Main takeaways

- Smoothing techniques (!= curve interpolation) as a kind of data processing step;
- Results are dataset dependent as well;
- Tasks for (single) line charts: Retrieve Value, Determine Range, Compute Derived (or Aggregated) Value, Find Extrema, Find Anomalies, Characterize Distribution, Sort (points), and Cluster (points or trends);
- General recommendations: Gaussian (smooth line) and Topology ("spiky" line);
- Task-specific recommendations: Cutoff and Douglas-Peucker (Compute Derived Value and Find Extreme/Anomalies, respectively);
- To avoid: Min, Max, Butterworth, Chebyshev, and Uniform.

LineSmooth: An Analytical Framework for Evaluating the Effectiveness of Smoothing Techniques on Line Charts

Interpolation method: cardinal spline (see references at the end of this page)

Paul Rosen and Ghulam Jilani Quadri

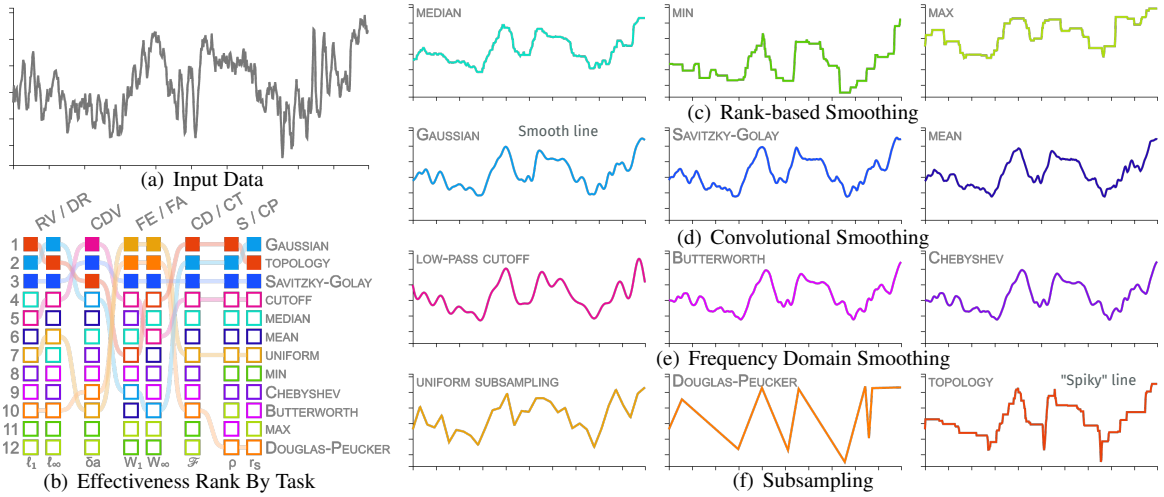


Fig. 1. Example of the (a) EEG Channel 10 (500 samples) dataset with (c-f) 4 classes / 12 techniques of line chart smoothing applied. (b) Our framework provides the capability to rank smoothing methods by their efficacy in visual analytics tasks. To form the rankings, the 12 smoothing examples are calibrated to have similar visual complexity. Then, 8 measures of effectiveness, described in Sect. 4.1, are calculated and ordered from best to worst. The results show that for this example, the GAUSSIAN, TOPOLOGY, and SAVITZKY-GOLAY methods are the 3 best techniques for the Retrieve Value (RV), Determine Range (DR), Characterize Distribution (CD), Cluster Trends (CT), Sort (S), and Cluster Points (CP) tasks. For the Compute Derived Value (CDV) task, CUTOFF, SAVITZKY-GOLAY, and TOPOLOGY perform best, respectively. Finally, for the Find Extrema (FE) and Find Anamolies (FA) tasks, UNIFORM subsampling, DOUGLAS-PEUCKER, and SAVITZKY-GOLAY perform best. Clearly, no single method is best for all visual analytics tasks, but for this particular data, SAVITZKY-GOLAY, being in the top 3 for all tasks, would be a reasoned choice.

