1 INTRODUCTION

Line charts, which date back to William Playfair [39], are commonly used for visualizing time-series and continuous data. Borkin et al. found that line charts are the second most frequently used visualization type, only behind bar charts, in scientific publications, news media, government, and world organizations materials [11].

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxxxx
of line charts is subject to the choice of aspect ratio, which can be automatically optimized for a chart [54]. Javed et al. evaluated the effectiveness of line charts with small multiples, horizon graphs, stacked graphs, and braided graphs for comparison, slope, and discrimination tasks [29]. The results showed that techniques with separate charts performed better for data with large visual spans, while shared-space techniques were better for short spans. A more recent study showed that overlaid line charts perform better than small multiples in comparison tasks [36]. Finally, adding user interaction to line charts can enhance the user experience without a loss to efficacy [5].

2.2 Decision-Making

Line charts have been studied in several decision-making scenarios as well. In time-sensitive application settings, the ability to accurately interpret a line chart "at a glance" is crucial. Recently, Pixel Approximate Entropy (PAE) was used as a metric for the perceptual complexity of line charts, and it was shown that increased chart PAE correlates with reduced judgment accuracy [42].

Missing Data Another decision-making challenge in visualization is when data are missing; one needs to impute the missing data into the visualization [51]. An early study that looked at the problem of missing data in line charts on trend and comparison tasks found that even with missing data, user performance was high [21]. One way to address missing data is to use additional visual channels, e.g., color, empty points, or error bars, which have been shown to improve analysis performance and confidence of users on average and trend finding tasks [43].

Distortion Another concern for decision-making is distortion in the visualization, such as an inverted axis or a distorted aspect ratio. Such situations can demonstrate a reversal of messaging, which can lead viewers to draw false inferences and judgments [37]. Another example is when missing data are misleadingly inserted into a visualization, e.g., assigned arbitrary values, user performance can go down significantly [21]. To address this weakness, multi-view systems have been proposed to assist in time-series data quality checking [8].

2.3 Line Chart Smoothing

Smoothing line charts can be considered a form of distortion, as the data are being distorted to improve clarity. There has been prior work looking at smoothing in the signal processing community, e.g., Shao et al. compared 5 smoothing methods for vegetation classification [47], and image processing community, e.g., Chen and Yeh developed a quantitative evaluation for edge-preservation in image smoothing [15]. To our surprise, we were unable to find any prior studies that evaluated the impact of various smoothing techniques to line charts, except for our own small-scale study that introduced a topology-based smoothing method [19]. Nevertheless, no comprehensive framework and evaluation, such as the one we are introducing in this paper, exists.

3 Taxonomy of Line Chart Smoothing Approaches

We discuss 4 classes of smoothing that can be used on line charts. They can be broadly broken down into methods that consider local properties of the input through 1 or more adjustable simplification parameters. The number of available smoothing techniques is large. Therefore, this list is intended to be representative of well-known techniques, not necessarily comprehensive.

3.1 Local Methods

Local methods only consider nearby data when calculating their smoothed output. Essentially, for each output data, a local neighborhood of the input data is extracted. Then, the neighborhood is processed by a filter, and the result is used as the output.
3.1.1 Rank Filters

Rank filters are nonlinear filters that, for each input point, ranks (i.e., sorts) a neighborhood window surrounding the input point. A single value is selected from the ranked set for output. The MEDIAN filter (see Fig. 1(c) (left)) selects the median value from the ranked neighborhood. The level of smoothing can be increased or decreased by enlarging or shrinking the neighborhood window, respectively. MEDIAN filters are known for being particularly good at removing salt-and-pepper noise [2], but if, on the other hand, those peaks represent important data, they will be lost with a MEDIAN filter.

To compute the MEDIAN filter (see Fig. 2(a)), for each of \( n \) input points, a window of size \( w \) is first selected. Next, the window is sorted. Finally, the median value is selected for output. The boundary of the domain requires special consideration. Several options exist for the boundary—we chose to repeat the boundary value infinitely. In a naive implementation of the MEDIAN filter, repeated sorting operations are required, 1 per input/output point, making the overall performance \( \mathcal{O}(n \cdot w \cdot \log w) \). The operation can be optimized by using a sliding window to achieve \( \mathcal{O}(n \log w) \) in the general case [23] and \( \mathcal{O}(n) \) in limited cases [38].

Additional examples of rank filters include MIN filter (see Fig. 1(c) (middle)) and MAX filter (see Fig. 1(c) (right)), which operate similarly, except that they select the minimum and maximum value from the ranked lists, respectively.

