12-812970-8.00006-3

will quote the data-to-ink ratio, calculate the number of information elements in the figure and so on. In the context of big-data, additional constraints emerge, as the amount of data to be plotted is very large. Therefore, in order to be able to convey meaningful messages, one needs to resort to preprocessing techniques, in order to extract some meaningful structure from the data. Therefore, with big data, the process of visualization becomes inherently entangled with data analytics.

Another important concept relating to visualization is beauty. Steele and Iliinsky (2010) describe beauty in this context as having four components; besides being aesthetically pleasing, it must also be novel, informative, and efficient. The authors identify Mendeleev's Periodic Table of Elements (Fig. 1) and the Harry Beck's London Metro Map as two historic visualization examples that satisfy these rules. The periodic table was a novel and efficient representation of dense information (providing up to nine pieces of information per item), and in its early versions, it did not include color (the figure could thus be produced in a typewriter). This stresses the point that "strong graphic design treatment is not a requirement for beauty" (Steele and Iliinksy, 2010). The London Map uses visual conventions and standards, but does not aim to be geographically accurate. Instead, it strips away unnecessary information, and focuses onto an abstract visual style that provides exceptional clarity.

It is therefore clear that visualization is part science and part art, and also that it is difficult, even when dealing with "small" data. The emergence of data-collection technologies, accompanied by the explosion of user-generated content, has led to an abundance of data. In the process, visualization has become both more important and more challenging. The challenging part is easily understood. The more important has to do with the fact that as data become larger, then extracting meaningful relationships from them becomes harder. Visualization of big data is not merely a process of showing data in the best way, but usually often involves a certain degree of actual analysis or modeling, such as clustering, data-mining, or data reduction.

In this chapter, we use a number of mobility-related data sets to demonstrate some of the state-of-the-art data visualization techniques.

A small data set with passenger car characteristics from the 70s and 80s is used to showcase parallel coordinates and the rest of the dimensionality reduction techniques (Figs. 2–6 and 10). Variables that describe the cars include miles per gallon (mpg), number of cylinders, displacement (cc), power (hp), weight (lb), time to reach 60 miles per hour, and year of construction. The data set contains information for 408 cars and be downloaded from https://bl.ocks.org/jasondavies/1341281 (Parallel Coordinates, 2018)

The corridor congestion data set was derived from the National Performance Management Research data set that was purchased by United States Federal Highway Administration (FHWA). The derived data set contains link travel times aggregated every 5 min for network links on the northbound direction of US Interstate I95 between Washington D.C. and Baltimore. The distance

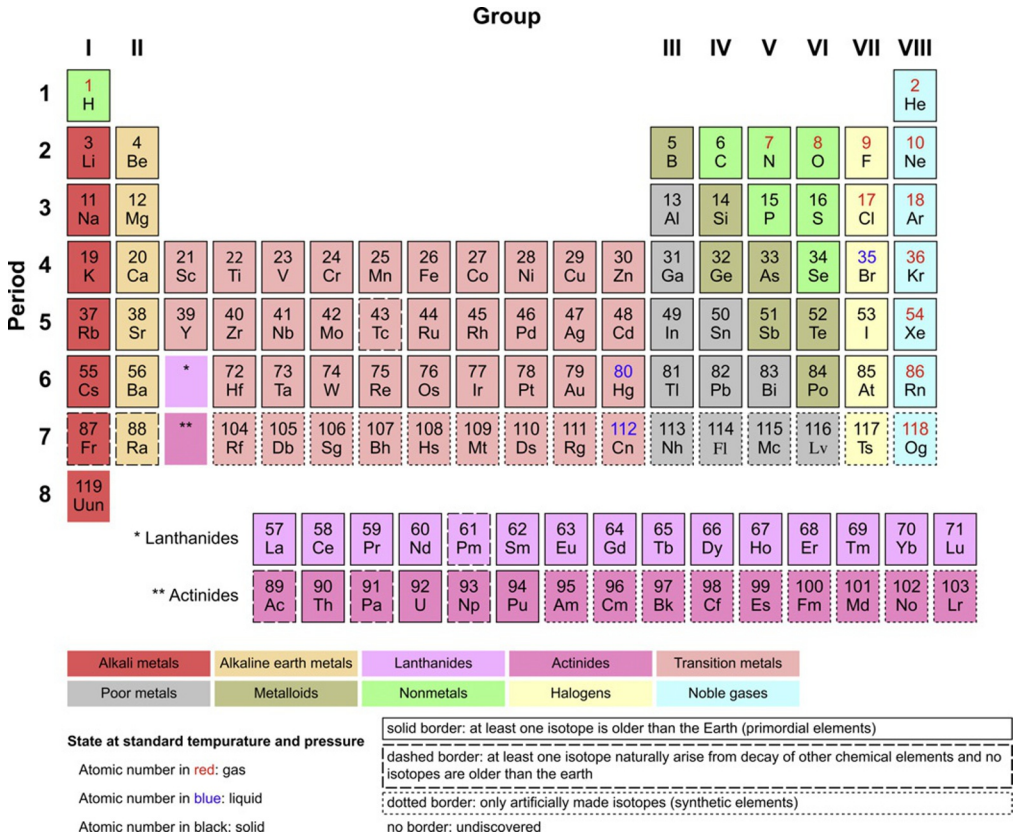

FIG. 1 Periodic table. *((By Armtuk—Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=2010645.))*

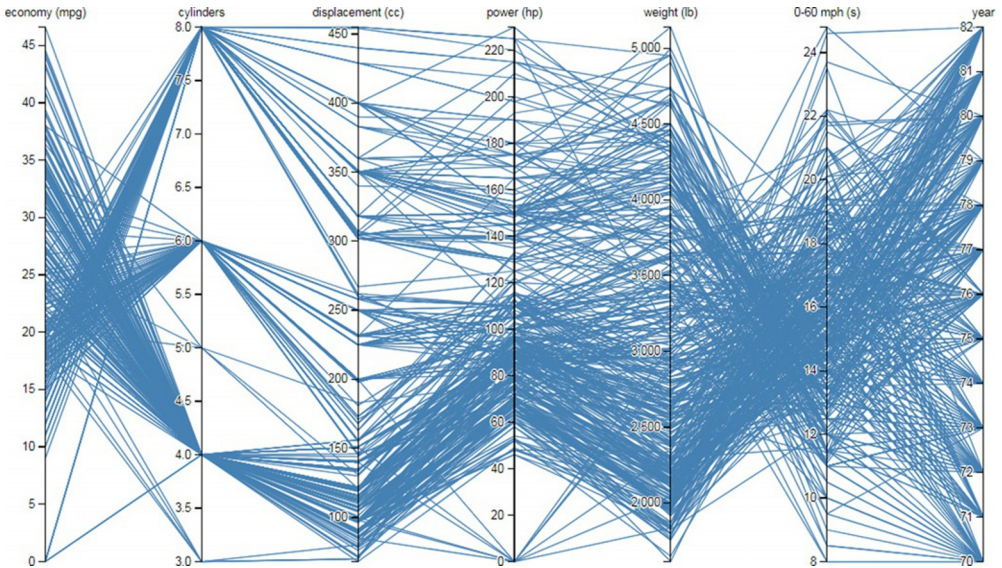

FIG. 2 Parallel coordinates plot.

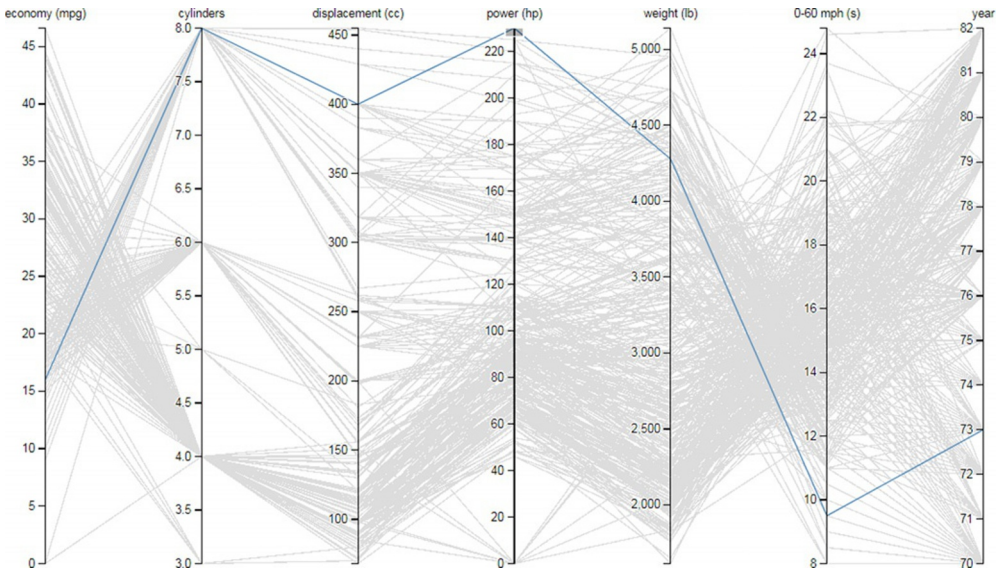

FIG. 3 Parallel coordinates plot: selected observation.

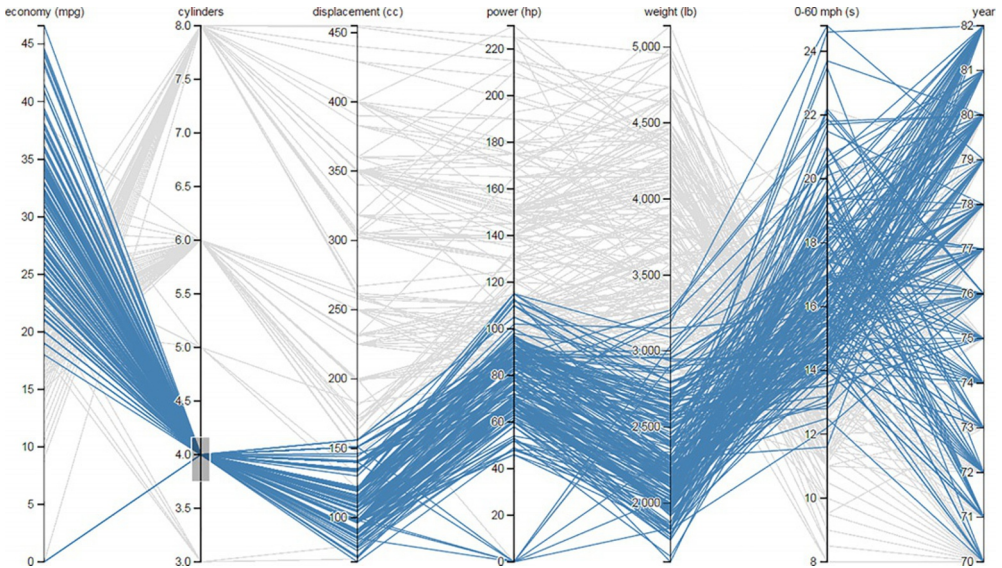

FIG. 4 Parallel coordinates plot: simple selection query.