Abstract—We present a comprehensive framework for evaluating line chart smoothing methods under a variety of visual analytics tasks. Line charts are commonly used to visualize a series of data samples. When the number of samples is large, or the data are noisy, smoothing can be applied to make the signal more apparent. However, there are a wide variety of smoothing techniques available, and the effectiveness of each depends upon both nature of the data and the visual analytics task at hand. To date, the visualization community lacks a summary work for analyzing and classifying the various smoothing methods available. In this paper, we establish a framework, based on 8 measures of the line smoothing effectiveness tied to 8 low-level visual analytics tasks. We then analyze 12 methods coming from 4 commonly used classes of line chart smoothing—rank filters, convolutional filters, frequency domain filters, and subsampling. The results show that while no method is ideal for all situations, certain methods, such as GAUSSIAN filters and TOPOLOGY-based subsampling, perform well in general. Other methods, such as low-pass CUTOFF filters and DOUGLAS-PEUCKER subsampling, perform well for specific visual analytics tasks. Almost as importantly, our framework demonstrates that several methods, including the commonly used UNIFORM subsampling, produce low-quality results, and should, therefore, be avoided, if possible.

Index Terms—Line chart, data smoothing, time-series.

1 INTRODUCTION

Line charts, which date back to William Playfair [37], 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 [8].

When line chart data are noisy, e.g., see Fig. 1(a), visualization designers can turn to *smoothing* to reduce the visual clutter. However, there are many techniques available (see Fig. 1(c-f)), and while the results they produce may look similar, each preserves different properties of the data. For example, rank-based (see Fig. 1(c)) and convolutional smoothing methods (see Fig. 1(d)) preserve local properties, such as local trends, while frequency-domain smoothing (see Fig. 1(e)) and subsampling (see Fig. 1(f)) preserve global properties, such as the most prominent peaks in the data.

To preserve some properties of the input data, each smoothing technique must also *lose information*, which can have a negative impact on the utility of the resulting data. To further complicate matters, the importance of the lost information can be influenced by both the data being used and the visual analytics tasks being performed. To date, the visualization community lacks a comprehensive framework for measur-

- *Paul Rosen is with the University of South Florida. E-mail: prosen@usf.edu.*
- *Ghulam Jilani Quadri is with the University of South Florida. E-mail: ghulamjilani@usf.edu.*

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.xxxx/TVCG.201x.xxxxxxx

https://en.wikipedia.org/wiki/Cubic_Hermite_spline#Cardinal_spline
https://github.com/USFDataVisualization/LineSmooth/blob/master/pages/javascript/fig_teaser.js#L49
https://github.com/d3/d3-shape/blob/main/README.md#curveCardinal (`curveCardinal`)
https://vega.github.io/vega/docs/marks/line/ (`cardinal`)

ing the influence of line chart smoothing on a range of visual analytics tasks. Task and dataset dependent

In this paper, we present an analytical framework for measuring the effectiveness of various smoothing techniques under 8 different low-level visual analytics tasks performed on line charts. We define a taxonomy of 4 classes of line chart smoothing and evaluate a total of 12 commonly available techniques on 80 datasets from 13 categories.

Through our analysis, we show that there is no single smoothing technique that is ideal for all visual analytics tasks. Furthermore, we show that the efficacy of each technique can vary by the datasets being analyzed. Nevertheless, we identify specific methods that consistently perform well, in particular GAUSSIAN filters and TOPOLOGY-based subsampling [40]. In other cases, some methods are particularly well suited for specific tasks, e.g., low-pass CUTOFF filters and DOUGLAS-PEUCKER subsampling [17] are well suited for computing a derived value and finding extrema, respectively. Finally, we identify several methods, including the commonly used UNIFORM subsampling, which perform consistently poorly.