3.1.2 Convolutional Filters

Convolutional filters are a stencil-based method, where for a given input point, a series of weights are applied to a neighborhood surrounding that point. To compute a convolutional filter (see Fig. 2(b)), for each of \( n \) input points, a window of size \( w \) is selected. Next, the elements are multiplied by their corresponding elements from the stencil, summed, and that value is placed in the output. Similar to rank filters, the boundary of the domain requires special consideration. For consistency, we chose to repeat the boundary values infinitely. The resulting computational complexity for general convolutional filters is \( \mathcal{O}(n \cdot w) \).

The GAUSSIAN filter (see Fig. 1(d) (left)) is commonly used in convolutional signal and image processing [12]. It weights the input neighborhood using a normal distribution. The smoothing level is increased or decreased by adjusting the standard deviation, \( \sigma \), of the distribution. The GAUSSIAN filter can be seen as a form of a low-pass filter, blurring both signal and noise from the data, producing smooth, visually appealing results. The window used for the GAUSSIAN filter is fixed using \( \sigma \) as a guide. In our implementation, a window size of \( \pm 4\sigma \) ensures that we capture over 99.9% of the distribution.

Another simple convolutional filter is the MEAN filter (see Fig. 1(d) (right)), also known as the moving average. In this case, equal weights are applied to all elements in the window, resulting in the average being calculated. Because of the equal weighting, a sliding window is used to improve performance to \( \mathcal{O}(n) \) complexity.

Finally, SAVITZKY-GOLAY [45] (see Fig. 1(d) (middle)) is a convolutional filter that uses a low-degree polynomial to smooth the data.

3.2 Global Methods

With global methods, the entire input data is considered in the calculation of the output.

3.2.1 Frequency Domain Filters

Frequency domain filtering converts the scalar data into a frequency domain representation, via wavelets or Fourier transform. Once in the frequency domain, undesirable frequencies are removed, and the signal is reconstructed. We consider a low-pass CUTOFF filter (see Fig. 1(e) (left) and Fig. 2(c)), which converts the input into the frequency domain using a Discrete Fourier Transform (DFT) [15]. High-frequency components are then zeroed out to smooth the output above a cutoff frequency. Lowering that cutoff frequency increases the level of smoothing. Finally, the output is computed by converting the frequency domain data back to the spatial domain using an inverse DFT. Much like the GAUSSIAN filter, the CUTOFF filter produces smooth, visually appealing output, in this case, only retaining the specified frequencies. However, the relationship between the frequency and spatial domains is often not intuitive, as multiple frequencies contribute to a single output. The computational complexity of the DFT and the CUTOFF filter is \( \mathcal{O}(n \log n) \).

Additional frequency domain low-pass filters we consider include the BUTTERWORTH filter [13] (see Fig. 1(e) (middle)) and CHEBYSHEV filter [41] (see Fig. 1(e) (right)). The CUTOFF filter is an idealized function that cannot be implemented in a circuit, while BUTTERWORTH and CHEBYSHEV filter can. Practically speaking, these methods differ from the CUTOFF filter in that they provide a gradual ramp-down of the cutoff frequency.

| Technique            | Class              | Average Complexity |
|----------------------|--------------------|--------------------|
| MEDIAN               | Rank               | \( \mathcal{O}(n \log w) \) |
| MIN                  | Rank               | \( \mathcal{O}(n) \) |
| MAX                  | Rank               | \( \mathcal{O}(n) \) |
| GAUSSIAN             | Convolutional      | \( \mathcal{O}(n \cdot \sigma) \) |
| MEAN                 | Convolutional      | \( \mathcal{O}(n) \) |
| SAVITZKY-GOLAY       | Convolutional      | \( \mathcal{O}(n \cdot w) \) |
| CUTOFF               | Frequency          | \( \mathcal{O}(n \log n) \) |
| BUTTERWORTH          | Frequency          | \( \mathcal{O}(n \log n) \) |
| CHEBYSHEV            | Frequency          | \( \mathcal{O}(n \log n) \) |
| UNIFORM              | Subsampling        | \( \mathcal{O}(n + s) \) |
| DOUGLAS-PEUCKER      | Subsampling        | \( \mathcal{O}(s \log n) \) |
| TOPOLOGY             | Subsampling        | \( \mathcal{O}(n + c \log c) \) |

\( n \): number of input points; \( w \): window size; \( \sigma \): standard deviation of a normal distribution; \( s \): number of output samples; \( c \): number of critical points (i.e., local minimum or maximum).
3.2.2 Subsampling

Subsampling approaches take the original data and select a subset of the original data points as representatives of the whole data. Simplification is increased by merely selecting fewer points.