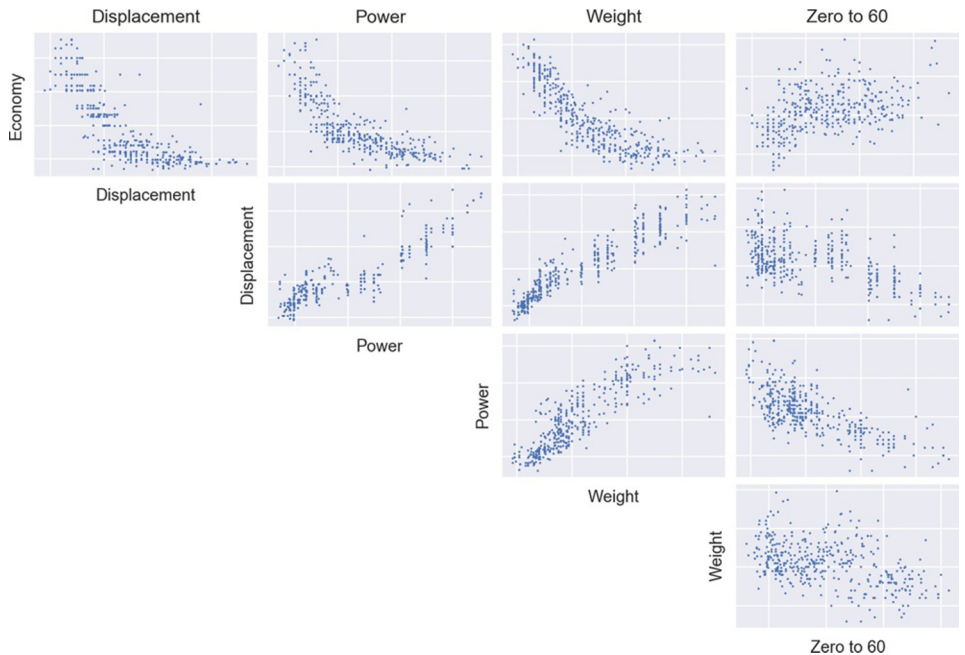

FIG. 5 Scatterplot matrix of the pairwise relationships in the car data set.

FIG. 6 t-SNE versus parallel coordinates on the car data set.

between the two beltways is 24.5 miles and it is covered by 26 links about one mile each. A feature in this data set consists of a matrix of 5-min speeds from 3 PM to 8 PM that pertain to a given day. The top-left graph of Fig. 14 shows the distribution of average corridor speeds in 2017. Fig. 7 uses a small multiple display to show each of these matrices for all the weekdays between March 27, 2017 and April 21, 2017.

The NYC Taxi and limousine commission publishes taxi trip records since January 2009. The NYC taxi data set contains a record for every ride a yellow taxi undertook between June 30, 2016 and June 30, 2017. The origins and the destinations of each of the trips are provided at the zone level. For each day in the data set, a variable is generated that contains the taxi trips from zone i to zone j for a given time period. Most of the taxi trips belong to a small percentage of zone pairs. Fig. 18 shows that 5000 zone pairs account for almost 80% of the total number of trips in the New York region. This is a publicly available data set that can be downloaded from http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.

The New York Metropolitan Transit Administration (MTA) turnstile data set contains turnstile counts for every turnstile and station in NYC from June 2010 to August 2017. The extended time period of reporting can be used to visualize operations during extreme events, such as hurricane Sandy (Fig. 8) or visualize ridership trends. Fig. 9 shows turnstile entry counts aggregated for every day of the year. The raw data set that is publicly available contains 2646 days and 466 stations. After eliminating days and stations that have a significant number of missing values or implausible measurements, the test data set contains 2364 days and 266 stations. In the test data set, an observation is a day and a count station measurement is a column. The data set can be downloaded from http://web.mta.info/developers/turnstile.html.

Naturally, the approaches demonstrated in this chapter are not the only techniques available, nor is it implied that they are the most suitable for each task. The reader should be able to adapt the background and insight obtained from reading this chapter, to seek and apply the most suitable techniques. All the data sets except from the corridor congestion one are publicly available online. Computational notebooks that replicate the analysis are available in the companion website.

The remainder of this chapter is structured as follows. Section 2 provides a quick review of structured visualization techniques. Section 3 provides an overview of the multidimensional data visualization techniques that are considered in this chapter. Section 4 presents the results of these analyses, starting from the experimental setup that is being employed to demonstrate these algorithms using the considered data sets. Section 5 provides some concluding remarks.

2 STRUCTURED VISUALIZATION

The literature about what constitutes good visualization is long and ranges from individual researchers to stakeholders (such as the European Environmental Agency, 2018). While many of these guidelines and suggestions seem obvious, they are often violated, not only by students, but also by more experienced researchers, journalists, and other graphics creators. For example, quite often figures have illegible elements (axis labels, legends, and titles); use color excessively; have distracting elements; and are ill constructed by having misleading

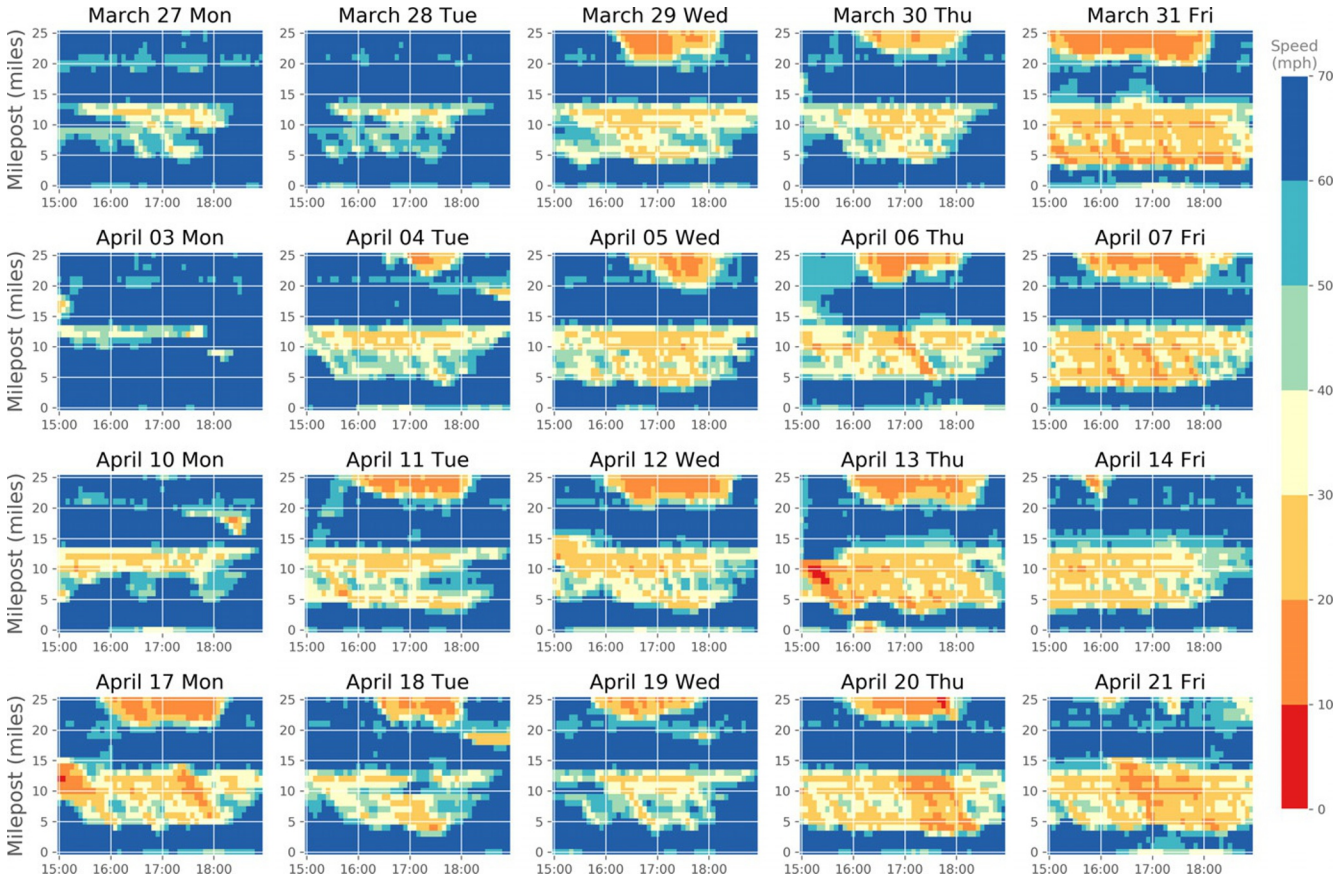

FIG. 7 Corridor link speed heatmaps for I-95 NB between District of Columbia and Baltimore.

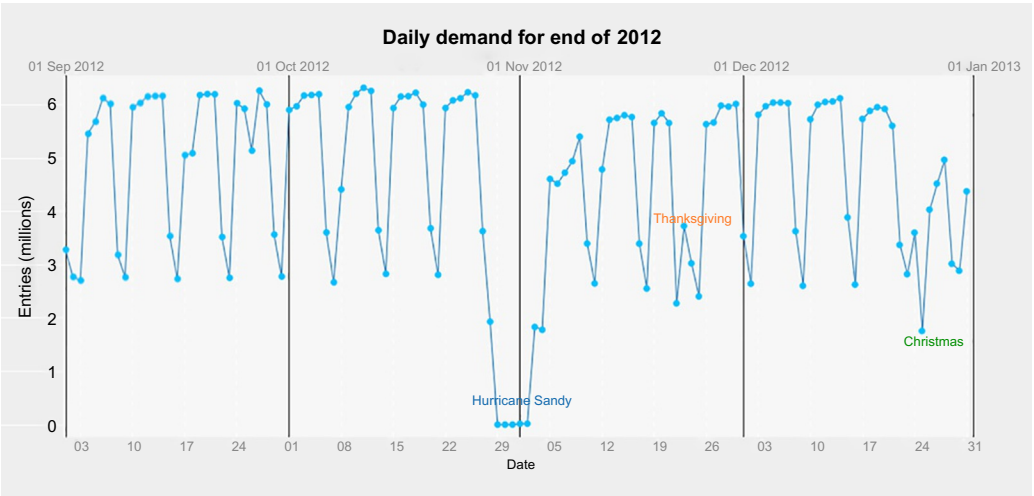

FIG. 8 New York city subway ridership for the end of 2012.