Visualization designers can use this framework and results of this paper to either: (1) select a smoothing technique, which is most effective in general or most effective for the tasks their users perform; (2) evaluate their data to select the technique that is specifically most effective; or (3) to understand how much error is introduced as they increase the level of smoothing used in their visualizations. An interactive version of our framework is at <https://usfdatavisualization.github.io/LineSmoothDemo>.

2 PRIOR WORK

We discuss prior work in the context of analytical tasks performed using line charts, decision-making with line charts, and line chart smoothing.

2.1 Task Efficacy

Line charts, which are traditionally used to visualize time-series and continuous 1D data [30, 50, 55], have been studied in the context of a variety of low-level visual analytics tasks [4]. A recent multi-chart experimental study found that line charts are significantly more accurate than other charts for the tasks of correlation and, to a lesser extent, finding extrema, characterizing distributions, and filtering [43]. Even so, line charts are used for a wider variety of visual analytics tasks.

Comparison Tasks Early work on horizon graphs [42] investigated their effectiveness as compared to line charts in a comparison task [24]. The work identified space-accuracy trade-off that could be used to optimize perception between the two. Another study compared line charts to horizon graphs and colorfields for similarity assessment [21]. The study showed that deformations in the data are perceived differently depending on the visualization, and, in particular, line charts are more sensitive to changes in amplitude than position. 1D heatmap

Statistical Tasks Line charts are used in many forms of statistical analysis [33]. Perception-based experiments that measure user's judgments concluded that line charts have low-to-medium precision on estimating correlation [23, 27]. More generally, when considering aggregation tasks, it has been shown that line charts are effective at finding minima and maxima and determining value range while falling short on determining the average, spread, and outliers in the data [2]. More specifically, when calculating averages, colorfields have been shown to outperform line charts [14].

Trend Assessment Another common task attended to with line charts is trend assessment. It has been shown that line charts are, in general, better at trend assessment than scatterplots and bar charts, particularly for nonlinear trends [7]. However, when outliers are introduced into the data, the trends in their estimates begin to diverge from standard regression models [15]. Furthermore, when data are noisy, trends in the data are easier to identify using scatterplots [52].

Visual Encoding, Layout, and Interaction The visual encodings, layout, and interaction with line charts can have an impact on their efficacy. For example, color is an important visual encoding. For line charts, it has been shown that color difference varies inversely with thickness [48]. In other words, to be effective, a light-colored line must be thicker than dark-colored ones. Concerning layout, the efficacy of line charts is subject to the choice of aspect ratio, which can be automatically optimized for a chart [53]. 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 [26]. 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 [34]. Finally, adding user interaction to line charts can enhance the user experience without a loss to efficacy [1].

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 [41].

Missing Data Another decision-making challenge in visualization is when data are missing; one needs to impute the missing data into the visualization [49]. 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 [18]. 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 [47].

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 [35]. Another example is when missing data are misleadingly inserted into a visualization, e.g., assigned arbitrary values, user performance can go down significantly [18]. To address this weakness, multi-view systems have been proposed to assist in time-series data quality checking [5].

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 [46], and image processing community, e.g., Chen and Yeh developed a quantitative evaluation for edge-preservation in image smoothing [12]. 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 [40]. Nevertheless, no comprehensive framework and evaluation, such as the one we are introducing in this paper, exists.