A common choice, due to its ease of implementation, UNIFORM subsampling (see Fig. 1(f)(left)) selects points at regular intervals. Between selected points, interpolation is used, with linear interpolation being the most straightforward case. UNIFORM subsampling makes few guarantees about the types of features it preserves unless the input is already oversampled, in which case it retains the original signal [46]. Computationally, UNIFORM subsampling is very efficient, only \( O(n + s) \), where \( s \) is the number of samples taken from the input.

Nonuniform subsampling, in contrast to UNIFORM subsampling, selects points at irregular intervals by considering/preserving some features of the data. DOUGLAS-Peucker [20, 40] (see Fig. 1(f)(middle)) is an example that establishes a priority queue of points by optimizing the \( L^n \)-norm of the residual error (i.e., the difference between the original and smoothed line charts). The algorithm (see Fig. 2(d)) starts by selecting the boundary points of the input data (i.e., first and last points) for initialization and connects them via linear interpolation. Points are then iteratively added by selecting the input point with the largest distance from the current output and inserting it into the output. The process continues until a user-specified threshold distance is reached. The simplification is increased or decreased by modifying this threshold. The output captured by DOUGLAS-Peucker is reliable and predictable, in that the output will deviate no more than the specified threshold. The worst-case complexity of the algorithm is \( O(n^2) \), while the average complexity is \( O(n \log n) \).

An additional nonuniform subsampling approach, the TOPOLOGY filter (see Fig. 10(right)), uses techniques from Topological Data Analysis to smooth data in a way that retains significant peaks and minimizes error [49]. The TOPOLOGY filter works by first identifying critical points, in the form of local minima and local maxima, and forms a hierarchical pairing (1 each, a local minimum and local maximum) between them. Pairs of critical points are then removed from the output if the difference in their value is below a given simplification threshold. Finally, monotonic regression is used to interpolate between the remaining critical points. The overall complexity of the operation is \( O(n + c \log c) \), where \( c \) is the number of local minima and maxima.

4 Analytical Framework for Measuring for Smoothing Efficacy in Line Charts

In this section, we describe our analytical framework for evaluating line chart smoothing. It consists of 3 parts: a set of effectiveness measures for line chart smoothing (see Sect. 4.1), a description of the relationship between the effectiveness measures and common visual analytics tasks (see Sect. 4.2), and a description of the methodology for comparing different line chart smoothing techniques (see Sect. 4.3).

4.1 Measures of Effectiveness

To better understand the quality of smoothing results produced by each smoothing technique, we consider a set of measures that compare the input data, \( X = \{x_0, x_1, \ldots, x_n\} \), and the smoothed data, \( Y = \{y_0, y_1, \ldots, y_n\} \). There is no single measure to evaluate the effectiveness of smoothing under all visual analytics tasks. Therefore, we use a series of measures, each of which relates how well the smoothing technique preserves a particular quality of the input data. For all measures, a value of 0 indicated no error, while larger positive values indicate increasing errors.

4.1.1 Total/Maximum Value Variation

The first measure considers calculating the difference between the input and the smoothed data using vector norms, which measure and sum the difference between the data at each sample location. Considering the illustration in Fig. 3(a), we apply 2 variations, the \( L^1 \)-norm and the \( L^\infty \)-norm.

The \( L^1 \)-norm, \( \ell_1 \), also known as the least absolute deviations or least absolute errors, measures the sum of the absolute value of the difference between the input and smoothed data. In other words, in Fig. 3(a), it measures the sum of the differences in black. As a measure, it is robust, in that it is resistant to the influence of outliers. The \( L^1 \)-norm is:

\[
\ell_1(X, Y) = \sum_{i=1}^{n} |x_i - y_i| \tag{1}
\]

The \( L^\infty \)-norm, \( \ell_\infty \), measures only the point of the largest difference between input and smoothed data. In Fig. 3(a) this is the point denoted by the arrow. The \( L^\infty \)-norm is:

\[
\ell_\infty(X, Y) = \max_i |x_i - y_i| \tag{2}
\]

4.1.2 Area Preservation

In some cases, the individual deviations matter less than the total area captured under the line chart. The change in the area, \( \delta_a \), is found by taking the difference between the integrals of the input and smoothed data. Fig. 3(b) illustrates the process. The change in area is the difference between the sum of all grey bars and the sum of all pink bars. The change in area is:

\[
\delta_a(X, Y) = \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} y_i \tag{3}
\]

4.1.3 Total/Maximum Peak Variation

The next measure identifies and matches the similarity of peaks, i.e., local minima and maxima, between the original and smoothed data. Fig. 3(c) shows examples of such peaks. To measure the similarity, we...
use techniques from Topological Data Analysis [22]. First, the local minima and maxima of the original and smoothed data are calculated and paired in a process described in detail in [39]. The pairs are placed into 2 sets \( \mathcal{X} \) and \( \mathcal{Y} \) and let \( \eta \) be a bijection between the 2 sets.