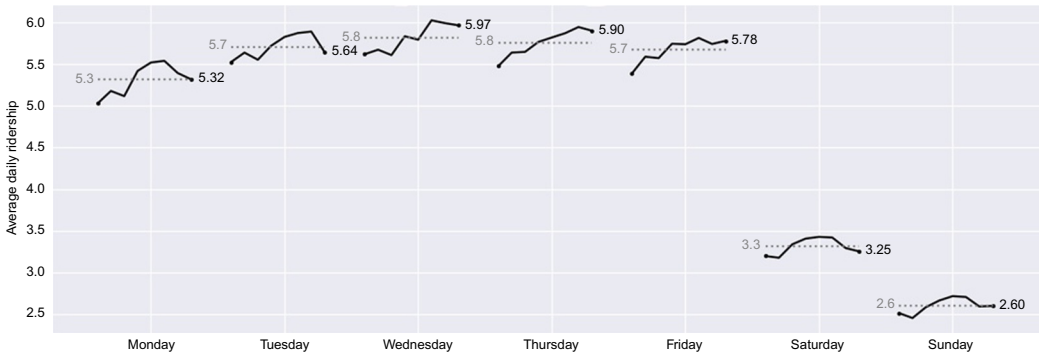

FIG. 9 Changes in average daily subway ridership in New York from 2010 to 2017.

axis ranges or misleading axis labeling (e.g., using equal spacing for unequal ranges).

Bateman et al., 2010 explore the potential value of visual embellishment on the effectiveness of charts, using both insight and focus groups. Contrary to expectations, it appears that visual embellishments are not always detrimental to comprehension, as they sometimes create mental associations for the viewer. Therefore, the choice of whether such approaches should be followed is harder to judge a priori, and it depends on the intended audience and specific circumstances.

Creating good graphics is not a new field. For example, Edward Tufte in his popular books (Tufte, 1983, 1990, 1997) has curated a number of exemplary and ill-constructed visualizations and put forward guidelines on information design. Tukey (1977), Cleveland and Cleveland (1985), and Kosslyn (1994) are seminal works on visualizing statistical information. The ideas and guidance in these seminal references are often simple and straightforward. For example, Tufte's six principles of graphical integrity, as outlined in Tufte (1983), include that representation of numbers should match the true proportions and that labeling should be clear and detailed. However, quite often, ill-constructed visualizations see the light of day, even from respected researchers or leading journalistic institutions. (This has also led to the creation of the term "chart-junk.")

The Grammar of Graphics (Wilkinson, 2006) is a seminal book that organizes information visualization through a series of structured rules: "the grammar of graphics takes us beyond a limited set of charts (words) to an almost unlimited world of graphical forms (statements)." Creating visualizations becomes thus an organic process that focuses on the essence and not the ornaments, although "the rules of graphics grammar are sometimes mathematical and sometimes aesthetic" (Wilkinson, 2006). One way to look at this, is as object-oriented visualization.

Wickham (2010) presents his "layered grammar of graphics", based on these ideas, which has been operationalized into the ggplot2 R package. Wickham (2016) provides a more detailed presentation of the powerful ggplot2 tool. While not the only such initiative, this is one that has been very well-established and allows the generation of very powerful and aesthetically pleasing visualizations.

The previously mentioned principles and ideas are very important and have served the research and practitioner communities for decades. In the era of big data and high-dimensional data, additional challenges emerge. Manual inspection, using traditional data visualization techniques, such as histograms and scatterplots is infeasible. Dimensionality reduction techniques create a two- or three-dimensional map from the high-dimensional data where each data point is a dot in the map. Investigating the different clusters that appear in the two-dimensional map *interactively* can reveal similarity patterns and other

properties that is hard to deduce otherwise. The remainder of this section will briefly introduce multidimensional and high-dimensional visualization techniques and apply them on the four test data sets introduced earlier.

3 MULTIDIMENSIONAL DATA VISUALIZATION TECHNIQUES

Multivariate visualization techniques can be classified into two broad groups (De Oliveira and Levkowitz, 2003). In the first group, additional dimensions in the data set are encoded using different elements, such as multiple axes or visual encodings, such as color and shape texture. To avoid overplotting, the amount of data displayed are usually not more than a few thousand records. The number of dimensions usually does not exceed 10, due to visual cluttering and human perception limitations. Effective multidimensional visualizations include, for example:

- The parallel coordinates plot which can effectively display up to a few thousand records and a dozen variables. This plot is discussed later in this chapter.
- Small multiple displays, such as the scatterplot matrix (Fig. 5) or a heatmap grid (Fig. 7).
- Pixel-oriented visualization techniques that map each data value in a data set to a color-coded pixel on the screen (Keim, 2000).
- Scatterplots or other frequently used two-dimensional graphs in which additional variables are encoded using color, shape, size, and other visual elements in addition to position. For example, gapminder, a visualization that has been viewed millions of times on the web, uses an animated scatter plot that shows the relationship between income and life expectancy for many countries from 1800 to 2018. In the scatterplot, point color represents a country's continent and point size represents population. Animation is used to show how life expectancy changes over time.

In the second broad group, often termed high-dimensional visualization, dimensionality reduction methods, such as principal component analysis or PCA (Jolliffe, 1986) are applied to data sets with dozens, hundreds, or thousands of variables to obtain a lower-dimensional projection that is frequently shown as two-dimensional scatterplot map. For example, the first two principal components of PCA can be visualized in a scatterplot to produce a two-dimensional projection of the full data set regardless of dimensions. Projection methods, such as multidimensional scaling (MDS) or t-SNE, can be applied either on the original data set, if it is computationally feasible, or on a lower-dimension projection obtained by PCA. A thorough comparison between dimensionality reduction techniques is presented by Van Der Maaten et al. (2009) and Bunte et al. (2012), while a more general overview of multidimensional

visualization techniques is given by Liu et al. (2017). Popular projection techniques include

- Principal component analysis (Jolliffe, 1986)
- Multidimensional scaling (Borg and Groenen, 2005)
- Isomap (Tenenbaum et al., 2000)
- Locally linear embedding (Roweis and Saul, 2000)
- t-distributed stochastic neighbor embedding (t-SNE, Van Der Maaten and Hinton, 2008 and 2014)

Projecting the multidimensional space into two dimensions is an ill-posed problem. Different techniques make different assumptions or tradeoffs and may be applicable to different problems. Unlike PCA, some of these techniques are probabilistic and parameterized and may not always yield the same result from the same inputs. For these reasons, a two-dimensional projection should be viewed as an exploratory method that is to be used interactively and possibly in combination with other techniques, such as clustering.

3.1 Parallel Coordinates

Parallel coordinates is a well-researched visualization that allows the analyst to discover predominant multivariate patterns interactively (Wegman, 1990). Parallel coordinates can visualize effectively a few thousand observations, each having up to 10 or 15 variables depending on the data. If judged only as a static plot, the visual display can be seen as cluttered by many spaghetti lines. But when used interactively, parallel coordinates can reveal outliers, clusters, and relationships that can be further clarified using scatterplots or other simpler displays. Parallel coordinates (through their interaction) can help investigate the following:

- Focus interactively on any given observation or subspace. Selection queries are easily generated by brushing.
- Examine the relationship between the variables in the data set. Parallel lines between adjacent axes show positive correlation and intersecting lines a negative correlation. Sliding a selection window is required to uncover relationships between nonadjacent variables.
- Uncover clusters or patterns in the data or in a selected subspace (Artero et al., 2004). This can be made easier if opacity and color are used.

Fig. 2 visualizes the previously described car data set using parallel coordinates. In the figure, each variable takes values in a parallel vertical axis whose scale corresponds to each variable's domain of values. An observation, or a car in this particular case, is represented by a point/coordinate on each of the parallel axes. A line connects all points that belong to a single observation. In Fig. 3, the car that has the highest horsepower, approximately 230 hp, is selected and shown as

a blue line. The rest of the cars/lines are shown in gray. It is easy to inspect the selected car's characteristics in relation to the rest of the data set. For example, the selected car was built in 1973 (last variable on the right); it was one of the heaviest and at the same time one of the fastest in the data set.

Relationships between adjacent variables are revealed by studying how lines are arranged between the two adjacent axes. If the lines are parallel to each other, regardless of slope, then there is a positive correlation between the two variables. In Fig. 4, this is the relationship between displacement and power for the selected four-cylinder cars. Negative correlation is shown as lines that cross each other. For example, in Fig. 4, this is the relationship between fuel economy and the number of cylinders. The higher the negative correlation the smaller the area of intersecting lines. Uncorrelated variables in adjacent axes are shown as lines with a mix of crossing angles. In the scatterplot matrix of the car characteristics data set (Fig. 5), the pairwise correlations can be seen more clearly, but a scatterplot matrix cannot visualize as many variables as a parallel coordinates plot.

The power of parallel coordinates relies in interactive record selection and animation. Data selection is achieved by drawing an adjustable rectangle around the desired range on any given axis. By selecting records based on multiple criteria, the relationship between variables in a subspace can be explored. When the user changes the selection interactively by moving the selection window up or down, an informative animation is generated that reveals the relationships between nonadjacent variables.

If lines are color-coded using different colors, and by possibly using opacity data, clusters can be visible as shown in the bottom part of Fig. 6. The range of each variable in this parallel coordinates plot has been normalized between zero and one as explained later in the chapter. Cars have been color-coded, based on the number of cylinders variable (which is otherwise not included in the plot). At least three distinct patters emerge from the display. Cars with a small displacement have low power and weight and are slower, but more fuel-efficient. On the contrary, cars with high displacement have more power and are faster, despite weighting more. A third category of cars, those with six cylinders, lie in between the four- and eight-cylinder cars. Unlike other high-dimensional visualizations, parallel coordinates can visualize side by side categorical, ordinal, and numeric variables. However, nonnumeric variables concentrate many lines through the same point, something that makes the plot harder to comprehend.

A scatterplot matrix is a much more effective way to visualize pairwise relationships, but can become unwieldy, when there are many variables to analyze. If there are n variables to visualize, there are $\frac{n(n-1)}{2}$ unique scatterplots to investigate. For example, for 10 variables, there are 45 scatterplots on the same display to go over. As stated earlier, Fig. 5 shows the scatterplot matrix of the car data set. While pairwise correlations are immediately evident in Fig. 6, the scatterplot matrix does not show the clusters that exist. Parallel coordinates and scatterplot matrices ought to be used complimentary to discover data set structure.

3.2 Multidimensional Scaling (MDS)

Given a matrix of dissimilarities between n observations represented as pairwise distances d_{ij}, MDS (Torgerson, 1952) constructs a representative point y_i in R^d for each observation i in R^D, in such a way that the pairwise distances between the original objects and their representations are maintained. In classical scaling (Borg and Groenen, 2005), the low-dimensional representations of the high-dimensional data are found by minimizing the sum of the square differences between the high-dimensional pairwise distances and the low-dimensional representations. Specifically, the objective function to be minimized is the following:

$$\text{Stress}\,(y_1, y_2, \ldots, y_n) = \left(\sum_{i<j}^{n} \left(d_{ij} - \left\| y_i - y_j \right\| \right)^2 \right)^{1/2}.$$

In classical scaling, the pairwise dissimilarities are Euclidean distances, $d_{ij} = \|x_i - x_j\|$, where x_i are input points in R^D with ($d \ll D$). MDS is particularly suited for high-dimension low sample size problems because it uses an $n \times n$ input distance matrix instead of the $D \times D$ covariance matrix that PCA uses.