3 TAXONOMY OF LINE CHART SMOOTHING APPROACHES

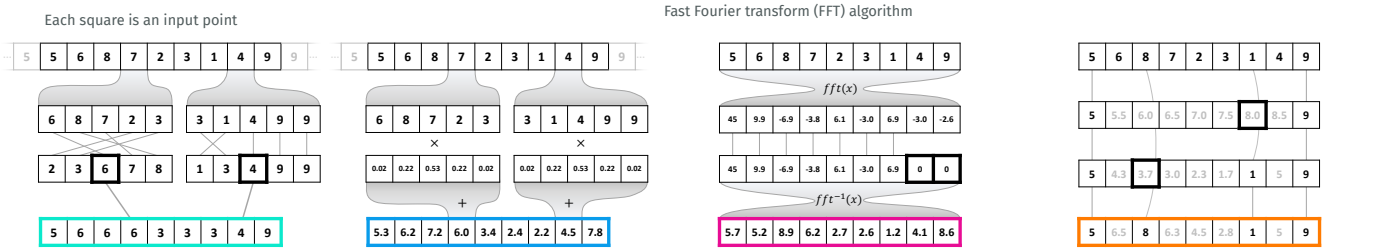
We discuss 4 classes of smoothing that can be used on line charts. They can be broadly broken down into methods that consider local neighborhoods of data or global structures when determining the output. A summary of the 12 smoothing techniques analyzed in this paper can be found in Table 1. Each technique preserves some particular 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.

Frequency domain: it refers to the analysis of mathematical functions or signals with respect to frequency, rather than time

https://en.wikipedia.org/wiki/Frequency_domain



(a) Rank Filter: MEDIAN (b) Convolutional Filter: GAUSSIAN (c) Frequency Domain: CUTOFF filter (d) Subsampling: DOUGLAS-PEUCKER

Fig. 2. Illustration of (a-b) local and (c-d) global smoothing methods for line charts. (a) Starting with the input data (top), the MEDIAN filter extracts a window (2nd row), sorts the window (3rd row), and selects the median value for output (4th row). (b) The GAUSSIAN filter similarly extracts a local window (2nd row). However, the window has a convolution applied based upon a normal distribution (3rd row), which is used for output (4th row). (c) The low-pass CUTOFF filter converts data into the frequency domain (2nd row), zeros out high-frequency components (3rd row), and converts the smoothed result back into the spatial domain (4th row). (d) The DOUGLAS-PEUCKER method subsamples the input data by iteratively selecting (2nd and 3rd rows) the points with the largest error to insert into the smoothed output (4th row). All techniques are colored by type (see Fig. 1).

It presents itself as sparsely occurring white and black pixels

https://en.wikipedia.org/wiki/Salt-and-pepper_noise

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 [6], 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 $\mathcal{O}(n \cdot w \log w)$. The operation can be optimized by using a sliding window to achieve $\mathcal{O}(n \log w)$ in the general case [20] and $\mathcal{O}(n)$ in limited cases [36].

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 [29]. It weights the input neighborhood using a normal distribution. The smoothing level is increased or decreased by adjusting the standard deviation, σ, 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 σ 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 can be used to improve performance to $\mathcal{O}(n)$ complexity.

Finally, SAVITZKY-GOLAY [44] (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.

Frequency: the number of occurrences of a repeating event per unit of time

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) [13]. 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 [10] (see Fig. 1(e)(middle)) and CHEBYSHEV filter [39] (see Fig. 1(e)(right)). The CUTOFF filter is an idealized function that cannot be implemented in an electronic circuit, while the BUTTERWORTH and CHEBYSHEV filters can. Practically speaking, these methods differ from the CUTOFF filter in that they provide a gradual ramp-down of the cutoff frequency.

decrease

Table 1. Summary of Smoothing Algorithms Analyzed.

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	$\mathcal{O}(n + c \log c)$

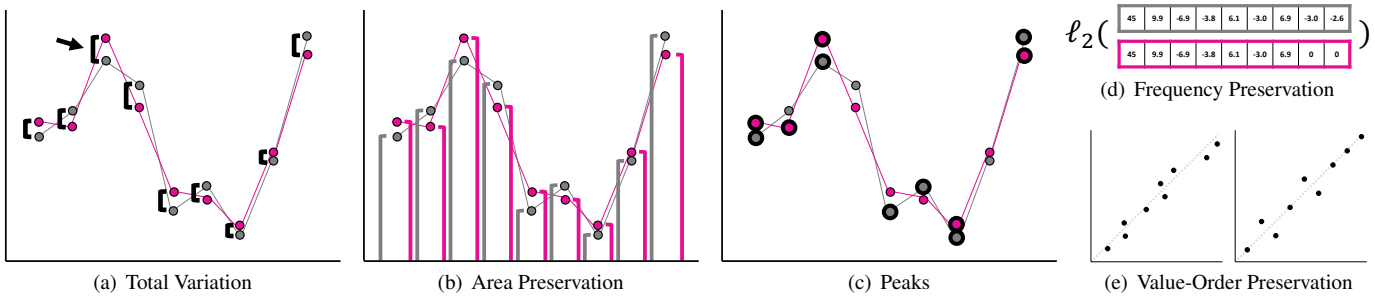
n: number of input points; w: window size; σ: standard deviation of a normal distribution; s: number of output samples; c: number of critical points (i.e., local minimum or maximum)