The Wasserstein distance measures the total difference between all peaks, giving higher weight to those with larger differences. The 1-Wasserstein distance, \( W_1(\mathcal{X}, \mathcal{Y}) \), is:

\[
W_1(\mathcal{X}, \mathcal{Y}) = \inf_{\eta : \mathcal{X} \to \mathcal{Y}} \sum_{x \in \mathcal{X}} ||x - \eta(x)||_1
\]

(4)

The Bottleneck distance only measures the peaks with the maximum difference. The Bottleneck distance is:

\[
W_\infty(\mathcal{X}, \mathcal{Y}) = \inf_{\eta : \mathcal{X} \to \mathcal{Y}} \sup_{x \in \mathcal{X}} ||x - \eta(x)||_{\infty}
\]

(5)

### 4.1.4 Frequency Preservation

Generally speaking, a smoothed signal should maintain as much of the frequency spectrum as possible. To measure the preservation of frequencies, \( \mathcal{F} \), we convert the original and smoothed data into the frequency domain using the Discrete Fourier Transform (DFT), \( F_X \) and \( F_Y \), respectively. Once the DFTs are calculated, their difference is found using the \( L^2 \)-norm between them:

\[
\mathcal{F}(F_X, F_Y) = \left( \sum_{k=1}^{n} (F_{X,k} - F_{Y,k})^2 \right)^{1/2},
\]

(6)

where \( k \) is a single frequency of interest. Fig. 3(d) illustrates the frequency domain before and after smoothing. The frequency preservation would be the \( L^2 \)-norm of the difference between these 2 vectors.

### 4.1.5 Value-Order Preservation

In some scenarios, knowing that the relative values of data items are maintained is more important than maintaining the correct values. The value-order relationship can be measured using the correlation between the input and smoothed data. To measure the relationship between relative values, the Pearson Correlation Coefficient, \( \rho \), can be employed. Fig. 3(e)(left) illustrates the value relationship by placing points at \((X_i, Y_i)\). Intuitively, \( \rho \) measures the proximity of the points to the diagonal in grey. In order to treat \( \rho \) consistently with other measures, we modify it, such that 0 is a perfect positive correlation, and 2 is a perfect negative correlation. The modified \( \rho \) is:

\[
\rho(X, Y) = 1 - \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}
\]

(7)

The order relationship between data items can be measured using Spearman Rank Correlation, \( r_s \), which is the Pearson Correlation Coefficient of the ranked data, in other words:

\[
r_s(X, Y) = \rho(\text{rank}(X), \text{rank}(Y))
\]

(8)

Fig. 3(e)(right) illustrates the order relationship. The points are placed by rank, instead of value, and their \( \rho \) is calculated.

### 4.2 Low-Level Task Taxonomy for Line Charts

To determine relevant visual analytics tasks, we adapt the low-level task taxonomy of Amar et al. [7] to line charts. For each, we provide a brief description of the task and an example query. Finally, we relate each of these tasks to 1 or 2 of the metrics from the previous section, which is summarized in Table 2.

1 For technical reasons, all diagonal points \((x, x)\) are added to make the cardinality infinite [31].
The **Find Extremum** task is concerned with finding local minima and maxima (i.e., valleys and peaks) in the data. For example, using Fig. 4, “What are the dates/values of all of the peaks in the data?“ (answer: Nov ’14/4.5, Jan ’15/5, Mar ’15/5). The accuracy of the task depends upon the peaks and/or valleys remaining present in the output and significant enough to be visible. The total/maximum peak variation (see Sect. 4.1.3 and Fig. 3(c)) measures both the existence and significance of peaks in the data. The average case is provided by the 1-Wasserstein distance (Eq. 4), which finds the total variation in peaks between the input and smoothed data. The worst case is found using the Bottleneck distance (Eq. 5), which measures the peak of maximal variation.