MDS focuses on preserving the distances between widely separated data points. Many variations of the Stress function presented earlier have been introduced in the literature (Sammon, 1969; Koch, 2013) that generate nonlinear projections that better preserve the structure of nearby data points in the high-dimensional space.

In this chapter, classical MDS is applied to all the test data sets and it is compared to t-SNE. The Shepard diagram (Borg and Groenen, 2005) provides a qualitative assessment of the effectiveness of MDS. For classical MDS, which is used in this chapter, the Shepard diagram takes the form of a scatterplot of the multidimensional versus the projected distances. A quantitative assessment of fit is given by the Stress value defined in this section.

3.3 t-Distributed Stochastic Neighbor Embedding for High-Dimensional Data Sets (t-SNE)

Linear dimensionality reduction techniques, such as classical MDS, are efficient in presenting the global structure of the data by keeping the representations of very dissimilar data points far apart. However, MDS and other linear dimensionality reduction techniques are not very effective in many applications in which the related high-dimensional data points lie on a nonlinear manifold. Such an example is the projection of handwritten digits into their natural clusters in which MDS and other techniques perform poorly (Van Der Maaten et al., 2009).

Stochastic neighbor embedding represents similarities in both the high-dimensional and projected space as probabilities. In the high-dimensional space, it assumes that a Gaussian distribution is centered at each data point

x_i and the similarity of data point x_i to data point x_j expressed as the probability p_{ij} is equal to

$$p_{i/j} = \frac{\exp\left(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\frac{\left\|x_i - x_k\right\|^2}{2\sigma_i^2}\right)},$$

$$p_{ij} = \left(p_{i/j} + p_{j/i}\right)/2.$$

The variance σ_i depends on the density of the points in the vicinity of x_i and is related to the user-specified perplexity input parameter. Perplexity is a guess from the user's perspective about the number of close neighbors each point has. Recommended values by Van Der Maaten and Hinton (2008) range between 5 and 50. A lower value of perplexity puts more emphasis on local differences and less on global structure. It is recommended that multiple maps are generated with different perplexity values and the results are examined carefully before settling on one perplexity value. Regardless of the perplexity value, t-SNE may not provide a clear picture of the global outliers in the data (Onderwater, 2015) at the expense of revealing structures at many different scales.

In the low-dimensional space, t-SNE uses a Student t-distribution with one degree of freedom to estimate the similarity between map point y_i and y_j represented as q_{ij}.

$$q_{ij} = \frac{\left(1 + \left\|y_i - y_j\right\|^2\right)^{-1}}{\sum_{k \neq l}\left(1 + \left\|y_k - y_l\right\|^2\right)^{-1}}.$$

The map coordinates y_i are calculated by minimizing the Kullback-Leiber divergence between the two distributions p_{ij} and q_{ij}.

$$KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

The fact that t-SNE is nonlinear dimensionality reduction implies that that dense areas will be expanded and sparse areas reduced. Unlike MDS and as it is shown later in this chapter, the resulting distances on the two-dimensional embedding may not be proportional to the point distances in the high-dimensional space. As a result, cluster size in the projected map is not always indicative of variation in the high-dimensional space.

4 CASE STUDIES

4.1 Experimental Setup

In our four experiments, we start by using PCA to examine the dimensionality of the problem. PCA is often used to reduce the dimensionality of the test data

set to a smaller one that results in faster t-SNE and MDS runtimes. However, for the test data sets of this chapter, PCA was not applied to preprocess the data, since t-SNE and MDS runtimes were in the order of minutes on a typical laptop. Each dimensionality reduction technique was used to convert the multidimensional representation to a two-dimensional scatterplot. All the scatterplots are color-coded by some data set property, such as number of cylinders, corridor travel time, or peak period that was not used in dimensionality reduction process. Euclidean distances were used for MDS while a number of different distance functions were tried with t-SNE, including Manhattan, cosine, Mahalanobis, Canberra, braycurtis, and Chebyshev. The Canberra distance followed by Manhattan and Euclidian yielded the most clustered and clear maps. Results are reported for Euclidian and at times for Canberra distances. For the t-SNE, the input perplexity was varied between 5 and 50 and the number of iterations between 500 and 2000 to obtain a stable mapping. In most cases, perplexity was set to 30 and the number iterations to 1000. The Barnes-hut t-SNE implementation of python's scikit-learn library (v0.19.1) was used for all runs. The PCA and MDS implementations of python's scikit-learn library were also used (Scikit-learn, 2018).

The Canberra distance between two vectors x_i and x_j is defined as the weighted Manhattan distance as follows:

$$\text{canberra}_{ij} = \sum_k \frac{|x_{ik} - x_{jk}|}{|x_{ik}| + |x_{jk}|}.$$

4.2 Car Characteristics Data Set

In this small data set, the parallel coordinates plot can be used to visualize all the data and validate the structure MDS and t-SNE reveal. In order to apply any of the dimensionality reduction techniques, the data need to be normalized to a common scale. Otherwise, the Euclidean distance or any other metric will be dominated by the variable that has the higher magnitude. All the variables for the MDS and t-SNE application were standardized based on the following equation:

$$\hat{x}_{ij} = \frac{\hat{x}_{ij} - \min_j}{\max_j - \min_j},$$

where $\min_j$ is the minimum value of feature j and $\max_j$ is the maximum value of the same feature.

Applying PCA to the car data set yields that two principal components can explain more than 90% of the variation in the data set (Table 1).

Fig. 6 shows the t-SNE visualization (with Canberra distances) of the car data set at the top versus the parallel coordinate representation at the bottom. Each line in the parallel coordinates plot has been color-coded by the cylinders variable and the same color-coding has been applied to the t-SNE diagram at the

TABLE 1 PCA Results for Car Data Set

Number of Principal Components	Cumulative Explained Variance Ratio
1	0.837
2	0.928
3	0.972
4	0.989
5	1.000

top. t-SNE shows cars with different number of cylinders belonging to different clusters. Cars with eight, six, and four cylinders are clearly separated from each other.

MDS produces a linear projection of the car data set that does not separate cars based on cylinders. Fig. 10 on the top row shows the first two principal components of PCA as a scatterplot on the left and the linear MDS projection on the right. Eight-cylinder cars are separated from the rest, but six and four-cylinder cars are shown without a gap between them. Clusters are also less compact than t-SNE's clusters. In the car data set, MDS is able to project the five-dimensional space to a two-dimensional one, without distorting the pairwise distances. The Shepard diagram for the MDS projection, shown at the bottom-left corner of Fig. 10, shows the vast majority of pairwise distances (d_{ij}^{D} vs. d_{ij}^{d}) equal to each other and along the diagonal. The minor dispersion along the diagonal for smaller-to-medium distances can be attributed to the lower contribution these distances make compared to the higher distances in the MDS stress function. In contrast, the same diagram for the Euclidean version of t-SNE on the car data set shows the nonlinear nature of t-SNE.

Changing the metric used for calculating the pairwise distances in the high-dimensional space can have a significant impact on the structure of the two-dimensional projection. In our test data sets, the choice of metric has a significant impact on the arrangement of the relative placement of clusters on the projected space on the formation of the clusters. Overall, the Canberra distance provided moderately better cluster separation than the Euclidean distance although additional research is required to identify which is better in the test data sets.

Fig. 11 compares the Canberra and Euclidean distances on the car data set. The resulting map projections at the top of Fig. 11 are similar, but the Canberra map identifies two clusters in the four-cylinder cars instead of one. The bottom two charts in Fig. 11 visualize the matrices of pairwise distances d_{ij}^{D} between cars i and j for the two distance metrics. The order of the cars in each matrix has been arranged by the x- or y-coordinate of the corresponding projection to

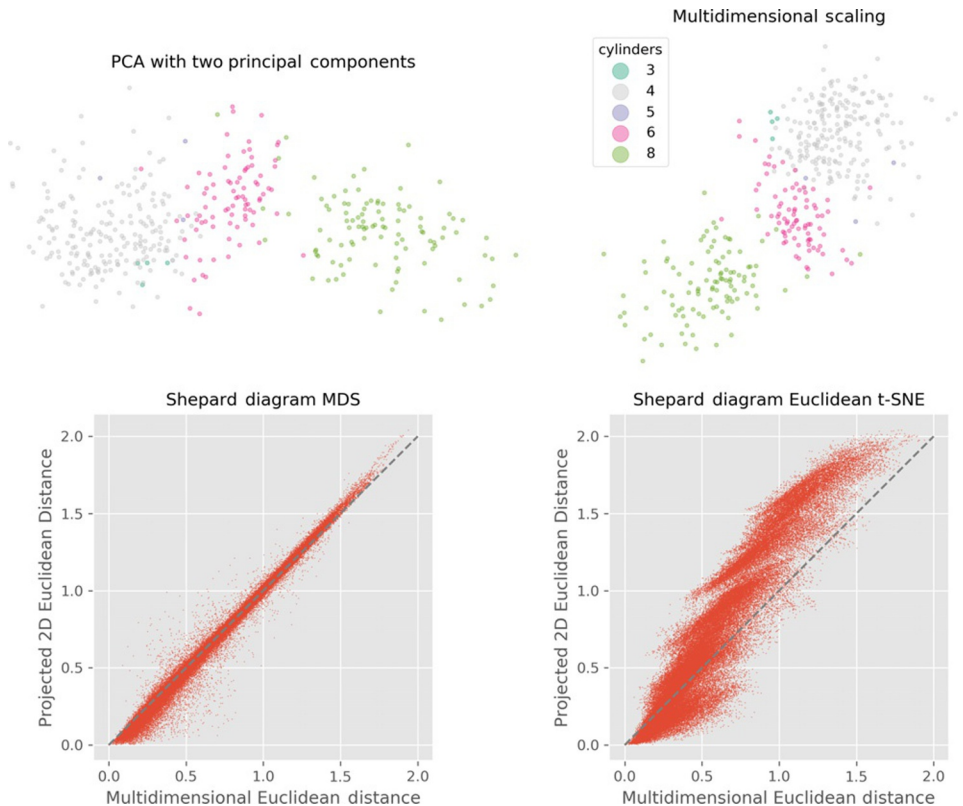

FIG. 10 Dimensionality reduction methods on the car data set.