Stencil: a thin sheet, either perforated or using some other technique, with which a pattern may be produced upon a surface

https://en.wiktionary.org/wiki/stencil

Low-pass filter: a filter that passes signals with a frequency lower than a selected cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency

https://en.wikipedia.org/wiki/Low-pass_filter



(a) Total Variation (b) Area Preservation (c) Peaks (d) Frequency Preservation (e) Value-Order Preservation

Fig. 3. Illustration of the 5 types of effectiveness measurement used in our approach. The example is based upon the low-pass CUTOFF filter from Fig. 2(c). (a) shows the total variation, where ℓ_1 is the sum of the black brackets, and ℓ_∞ is the value of the largest (see arrow). (b) measures the change in the area, δa, by taking the difference between the sum of the grey bars and the sum of the pink bars. (c) illustrates the peaks in the data in bold. The W_1 measures the total difference between peaks, while W_∞ measures the largest variation. (d) shows the frequency domain of the input and smoothed data. $\mathscr{F}$ measures the L^2-norm of the difference between these 2 vectors. (e) illustrates the (left) value preservation, ρ, and (right) order preservation, r_s, of the smoothed line chart. Proximity to the diagonal indicates how well the value/order is preserved.

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 [45]. Computationally, UNIFORM subsampling is very efficient, only $\mathscr{O}(n+s)$, 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 [17,38] (see Fig. 1(f)(middle)) is an example that establishes a priority queue of points by optimizing the L^∞-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 $\mathscr{O}(n^2)$, while the average complexity is $\mathscr{O}(n \log n)$.

An additional nonuniform subsampling approach, the TOPOLOGY filter (see Fig. 1(f)(right)), uses techniques from Topological Data Analysis to smooth data in a way that retains significant peaks and minimizes error [40]. 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 $\mathscr{O}(n + c \log c)$, 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, $X = \{x_0, x_1, .., x_i, .., x_n\}$, and the smoothed data, $Y = \{y_0, y_1, .., y_i, .., y_n\}$. 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 measures consider 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^∞-norm.

The L^1-norm, ℓ_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^∞-norm, ℓ_∞, 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^∞-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, δa, 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:

$$\delta a(X,Y) = \left| \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} y_i \right| \tag{3}$$

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

Monotonic (or isotonic) regression: the fitted line is non-decreasing (or non-increasing) everywhere, and lies as close to the observations as possible

https://en.wikipedia.org/wiki/Isotonic_regression

Bijection: a function between the elements of two sets, where each element of one set is paired with exactly one element of the other set, and each element of the other set is paired with exactly one element of the first set

https://en.wikipedia.org/wiki/Bijection

The infimum indicates that we are only interested in the smallest cost

https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html#what-is-wasserstein-distance

use techniques from Topological Data Analysis [19]. First, the local minima and maxima of the original and smoothed data are calculated and paired in a process described in detail in [40]. The pairs are placed into 2 sets $\overline{X}$ and $\overline{Y}$[1], and let η be a bijection between the two 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, is:

Infimum or greatest lower bound

https://en.wikipedia.org/wiki/Infimum_and_supremum

$$W_1(\overline{X},\overline{Y}) = \inf_{\eta:\overline{X}\to\overline{Y}} \sum_{\overline{x}\in\overline{X}} \|\overline{x} - \eta(\overline{x})\|_1 \quad (4)$$

Bijection/Bijective function

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

Supremum or least upper bound

$$W_\infty(\overline{X},\overline{Y}) = \inf_{\eta:\overline{X}\to\overline{Y}} \sup_{\overline{x}\in\overline{X}} \|\overline{x} - \eta(\overline{x})\|_\infty \quad (5)$$

b

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, $\mathscr{F}$, 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:

$$\mathscr{F}(F_X,F_Y) = \sqrt{\sum_{k=1}^{n} \left(F_{X,k} - F_{Y,k}\right)^2}, \quad (6)$$

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, ρ, can be employed. Fig. 3(e)(left) illustrates the value relationship by placing points at (x_i, y_i). Intuitively, the ρ measures the proximity of the points to the diagonal in grey. In order to treat ρ consistently with other measures, we modify it, such that 0 is a perfect positive correlation, and 2 is a perfect negative correlation. The modified ρ is:

$$\rho(X,Y) = 1 - \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} \quad (7)$$

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:

$$r_s(X,Y) = \rho(rank(X), rank(Y)) \quad (8)$$

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

4.2 Low-Level Task Taxonomy for Line Charts

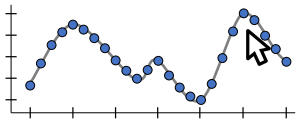
To determine relevant visual analytics tasks, we adapt the low-level task taxonomy of Amar et al. [4] to line charts. For each task, we provide a brief description and an example query. Finally, we relate each of these tasks to 1 or 2 of the metrics from the previous section, which is summarized in Table 2.

[1]For technical reasons, all diagonal points $(\overline{x},\overline{x})$ are added to make the cardinality infinite [28].

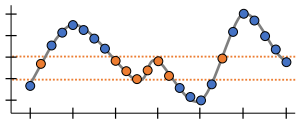
Table 2. Matrix of Tasks and Metrics

For single line charts

	Total Value Variation (ℓ_1)	Maximum Value Variation (ℓ_∞)	Area Preservation (δa)	Total Peak Variation (W_1)	Maximum Peak Variation (W_∞)	Frequency Preservation ($\mathscr{F}$)	Value Preservation (ρ)	Order Preservation (r_s)
Retrieve Value	✓	✓						
Determine Range	✓	✓						
Compute Derived Value			✓					
Find Extrema				✓	✓			
Find Anomalies				✓	✓			
Characterize Distribution						✓		
Sort							✓	✓
Cluster (Trends / Points)						✓	✓	✓

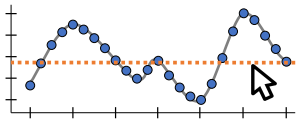


The **Retrieve Value** task is focused on finding the function value at an exact location in a given dataset/chart. For example, using Fig. 4, "What is the stock price on Mar '15?" (answer: ~ 4.5). The accuracy of retrieving a value is dependent upon how closely chart values match the data value, which mirrors the total/maximum value variation measures (see Sect. 4.1.1 and Fig. 3(a)). Measuring the total difference between the smoothed data and the original data, in other words, the L^1-norm (Eq. 1), provides an *average case* performance for the retrieve value task. By measuring the maximum difference between the smoothed and original function, in other words, the L^∞-norm (Eq. 2), the *worst case* performance can be calculated.