The **Find Anomalies** task involves looking for values that do not conform to the overall trend in the data. For example, “Between Nov ’14 and Feb ’15, what month, if any, does not follow the data trend?” (answer: Dec ’14/Jan ’15). The task of finding anomalies is similar to finding extrema, in that anomalies are generally peaks in the data, but in this case, they do not follow the trend of the data. Since the task involves identifying peaks, the average case is provided by the 1-Wasserstein distance (Eq. 4), and the worst case is found using the Bottleneck distance (Eq. 5). Interestingly, the removal of anomalies is also one of the reasons smoothing is applied to line charts. Therefore, when performing other tasks, the preservation of anomalies might be considered a negative quality.

The **Characterize Distribution** task involves summarizing a trend in the data. For example, using Fig. 4, “What is the trend in the data between Nov ’14 and Feb ’15”? (answer: downward). Trends in the data are synonymous with the frequency domain of the data. To be effective, the frequency domain of the smoothed data should be as similar as possible to that of the input data. Therefore, the average case accuracy of this task is measurable using the frequency preservation measure (see Sect. 4.1.3 and Fig. 3(c)).

The **Sort** task asks users to give some criteria, rank or order the data values. An example, using Fig. 4, would be, “What is the order of stock values for the dates Nov ’14, Jan ’15, and Mar ’15, from lowest to highest?” (answer: Jan ’15, Nov ’14, Mar ’15). While similar to the retrieving value task, the accuracy of this task relies both upon the relative order of values (not the exact values) of data remaining the same, and further, the difference between those relative values is reasonably discernible. The value-order preservation measures (see Sect. 4.1.5 and Fig. 3(c)) provide two mechanisms to understand the average case performance. First, Spearman Rank Correlation (see Eq. 8) can be used to compare the relative order of all points in the original and smoothed data. The Pearson Correlation Coefficient (see Eq. 7) can be used to determine, on average, how discernible the values in the smoothed data are from one another, as compared to the input.

The **Cluster** task asks users to group data with similar values or trends. For example, “Group months with similar trends” (answer: see left). Depending upon the nature of the query (clustering individual points vs. clustering trends), this task depends upon both the judgment of relative values and trends in the data. Therefore, the average case performance is summarized by the frequency preservation measure (see Sect. 4.1.3 and Fig. 3(d)) from Eq. 6 for clustering trends, as well as the value-order presentation measures (see Sect. 4.1.5 and Fig. 3(c)). Spearman Rank Correlation (see Eq. 8) and Pearson Correlation Coefficient (see Eq. 7) for clustering individual points.

Tasks Not Considered  Amar et al. [7] defined additional low-level tasks that we found redundant or out-of-scope for this analysis. First was the filter task, which we felt was redundant with tasks such as determine range, characterize distribution, and find anomalies. The second was the correlate task, which we felt would require comparing multiple distributions for their similarity. While this task could potentially be analyzed with our framework, we only consider the effectiveness of tasks on a single line chart.

### 4.3 Evaluation Framework

Given the metrics and tasks described in the prior subsections, we establish our framework for comparing the efficacy of smoothing techniques. The idea is to rank the effectiveness of all smoothing techniques for given data on a specific visual analytics task.

This requires 2 things: (1) a common measure of the smoothing level (i.e., a baseline for measurement); and (2) a method for ranking the efficacy of methods, using a specific metric.

#### 4.3.1 Visual Complexity as a Proxy for Smoothing Level

As noted in our taxonomy of smoothing techniques, each method provides 1 or more input parameters for adjusting the level of smoothing. However, the input parameters for each technique have little to no direct relationship to any other technique. For example, Gaussian smoothing with $\sigma = 5$ samples has no analog to UNIFORM subsampling of 50% of points, even though they may produce similar results. This makes the analytical comparison of techniques difficult.

Recently, approximate entropy ($ApEx$) was shown to be a high quality proxy for the visual complexity of line charts [22]. Generally speaking, approximate entropy is a measure that quantifies predictability of fluctuations in the data. In our case, we see visual complexity and smoothing level as synonymous—therefore, $ApEx$ is used as a baseline for our analysis. In other words, if the smoothed outputs of 2 different techniques have the same $ApEx$, we consider their smoothing level identical, e.g., in Fig. 1, all methods have similar $ApEx$ values. See [22] for a description of how to compute $ApEx$.

#### 4.3.2 Ranking the Effectiveness of Smoothing

To select the most effective technique for a given metric, we want to focus on those that have the smallest value.

**Ranking a Single Smoothing Level** When smoothing results have equivalent $ApEx$, ranking their effectiveness is fairly trivial. For a given metric, the techniques are simply ordered from lowest (best) to highest (worst). For example, in Fig. 1(b), the $L^1$-norm, $\ell_1$ in the first column, shows that TOPOLOGY has the lowest error, followed by GAUSSIAN and SAVITZKY-GOLAY. The rankings can be computed for all metrics, and the smoothing techniques evaluated for all tasks. For example, in Fig. 1(b), SAVITZKY-GOLAY is in the top 3 for all metrics/tasks, making it a reasonable choice to represent the input data.