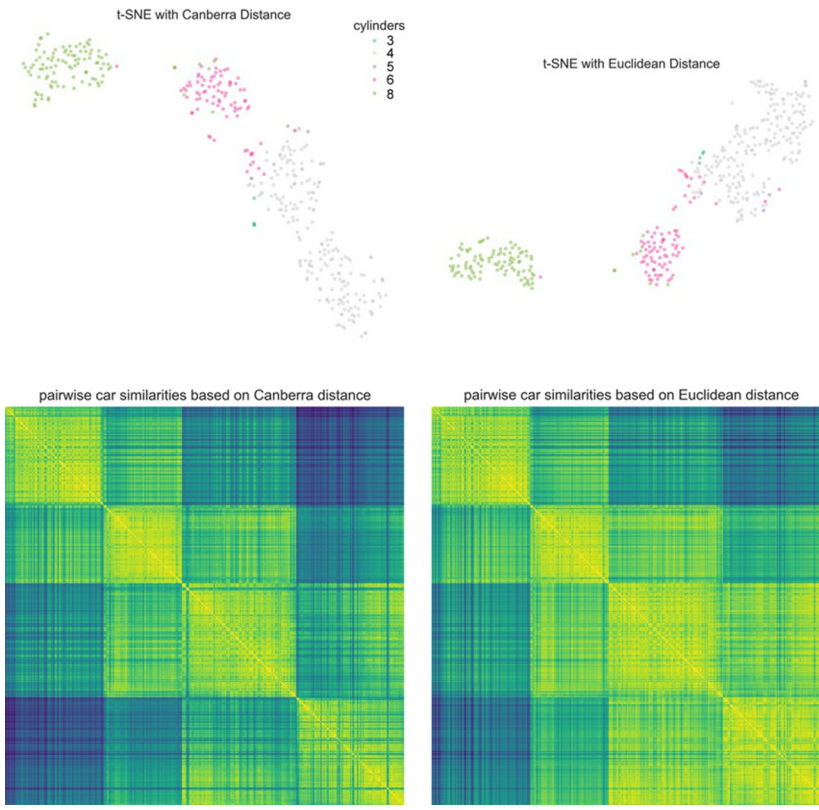

FIG. 11 t-SNE using the Canberra and Euclidean distance metrics on the car data set.

arrange the cars in the same clusters shown on the top row. For example, the bottom-left matrix in Fig. 11 has its elements sorted by the x-coordinate of the Canberra t-SNE plot shown on the top-left, so that eight-cylinder cars are followed by six- and four-cylinder cars. A linear color mapping (Nunez et al., 2017) has been used to map each distance domain to the viridis color scale which associates yellow with short distances and dark green with long ones. A cluster is represented as a rectangle of yellow points along the diagonal. Overall, cluster separation is more distinct, when using the Canberra, as opposed to Euclidean distance.

4.3 Congestion on I95

Trying to find similar congestion patterns on I-95 Northbound between the DC and Baltimore beltways is similar to visualizing handwritten digits to their natural clusters. In Van Der Maaten and Hinton (2008), each handwritten digit is represented by a 28×28 matrix of pixels representing grayscale values from

TABLE 2 PCA Results for the I95 Data Set

Number of Principal Components	Cumulative Explained Variance Ratio
1	0.474
2	0.537
3	0.593
4	0.622
5	0.649
19	0.805
29	0.853
47	0.902
85	0.950

0 to 255. Similarly, each congestion heatmap pattern in Fig. 7 is represented by a matrix of 26×48 speed values ranging from 1 to 70 mph. For the analysis, each congestion heatmap matrix for a given day of the year is converted to a one-dimensional array by appending one row after the other. The data set to be analyzed is thus 365 rows by 1248 columns.

Table 2 shows the explained variance ratio as a function of the number of principal components. In the handwritten digit data set used by Van Der Maaten and Hinton (2008), 86 principal components are required to represent 90% of the information in the data set. Congestion pattern heatmaps have a shape that is significantly simpler than handwritten digits. Nevertheless, 47 components are required to represent 90% of the variability in the data set.

Fig. 12 shows the two-dimensional MDS projection of the I95 congestion pattern data set. Each point represents the congestion heatmap of a given day. The points are color-coded based on the average corridor speed in the entire 5-h period of measurement from 3:00 PM to 8:00 PM. Points closer to the bottom of Fig. 12 correspond to low congestion days with speed close to 60 mph. Points at the top of Fig. 12 show high congestion with average speed as low as 17 mph. It is observed that the lower the speed the higher the dispersion of points in the MDS projection. This is probably a manifestation of the variability of congestion causes that result to different congestion heatmaps. Some of the points/days in Fig. 12 are labeled and the corresponding heatmaps are shown in the periphery of the MDS scatterplot.

Fig. 13 shows the t-SNE projection for the corridor congestion data set. Points are arranged along the diagonal of Fig. 13 in a strip rather than a parabola as in Fig. 12. Average corridor speed is the lowest at the top-left of Fig. 13 and it

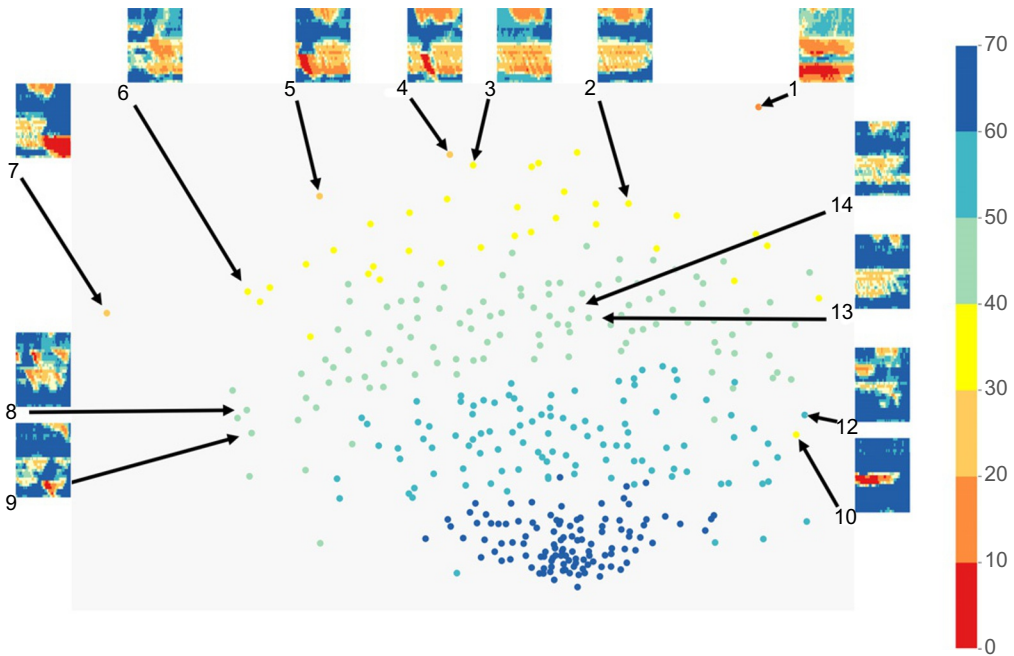

FIG. 12 Multidimensional scaling on the corridor congestion data set.

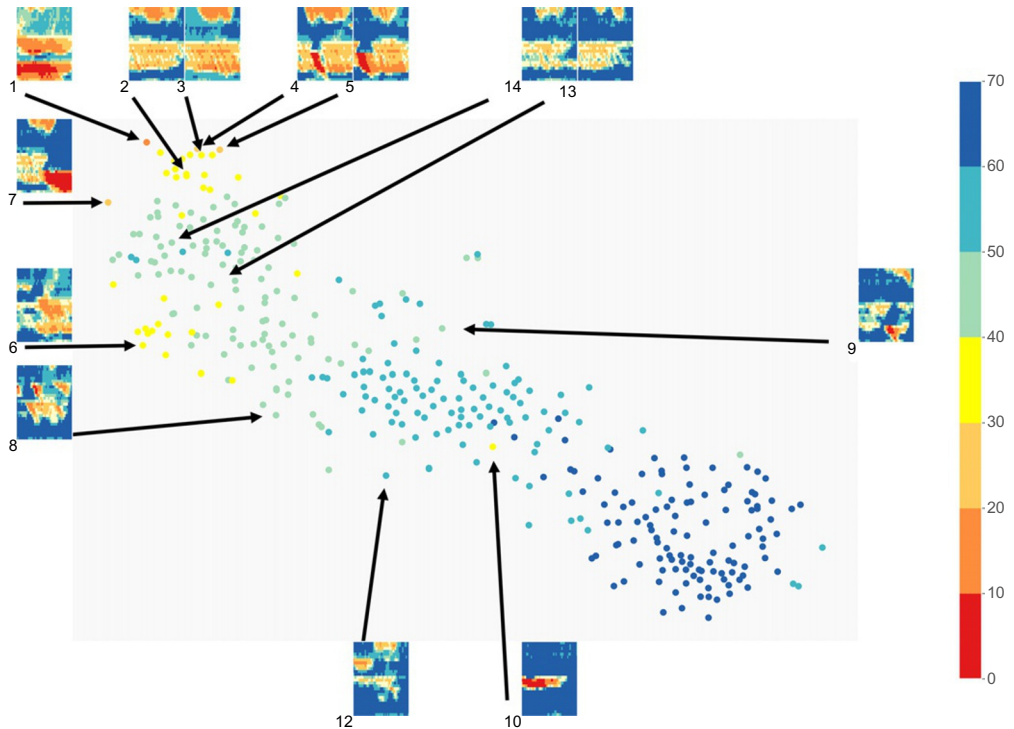

FIG. 13 t-SNE on the corridor congestion data set.

gradually, but loosely, decreases, when moving towards the bottom-right corner. The dispersion of heatmaps with high speed is approximately the same as the dispersion of heatmaps with low average speed, shown in orange or yellow. This is due to the nonlinear nature of t-SNE applying different transformations to different areas. Since the variance σ_i depends on the density of the points in the vicinity of x_i in the definition of pairwise similarities in the high-dimensional space, low-density areas will be contracted and high-density areas will be expanded. Global outliers that are clearly distinguishable in the MDS projection become local outliers in the t-SNE map.

Fig. 14 shows the first two principal components of PCA as a scatterplot on the top-right corner. Qualitatively, the resulting map is similar to MDS in Fig. 12, although there appears to be significantly less dispersion in the PCA projection. The first principal component (the x-axis) appears to be very well correlated to average corridor speed. The bottom row of Fig. 14 shows the Shepard diagrams for the MDS and t-SNE projections. Each point corresponds to a pair ij of observations or heatmaps. The sigma shape of the Shepard diagram for MDS reveals that small distances in the high-dimensional space become even smaller in the projection. At the same time, medium-to-large distances have a tendency to become larger. Using MDS to project in three, four, or five dimensions did not change the sigma shape of the Shepard diagram. The Shepard diagram for the t-SNE projection in the bottom-right corner of Fig. 14 has significantly higher point dispersion compared to MDS.

4.4 Dimensionality Reduction on NYC Taxi Flows

This section analyses and visualizes the similarities of Manhattan neighbors in terms of taxi trip making behavior. It also visualizes the similarities of daily taxi flow patterns. The raw time-stamped trip records containing origin and destination zones are aggregated by zone and time period to construct the test data sets. In total, there are 265 zones that cover NY's five boroughs (Manhattan, Brooklyn, Queens, Bronx, and Staten Island). The analysis of similarities of Manhattan neighbors focuses on the top 25 Manhattan zones in terms of trip origins. The analysis of daily taxi flow patterns uses the top 2000 OD pairs in terms of volume regardless of location.