For the **Determine Range** task, specific criteria, e.g., a range of values, are provided for identifying data points, e.g., the dates/times that are within that range. Using Fig. 4, an example is, "What months saw values between 3 and 4?" (answer: Oct '14, Nov '14, Feb '15, Mar '15). Accuracy in performing this task is highly dependent upon the criteria provided. Nevertheless, generally, it is important that the values of the data closely reflect those of the input data, in other words, the total/maximum value variation (see Sect. 4.1.1 and Fig. 3(a)). The *average case* is measured using the L^1-norm (Eq. 1). The *worst case* is measured using the L^∞-norm (Eq. 2).

or Compute Aggregated Value



The **Compute Derived Value** task focuses on computing an aggregate, such as the average or total value of a function. For example, using Fig. 4, "What is the average stock price from Oct '14 to Apr '15?" (answer: ~ 3). The task accuracy is mostly dependent on how well the line chart globally (i.e., the sum of all values) matches the input data. The task essentially requires the user to visually determine an integral of the data, which is equivalent to the area preservation measure (see Sect. 4.1.2 and Fig. 3(b)). The area preservation measure, δa, provides the *average case* performance by measuring the difference in integrals between the original and smoothed data.

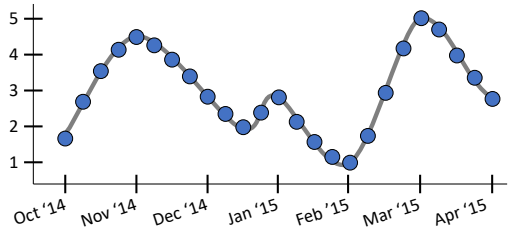
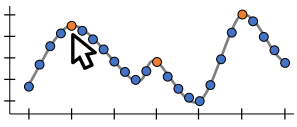
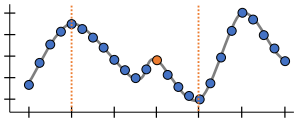


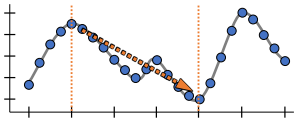
Fig. 4. Example "stock price" line chart for demonstrating task queries.



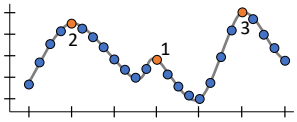
The **Find Extrema** task is concerned with finding local minima and maxima (i.e., valleys and peaks) in the data. For example, using Fig. 4, "What are the dates/values of all of the peaks in the data?" (answer: Nov '14/4.5, Jan '15/3, Mar '15/5). The accuracy of the task depends upon the peaks and/or 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.



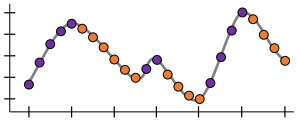
The **Find Anomalies** task involves looking for values that do not conform to the overall trend in the data. For example, "Between Nov '14 and Feb '15, what month, if any, does not follow the data trend?" (answer: Dec '14/Jan '15). The task of finding anomalies is similar to finding extrema, in that 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. 4.1.4 and Fig. 3(d)) from Eq. 6.



The **Sort** task asks users to, given 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(e)) provide 2 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. 4.1.4 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(e)), Spearman Rank Correlation (see Eq. 8) and Pearson Correlation Coefficient (see Eq. 7) for clustering individual points.

Tasks Not Considered Amar et al. [4] 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 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 [41]. 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 [41] 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, ℓ_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 $ApEx$

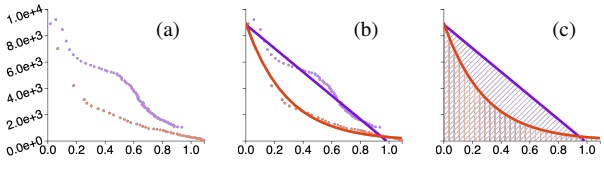


Fig. 5. Entropy plot for the EEG Channel 10 (500 samples) dataset with TOPOLOGY (in red) and CHEBYSHEV (in purple). The L^1-norm is plotted vertically, and the $ApEx$ horizontally. (a) The methods are sampled at 100 levels to capture a range of $ApEx$ values. (b) Linear and logarithmic regression are used to obtain a best-fit model. (c) Mathematical integration is used to capture the area under the curve. Methods with smaller areas produce less error, thus performing better. In this case, TOPOLOGY is ~ 2500, while CHEBYSHEV is ~ 4350, making TOPOLOGY the more effective method.