**Ranking All Smoothing Levels** Summarizing the performance of different smoothing techniques across all smoothing levels requires additional analysis. We calculate 100 different smoothing levels, across a range of entropy values. For each metric, we create an entropy plot, which is a scatterplot of the metric value against a range of $ApEx$.
values. In Fig. 5(a) the $L^1$-norm is plotted vertically, against the ApEx horizontally for TOPOLOGY, in red, and CHEBYSHEV, in purple.

Next, both linear and logarithmic regression are performed using iterative reweighted least-squares (IRLS) [28]. The model, linear or logarithmic, with the larger $R^2$ value is selected as a proxy for the efficacy of the technique. For Fig. 5(b) TOPOLOGY is best modeled logarithmically, and CHEBYSHEV is best modeled linearly.

Since the goal is again to minimize the error induced in the data, the total area under the regression is computed (i.e., mathematical integration), and the methods are ranked smallest to largest area. In Fig. 5(c) the total area is $\sim 2500$ for TOPOLOGY and $\sim 4350$ for CHEBYSHEV, making TOPOLOGY more effective than CHEBYSHEV.

The resulting ranks are placed into a rank plot, as seen in Fig. 7. In this plot, the $L^1$-norm is ranked across multiple datasets. Each dataset receives a column, and the smoothing methods are ranked from best (top) to worst (bottom). The tracks are added to improve readability.

Ranking Across Multiple Datasets To summarize the overall efficacy of techniques across multiple datasets, an average rank is calculated. The average rank simply takes the sum of the rank across all datasets and orders them from lowest (top) to highest (bottom). In Fig. 7 the average rank is the final column. The average rank of TOPOLOGY is 3.5, while the GAUSSIAN is 2.0, SAVITZKY-GOLAY is 3.5, etc.

Using IRLS helps to minimize the impact of outliers.

5 RESULTS

The source code for our evaluation is available at <https://github.com/USFDataVisualization/LineSmooth>, and an interactive version of our framework at <https://usfdatavisualization.github.io/LineSmoothDemo>. The results for all data and smoothing methods can also be found in our supplementary materials.

5.1 Data Sources

We evaluate 80 datasets in 13 categories from 8 data sources (see Fig. 6).

- Chicago Homicide Rates (chi_homicide) data (see Fig. 6(a)) contains weekly (969 samples) and monthly (222 samples) counts of the number of homicides in the city from January 2001 through July 2019. Data is provided by the City of Chicago [14].

- EEG (eeg_500, eeg_2500, and eeg_10000) data (see Fig. 6(b)) contains windows of 3 different lengths (500, 2500, and 10000 samples) from 6 (of 32 total) channels from a single subject undergoing a visual attention task and was acquired from the EEG/ERP Public Archive [19].

- New Zealand Tourist (nz_tourist) data (see Fig. 6(c)) contains the monthly (1165 samples) and annual (96 samples) number of tourists visiting the country from April 1921 through April 2018. Data collected from Trading Economics [2].

- US Domestic Flights (flights) data (see Fig. 6(d)) contains the number of daily, weekly, and monthly (7074, 1095, and 252 samples, respectively) number of US flights from January 1, 1988 through December 31, 2008. Data from Observable [12].

- Stock Price (stock_price) and Stock Volume (stock_volume) data (see Fig. 6(e)) contains daily closing values and trading volumes, respectively, for 9 companies (Apple, Amazon, Bank of America, Google, Intel, JP Morgan, Microsoft, Toyota, and Tesla) over a 5 year period, January 2015 through December 2019 (1257 samples each), collected from Yahoo Finance [4].

- Average Wind Speed (climate_awnd), High Temperature (climate_tmax), and Total Precipitation (climate_prcp) data (see Fig. 6(f)) contains 10 years of daily weather values (3651 samples each) from 6 US Airports (Atlanta, New York JFK, Los Angeles,
Chicago O’Hare, Seattle-Tacoma, and Salt Lake City), collected from NOAA Climate Data Service [34].

- US Unemployment (unemployment) data (see Fig. 6(g)) are the monthly number of unemployed individuals in 14 economic sectors (e.g., agriculture, finance, health, etc.) from January 2000 through February 2010 (122 samples each). The data were collected from the US Bureau of Labor Statistics [3].