The average daily trips between the top 25 Manhattan zones are shown in Fig. 15. Fig. 15 is a heatmap matrix, in which each cell is representing the flow between two zone pairs. Midtown and uptown zones are shown at the bottom of the matrix, while downtown zones are represented at the top. The trip volumes in the lower right corner of Fig. 15 suggest that there are significant flows from uptown to midtown and vice versa. Likewise, the upper left corner of Fig. 15 reveals that taxi trips that start downtown are more likely to exit downtown. Trips from downtown to uptown are less frequent, with some exceptions, such as trips from Upper East Side South to TriBeCa. Fig. 16 shows the variability of

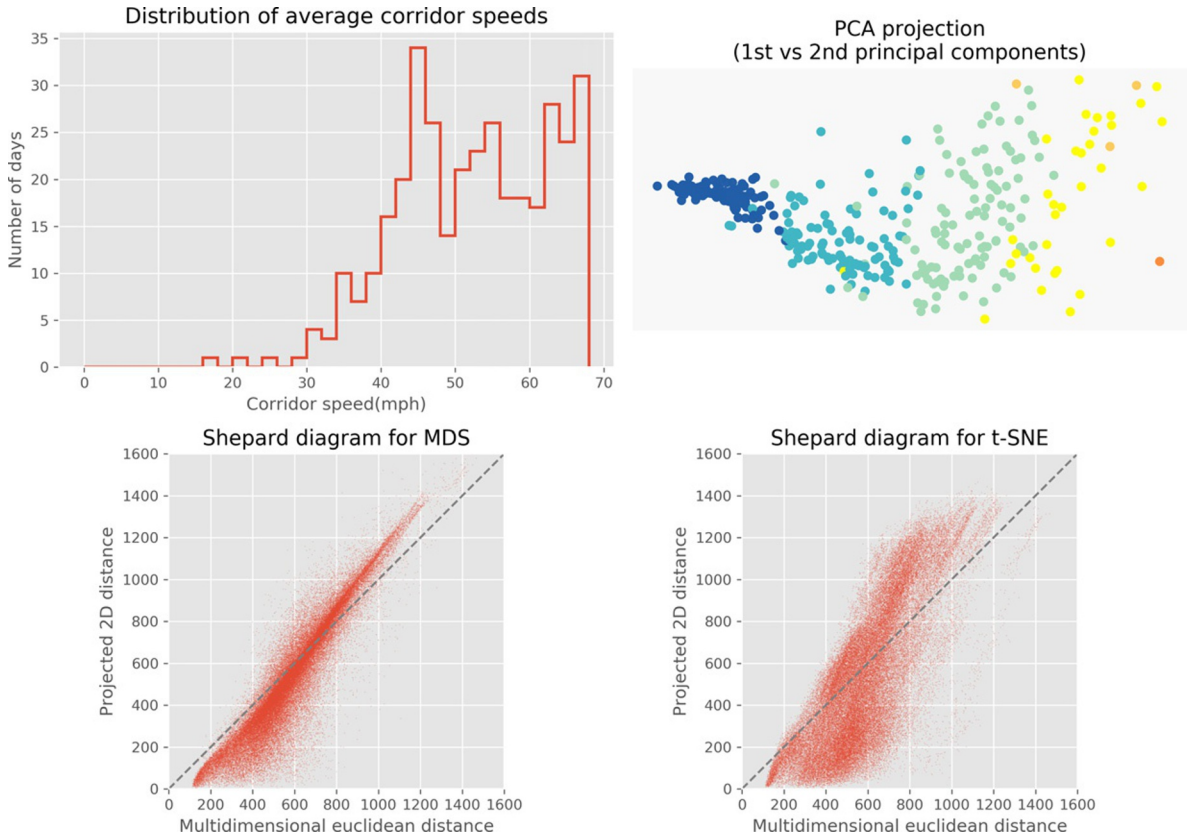

FIG. 14 Dimensionality reduction methods on corridor congestion data set.

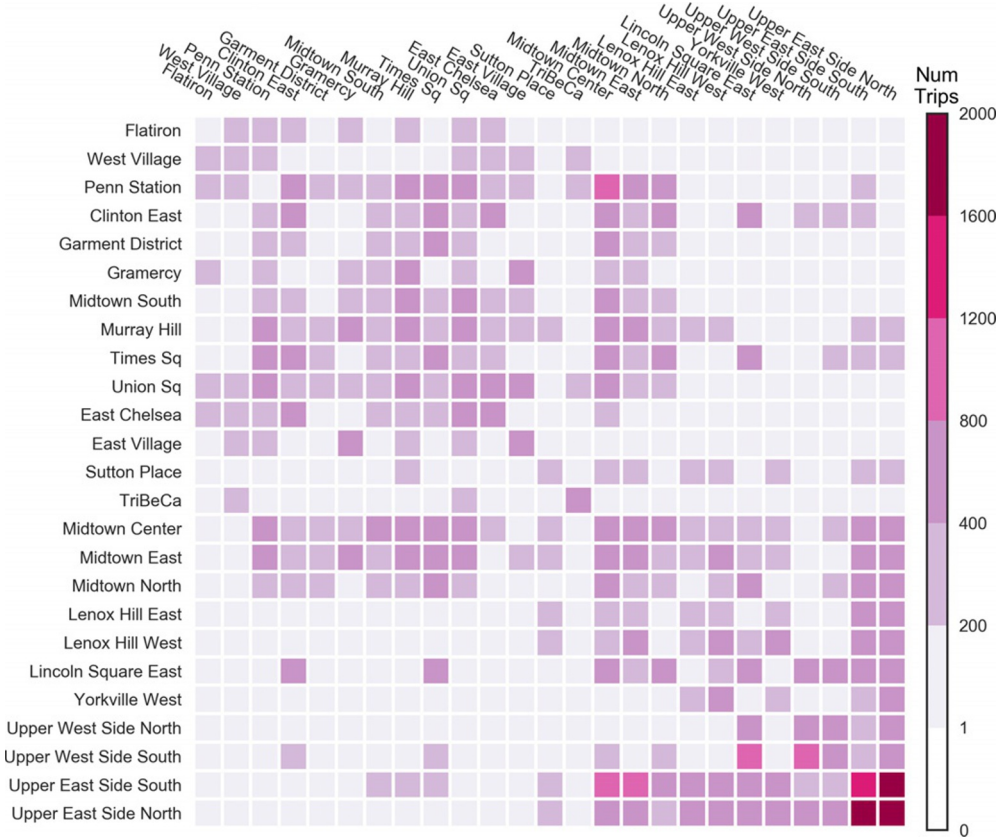

FIG. 15 Zone to zone mean daily taxi trips among the top 25 Manhattan neighborhoods.

FIG. 16 Zone to zone daily trips among the top 25 Manhattan zones between December 19, 2016 and January 15, 2017.

daily trip interchanges between the top 25 zones using the color-coding and the order of the zones in Fig. 15 for the time period between December 19, 2016 and January 15, 2017.

To gain insights on how taxi trips from different zones differ from each other, the test data set was restructured to contain a row for each day and top 25 zone. Columns are the top 25 zones that correspond to destinations. Therefore, a cell in the test data set represents the total volume between an origin zone to a destination zone for a given day. The size of the test data set is 9125 rows and 25 columns. Daily flow values range from 0 to 2970. Two principal components explain 71% of the variation of the data set, 10 principal components explain 97% of the variation.

Applying t-SNE on the test data set produces the map in Fig. 17. Each record, which contains the destination flow vector from a given zone and day, is represented by a circle in the map. The circle is color-coded based on

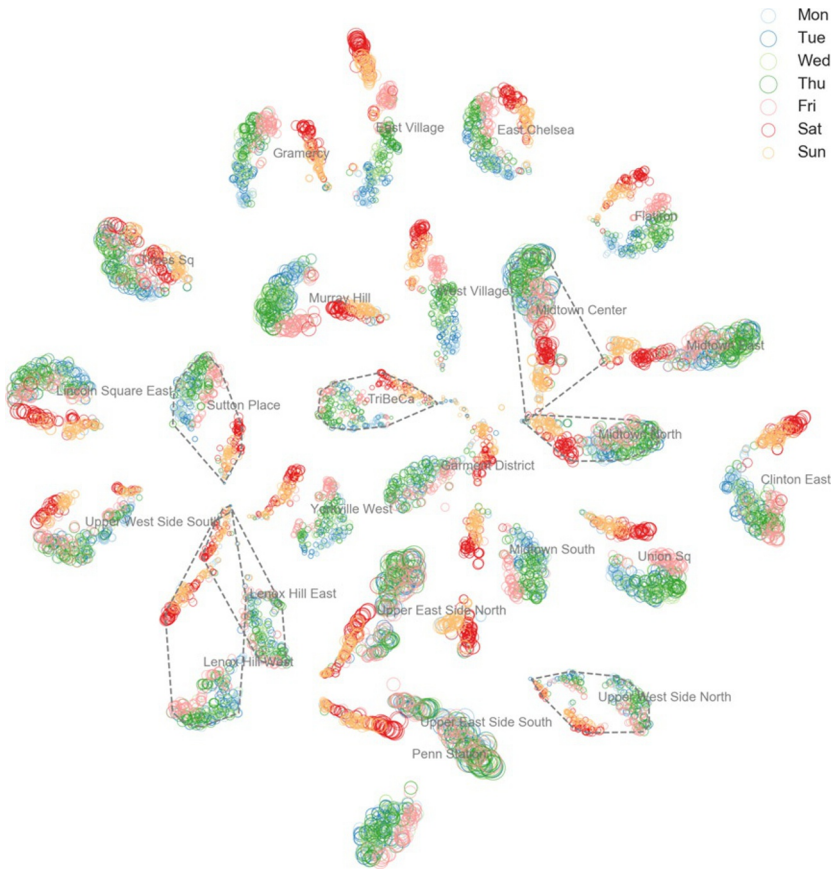

FIG. 17 t-SNE on daily trip volumes to 25 Manhattan neighborhoods.

the day-of-week, while the size of the circle is proportional to the total trips that originate from the zone. The centroid that corresponds to all the points from a given zone has been used to label the zone. For a small number of zones, such as Upper West Side North and Midtown Centre, the convex hull of all zone points has been calculated and drawn with a grey dashed line around the zone points.

From the arrangements of the circles in the map of Fig. 17, it is clear that t-SNE can distinguish trip making patterns between different Manhattan neighborhoods. Also, for most of the zones, points that correspond to Saturdays and Sundays are separated from the rest of the weekdays. Points that correspond to Fridays are usually placed at the edge of the cluster that contains weekdays. The zones that are not encapsulated in their convex hull are not homogeneous with respect to trip origin. For those zones, a small percentage of their points belong to a different cluster that has mostly points from another zone. In terms of general structure, downtown zones tend to be placed at the top of Fig. 17 and uptown or midtown at the bottom.