- Radio Astronomy (astro) data (see Fig. 6(b)) contains 5 spectral ‘lines’ (1947 samples each) that measure the frequency and amplitude of radio waves emitted by extraterrestrial matter (i.e., gas and dust), collected from the ALMA Science Archive [1].

5.2 Evaluation By Task

We evaluate 12 smoothing methods from Table 1 with our framework (see Fig. 1). Tasks that use the same metrics are combined to reduce space. We produce rank plots summarizing all datasets for the metrics related to those tasks. Each column is the average rank for a given data category, with the average rank across all 80 datasets (i.e., total performance) in the final column. To reduce the clutter, we only show the tracks for smoothing methods with the top 4 overall performance.

5.2.1 Retrieve Value / Determine Range

The results for Retrieve Value and Determine Range tasks, in Fig. 8, show that 3 smoothing techniques—TOPOLOGY, GAUSSIAN, and SAVITZKY-GOLAY—produced the best results. Still among those datasets, there is a distinction between results depending upon the dataset category. For example, TOPOLOGY excelled at data that are predominated by “spiky” features, e.g., climate_prcp and stock_volume. On the other hand, GAUSSIAN and SAVITZKY-GOLAY performed better on data with long-term trends, e.g., stock_price, and cyclical behaviors, e.g., climate_max. Among the worst performing techniques were all rank-based approaches, all frequency domain-based approaches, and the UNIFORM and DOUGLAS-PEUCKER subsampling approaches.

5.2.2 Compute Derived Value

The results for the Computing Derived Value task, in Fig. 9, show that the CUTOFF filter clearly outperforms all other techniques. Examining the entropy plots (not shown), it can be observed that TOPOLOGY performs similarly well on most of the data sets. Finally, the convolutional methods, GAUSSIAN, MEAN, and SAVITZKY-GOLAY, occasionally performed well. Among the worst performing techniques are all rank-based techniques, frequency domain-based techniques, excluding CUTOFF, and subsampling techniques, excluding TOPOLOGY.

5.2.3 Find Extrema / Find Anomalies

The results for Find Extrema and Find Anomalies tasks, in Fig. 10, show that DOUGLAS-PEUCKER produced the best results, with GAUSSIAN, TOPOLOGY, and, interestingly, UNIFORM occasionally performing well. Since the subsampling techniques, including UNIFORM, use a subset of the original data, it is safe to assume that at some levels of smoothing that will include peaks in the data. Among the techniques that performed poorly were once again rank-based, frequency domain-based, and convolutional techniques, excluding GAUSSIAN.

5.2.4 Characterize Distribution / Cluster: Trends

The results for the Characterize Distribution and Cluster: Trends task, in Fig. 11, show that depending upon the datasets, GAUSSIAN, TOPOLOGY, or SAVITZKY-GOLAY produce the best results, with TOPOLOGY working best on “spiky” datasets and convolutional techniques working better on those datasets with long-term trends and cyclical behaviors. Among the worst performing techniques are again all rank-based techniques, frequency domain-based techniques, and subsampling, excluding TOPOLOGY.

5.2.5 Sort / Cluster: Points

The results for the Sort and Cluster: Points tasks, in Fig. 12, show that GAUSSIAN was the best performer, followed by TOPOLOGY and SAVITZKY-GOLAY, depending upon whether the Pearson metric, in Fig. 12(a), or the Spearman metric, in Fig. 12(b), is used. Once again, the TOPOLOGY method appears to work best on “spiky” datasets, while
the convolutional methods worked better on data with long-term trends or cyclical behaviors. The worst performing techniques for these tasks are largely the same as for other tasks.

6 DISCUSSION: SMOOTHING RECOMMENDATIONS

Given our evaluation in Sect. 5 we summarize the efficacy of the techniques tested in Table 3. For these grades, we measure the frequency of a method being ranked in the top 3 for a given task across each of the 80 datasets. Grades are assigned using that frequency: A: > 75%; B: 50% – 75%; C: 25% – 50%; D: 5% – 25%. In other words, a method scoring a grade of A ranks 1st, 2nd, or 3rd in at least 75% of datasets.

General Recommendation If designing a visualization without particular concern for the data or visual analytics task, GAUSSIAN and TOPOLOGY performed well in all categories. The main difference between the 2 is that while GAUSSIAN had more A’s, TOPOLOGY has no score lower than B. The heart of which to pick really lies in the type of data being visualized. As we pointed out in our evaluation, TOPOLOGY tended to do better on data with “spiky” features, while GAUSSIAN did better with cyclical behaviors or long-term trends.

Task Specific Recommendations If the visual analytics tasks are known ahead of time, a more nuanced decision can be made about what method to use. While GAUSSIAN and TOPOLOGY did well on most tasks, CUTOFF and DOUGLAS-PUECKER performed the best on Compute Derived Value and Find Extrema/Anomalies, respectively.

Data Specific Recommendations If the data are available ahead of time, analyzing them with our framework to select the best smoothing method is recommended. In addition to GAUSSIAN and TOPOLOGY, several methods generate better results in limited situations, particularly if the tasks are known as well. These methods include MEDIAN, SAVITZKY-GOLAY, MEAN, CUTOFF, and DOUGLAS-PUECKER.

Methods to Largely Avoid Several methods performed poorly across the board. These include MIN, MAX, BUTTERWORTH, CHEBYSHEV, and UNIFORM subsampling. These methods rarely performed in the top 3. They should only be used when there is a very specific reason to do so, which should not be a problem as most of these techniques are rarely used anyways. However, this finding is particularly relevant for UNIFORM subsampling, as it is essentially the default methodology used for data reduction.

6.1 Conclusions

In conclusion, we have presented and demonstrated a framework for evaluating line chart smoothing in the context of the visual analytics tasks being performed. There remain several study limitations and future works.

Perceptual Effects and User-based Validation Our study considers only the effects of data modification in the evaluation of smoothing effectiveness. There may be additional perceptual effects that make the results of some techniques better or worse than others. We considered performing a user study to validate our framework further. However, it became quickly apparent that the scale of such a study would be impractical. As an example, testing 12 smoothing techniques, across 8 tasks, 80 datasets, and 20 different smoothing levels would require in excess of 150K experimental stimuli.

Feature Types and Representing Lost Information Throughout our analysis, we discussed “spiky”, cyclical, and long-term trends in data. These categories are ill-defined, and a broader study of feature types that appear in line charts would be valuable to the community. Furthermore, smoothing removes information from the representation of the line chart, introducing uncertainty. One additional direction of future work would be to use this framework to model and represent the uncertainty by considering the context of the visual analytics task.

There’s No Accounting for Taste Aesthetics play an important role in visualization design. Without a good aesthetic, users are less likely to remember what they see [11]. Although TOPOLOGY and GAUSSIAN were largely the most effective techniques, their aesthetics are quite different, “spiky” for TOPOLOGY and smooth for GAUSSIAN. Our framework completely ignores aesthetic in its recommendation, in part because aesthetic is both art and science, thus difficult to model mathematically or algorithmically.

ACKNOWLEDGMENTS

We would like to thank Bei Wang and Ashley Suh for providing valuable feedback on this project. This work was supported in part by a grant from the National Science Foundation (IIS-1845204).
[47] Y. Shao, R. S. Lunetta, B. Wheeler, J. S. Iiames, and J. B. Campbell. An evaluation of time-series smoothing algorithms for land-cover classifications using modis-ndvi multi-temporal data. *Remote Sensing of Environment*, 174:258–265, 2016. doi: 10.1016/j.rse.2015.12.023

[48] H. Song and D. Albers Szafir. Where’s my data? evaluating visualizations with missing data. *IEEE Transactions on Visualization and Computer Graphics*, PP:1–1, 2018. doi: 10.1109/TVCG.2018.2864914

[49] A. Suh, C. Salgado, M. Hajij, and P. Rosen. Topolines: Topological smoothing for line charts. *arXiv preprint arXiv:1906.09457*, 2019.

[50] D. A. Szafir. Modeling color difference for visualization design. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):392–401, 2018. doi: 10.1109/TVCG.2017.2744359

[51] M. Templ, A. Alfons, and P. Filzmoser. Exploring incomplete data using visualization techniques. *Advances in Data Analysis and Classification*, 6:29–47, 2012. doi: 10.1007/s11634-011-0102-y

[52] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1986. doi: 10.1119/1.14057

[53] Y. Wang, F. Han, L. Zhu, O. Deussen, and B. Chen. Line graph or scatter plot? automatic selection of methods for visualizing trends in time series. *IEEE Transactions on Visualization and Computer Graphics*, 24(2):1141–1154, 2018. doi: 10.1109/TVCG.2017.2653106

[54] Y. Wang, Z. Wang, L. Zhu, J. Zhang, C. Fu, Z. Cheng, C. Tu, and B. Chen. Is there a robust technique for selecting aspect ratios in line charts? *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3096–3110, 2018. doi: 10.1109/TVCG.2017.2787113

[55] J. Zacks and B. Tversky. Bars and lines: A study of graphic communication. *Memory & cognition*, 27(6):1073–1079, 1999. doi: 10.3758/ BF03201236