A flow pattern is determined by the trip interchanges between the 265 zones in NYC. Fig. 18 shows the cumulative percentage of taxi trips as a function of the number of zone pairs that contain those trips. It is clear from the figure that approximately 80% of the total demand is carried by about 5000 zone pairs (that correspond to less than 10% of the total zone pairs). To analyze daily taxi flow

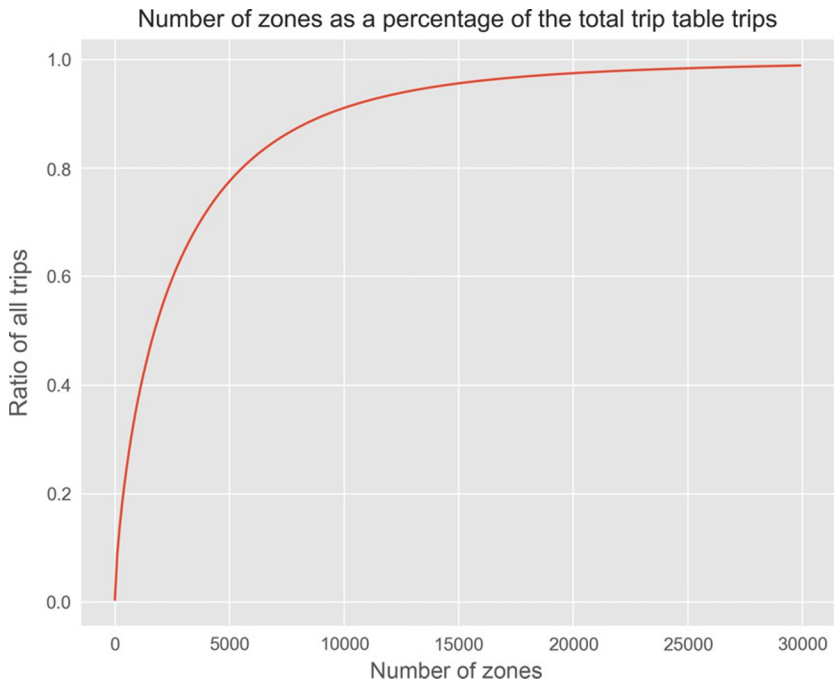

FIG. 18 Cumulative percentage of NYC taxi trips as a function of the number of zone pairs.

patterns, we used the top 2000 zone pairs with the highest average demand. Applications using 100 or 500 zone pairs as features produced fuzzier maps than the ones shown in this section. Also, randomly selecting 2000 pairs instead of the top volume ones produced fuzzier results. In Fig. 19, the cumulative variance explained by the 50 first principal components is shown.

The test data set of the NYC taxi flow patterns has 2000 columns and 1460 rows. Each row contains the vector of trip interchanges that corresponds to a given day and time period. The MDS projection is shown in Fig. 20 and the t-SNE one in Fig. 21. In each of these figures, a glyph corresponds to a period and day. The size of the glyph is proportional to the total trip volume for that particular day and time period. The x-axis of the MDS projection in Fig. 20 appears to correlate with the total volume. Judging from the overlap between the square and left-triangle glyphs, the MD and PM clusters are not completely separated. Also, weekend days are not clearly distinguished into different clusters, with the exception of AM peak patterns.

The t-SNE projection, in contrast, separates all time periods into different clusters. Furthermore, weekend flow patterns are clearly separated from the weekday ones for all time periods. For the nightly patterns, Fridays are mapped separately than Saturdays and Sundays. The Saturday vs. Sunday separation is also evident for PM and AM but not for MD. In all the periods, but for AM, Mondays appear at the one end of the weekday cluster and Fridays at the other end. The t-SNE projection in Fig. 21 uses the Canberra distance. Using the Euclidean distance results in less separation, but without changing the insights presented earlier.

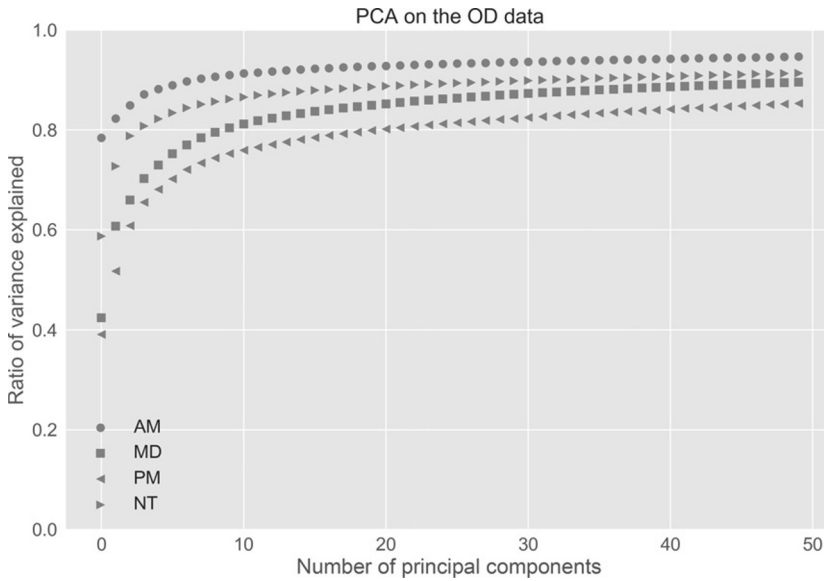

FIG. 19 PCA results for NYC taxi trip data. *AM*, morning; *MD*, midday; *PM*, afternoon/evening; *NT*, night.

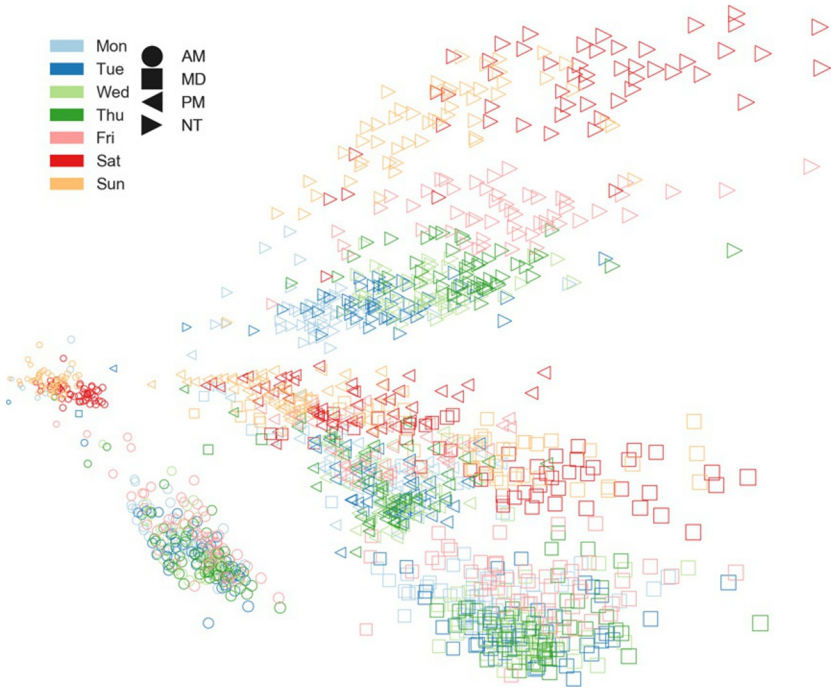

FIG. 20 Multidimensional scaling on NYC taxi flow patterns.

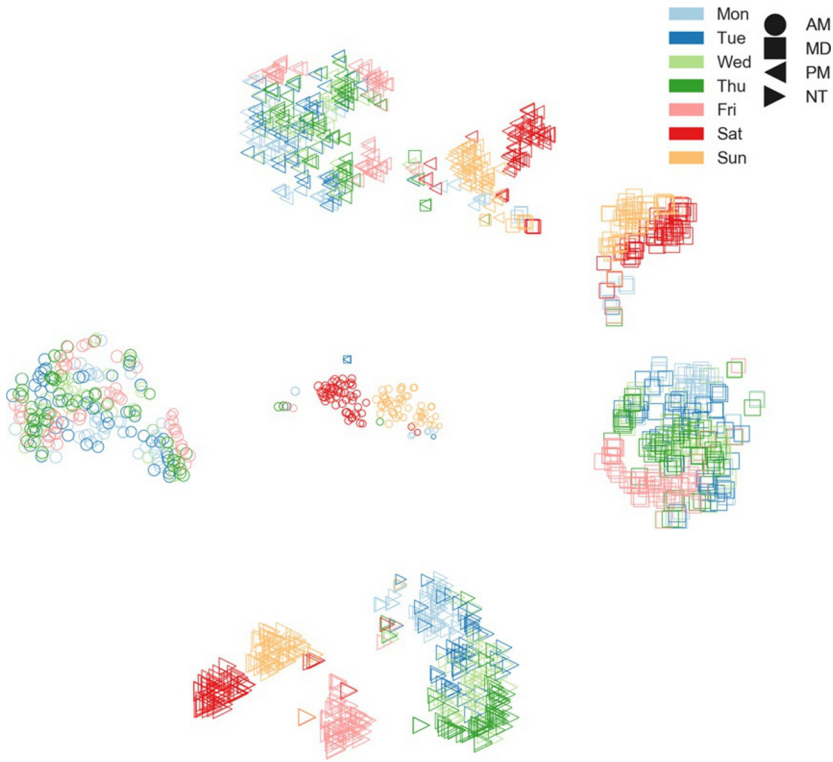

FIG. 21 t-SNE on NYC taxi flow patterns.

4.5 Dimensionality Reduction on the NYC Turnstile Data Set

The MTA daily station count data set contains 2364 days and 266 stations, but it has a low inherent dimensionality. Two principal components are required to represent almost 90% of the variability in the daily count station data (Table 3). The first two principal components are plotted as a scatterplot on the top-left graph of Fig. 22. Each dot is a unique date color-coded for each day of the week (Monday to Sunday). Saturdays and Sundays, in the two principal components graph, form a separate cluster from the weekday points. Fridays are closer to the weekend cloud of points, but they are not clearly separated from the rest of the weekdays.

The MDS graph on the top-right of Fig. 22 is similar to the PCA scatterplot in terms of cluster formation: there is one cluster for the weekends and one for the weekdays. However, the size of the two clusters, which is an indication of the variability in each group is different from the PCA graph. Furthermore, many more outliers are shown, some of them far away from the majority of the data.

The two t-SNE maps at the bottom row of Fig. 22 have more clusters and separate the points further then the maps of the top row. The t-SNE graph on the bottom-left uses Euclidean distance, while the one on the bottom-right the Canberra distance. Saturdays and Sundays are clearly separated from each other in both t-SNE plots. The Euclidean t-SNE plot has most Fridays clustered together and closer to the weekend days, while the Canberra t-SNE has Fridays scattered with the rest of the weekdays. Other than Fridays, different weekdays are not separated from each other, the way Saturday and Sunday are. Rather, various weekdays from separate clusters in both t-SNE maps. It appears that using the Canberra distance produces more separation and emphasis on the local differences. Finally, both t-SNE plots there have no global outliers.

TABLE 3 PCA Results for the NYC Turnstile Data Set

Number of Principal Components	Cumulative Explained Variance Ratio
1	0.872
2	0.896
3	0.913
4	0.925
8	0.951
9	0.954
46	0.990
125	0.999

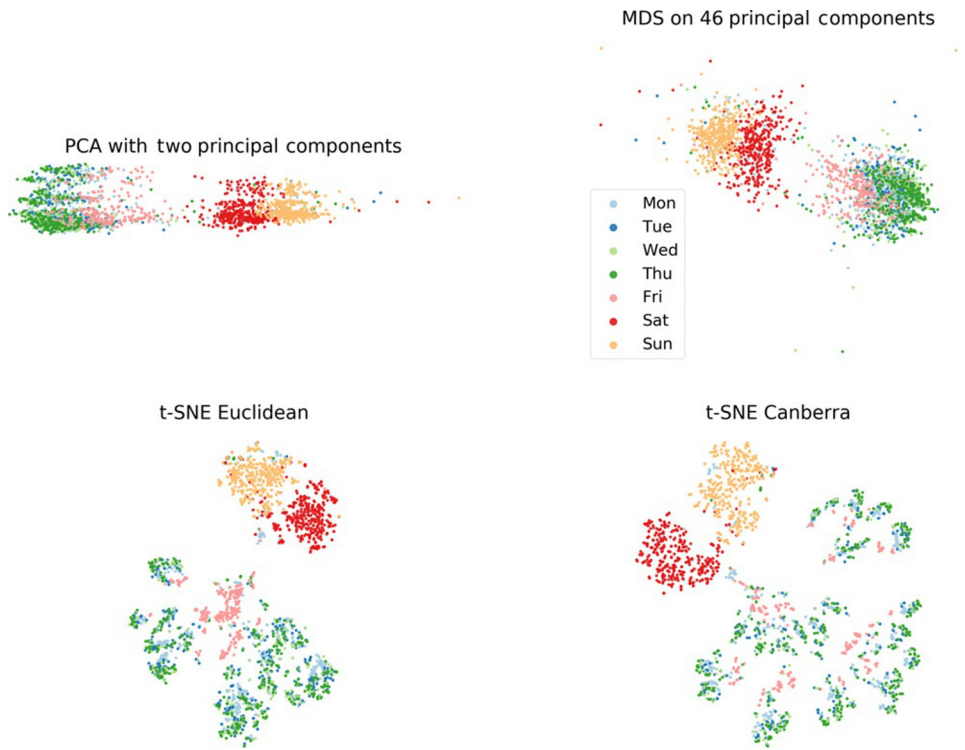

FIG. 22 Dimensionality reduction methods on the New York City turnstile station counts.

5 CONCLUSIONS

Understanding the characteristics and daily fluctuations of demand at the city level can be useful to city planners but also to a growing number of companies that provide services, such as for-hire vehicles. A growing number of sources, including cell-phone and location-based-services data, allow the transportation planner to measure and analyze the demand for travel. Traditional transportation data sets, such as count data, do not measure demand, as they always report the maximum throughput volume, which is less than demand at congestion. Travel surveys, on the other hand, are conducted very infrequently and usually collect information about only a small sample of the population.

In this section, multidimensional and high-dimensional data visualization techniques were applied to a number of transportation data sets of growing importance, such as link performance data obtained from probe vehicles and observed demand in the form of trip data that contain the full population. High-dimensional visualization techniques were applied to three levels of demand aggregation starting from the most aggregate to the most disaggregate.

A. Daily counts (number of transit trips) for 266 MTA stations. Observations are days and columns are daily counts.
B. Daily taxi trip interchanges from a given origin zone to 25 destinations. Each observation is a Manhattan taxi zone for a given day. The number of rows is number of zones by number of days. A column contains the daily trip volume to a top 25 destination.
C. Within day peak period trip interchanges for the top 2000 taxi zone pairs. Each row is a peak period demand pattern that consists of zone to zone volumes among the top 2000 zone pairs, in terms of generated demand. The total number of rows is number of days by number of time periods. Each column contains the peak volume for one of the 2000 zone pairs.

High-dimensional data visualization and specifically t-SNE was able to distinguish the following patterns in the projection map for each of the data sets presented earlier:

A. Sundays, Saturdays, and Fridays formed separate clusters if Euclidean distance is used. The rest of the weekdays formed multiple clusters with a combination of days in each cluster.
B. Each origin zone formed its own cluster. Saturdays and Sundays are separated from the rest of the weekdays and in some cases are separated from each other. Weekdays usually form a continuous spectrum. Sometimes there is an ordering among the days in the weekday cluster as in Murray Hill or West Village but the days are usually not separated from each other. Fridays most of the times are at one edge of the weekday cluster.
C. Peak periods are separated into different clusters. Saturdays and Sundays are separated from the rest of the weekdays and in all periods, except

midday, are separated from each other. For all periods, except the morning, there appears to be an ordering of the weekdays within the weekday cluster.

MDS was successful in presenting a global map of the data that focuses on keeping very dissimilar observations in the high-dimensional space further apart in the projection. MDS was able to project the pairwise distances faithfully in the small car data set, but was not so successful doing so with the rest of the data sets, as revealed qualitatively from the Shepard diagrams. Nevertheless, the MDS map projection of the corridor congestion patterns is a useful one, because it reveals that as congestion increases, so does the variability of corridor congestion patterns represented by speed heatmaps. For the rest of the data sets, MDS maps were significantly less clustered than those produced by t-SNE and there was significant overlap between expected groupings.

As data and methods evolve, the earlier findings will certainly be further explored. We expect that the application of these techniques in other transportation and mobility data sets by researchers and practitioners are going to reveal additional depth and emerging patterns. For example, further research is required to investigate the properties and robustness of clusters in the t-SNE maps.

REFERENCES

Artero, A.O., de Oliveira, M.C.F., Levkowitz, H., 2004. In: Uncovering clusters in crowded parallel coordinates visualizations. IEEE Symposium on Information Visualization, 2004. INFOVIS 2004. IEEE, pp. 81–88.

Bateman, S., Mandryk, R.L., Gutwin, C., Genest, A., McDine, D., Brooks, C., 2010. In: Useful junk?: the effects of visual embellishment on comprehension and memorability of charts. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, pp. 2573–2582.

Borg, I., Groenen, P.J., 2005. Modern Multidimensional Scaling: Theory and Applications. Springer Science & Business Media, New York.

Bunte, K., Biehl, M., Hammer, B., 2012. A general framework for dimensionality-reducing data visualization mapping. Neural Comput. 24 (3), 771–804.

Cleveland, W.S., Cleveland, W.S., 1985. The Elements of Graphing Data. vol. 2 Wadsworth Advanced Books and Software, Monterey, CA.

De Oliveira, M.F., Levkowitz, H., 2003. From visual data exploration to visual data mining: a survey. IEEE Trans. Vis. Comput. Graph. 9 (3), 378–394.

European Environmental Agency, 2018. Chart Do's and Don'ts. Available from:https://www.eea.europa.eu/data-and-maps/daviz/learn-more/chart-dos-and-donts.

Jolliffe, I.T., 1986. Principal component analysis and factor analysis. In: Principal Component Analysis. Springer, New York, NY, pp. 115–128.

Keim, D.A., 2000. Designing pixel-oriented visualization techniques: theory and applications. IEEE Trans. Vis. Comput. Graph. 6 (1), 59–78.

Koch, I., 2013. Analysis of Multivariate and High-Dimensional Data. 32Cambridge University Press, Cambridge.

Kosslyn, S.M., 1994. Elements of Graph Design. WH Freeman, New York, NY.

Liu, S., Maljovec, D., Wang, B., Bremer, P.-T., Pascucci, V., 2017. Visualizing high-dimensional data: advances in the past decade. IEEE Trans. Vis. Comput. Graph. 23 (3), 1249–1268.

Nuñez, J.R., Anderton, C.R., Renslow, R.S., 2017. Optimizing Colormaps With Consideration for Color Vision Deficiency to Enable Accurate Interpretation of Scientific Data. arXiv preprint arXiv:1712.01662.

Onderwater, M., 2015. Outlier preservation by dimensionality reduction techniques. Int. J. Data Anal. Tech. Strat. 7 (3), 231–252.

Parallel Coordinates, 2018. Parallel Coordinates. Available from:https://bl.ocks.org/jasondavies/1341281.

Roweis, S.T., Saul, L.K., 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290 (5500), 2323–2326.

Sammon, J.W., 1969. A nonlinear mapping for data structure analysis. IEEE Trans. Comput. 100 (5), 401–409.

Scikit-learn, 2018. sklearn.manifold.TSNE. Available from:http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html.

Steele, J., Iliinsky, N. (Eds.), 2010. Beautiful Visualization: Looking at Data Through the Eyes of Experts. O'Reilly Media, Inc., Sebastopol, CA

Tenenbaum, J.B., De Silva, V., Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. Science 290 (5500), 2319–2323.

Torgerson, W.S., 1952. Multidimensional scaling: I. Theory and method. Psychometrika 17 (4), 401–419.

Tufte, E.R., 1983. The Visual Display of Quantitative Information. Graphics Press, London.

Tufte, E.R., 1990. Envisioning Information. Graphics Press, London.

Tufte, E.R., 1997. Visual and Statistical Thinking: Displays of Evidence for Making Decisions. Graphics Press, London.

Tukey, J.W., 1977. Exploratory Data Analysis. 2 Addison-Wesley, Reading, MA.

Van Der Maaten, L., 2014. Accelerating t-SNE using tree-based algorithms. J. Mach. Learn. Res. 15 (1), 3221–3245.

Van Der Maaten, L., Hinton, G., 2008. Visualizing data using t-SNE. J. Mach. Learn. Res. 9 (Nov), 2579–2605.

Van Der Maaten, L., Postma, E., Van den Herik, J., 2009. Dimensionality reduction: a comparative. J. Mach. Learn. Res. 10, 66–71.

Wegman, E.J., 1990. Hyperdimensional data analysis using parallel coordinates. J. Amer. Stat. Assoc. 85 (411), 664–675.

Wickham, H., 2010. A layered grammar of graphics. J. Computat. Graph. Stat. 19 (1), 3–28.

Wickham, H., 2016. ggplot2: Elegant Graphics for Data Analysis. Springer, New York 2nd ed.

Wilkinson, L., 2006. The Grammar of Graphics. Springer Science & Business Media, New York.

FURTHER READING

Elmqvist, N., Fekete, J.D., 2010. Hierarchical aggregation for information visualization: overview, techniques, and design guidelines. IEEE Trans. Vis. Comput. Graph. 16 (3), 439–454.

Gapminder, 2018. Tools. Available from:https://www.gapminder.org/tools/#$chart-type=bubbles.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Vanderplas, J., 2011. Scikit-learn: machine learning in python. J. Mach. Learn. Res. 12 (October), 2825–2830.

Scipy 2018, Distance Computations, Available from: https://docs.scipy.org/doc/scipy/reference/spatial.distance.html.