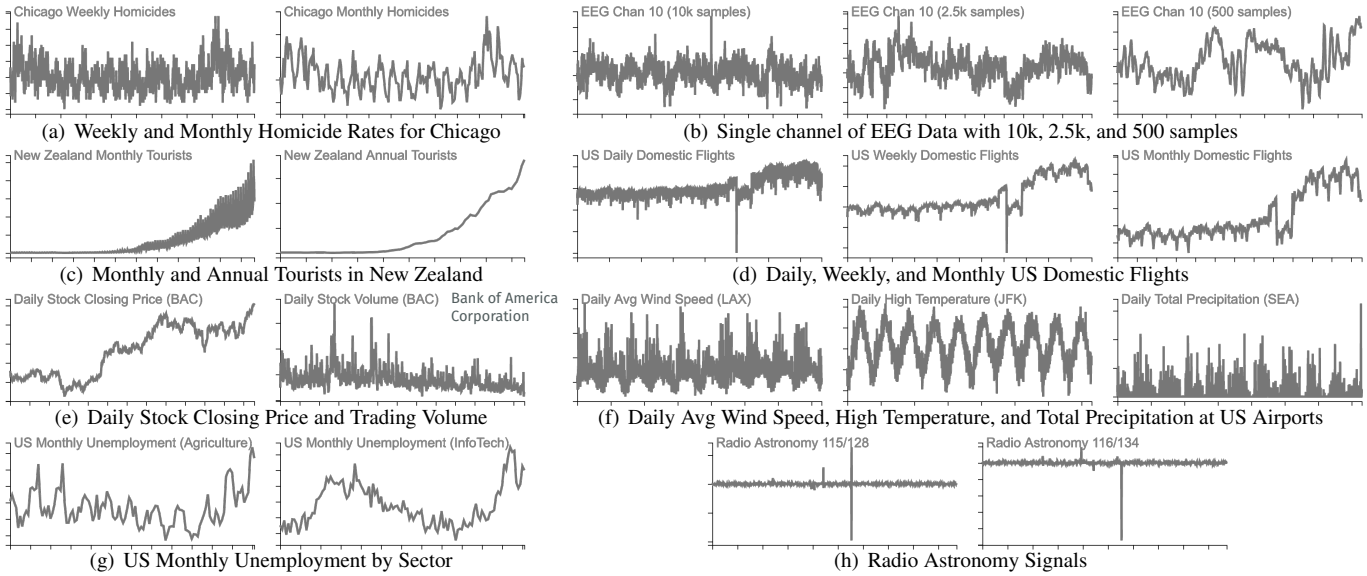


Fig. 6. Examples of datasets used to analyze smoothing techniques.

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)[2] [25]. 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 ~ 2500 for TOPOLOGY and ~ 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 best (top) to worst (bottom). In Fig. 7, the average rank is the final column. The average rank of TOPOLOGY is $\frac{4+2+2+1+1+1}{6} = 1.8$, while the GAUSSIAN is 2.0, SAVITZKY-GOLAY is 3.5, etc.

[2]Using IRLS helps to minimize the impact of outliers

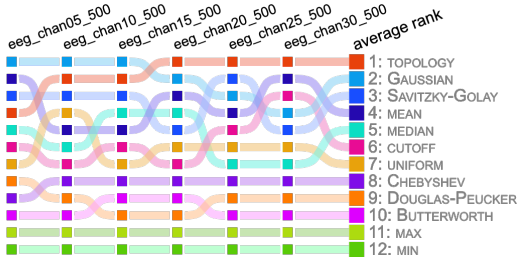


Fig. 7. Rank plot of the L^1-norm for all EEG (500 samples) datasets. Each column ranks the techniques, from best (top) to worst (bottom), on a different dataset, with the average rank in the final column. The result shows that for this data, on average, TOPOLOGY, GAUSSIAN, and SAVITZKY-GOLAY are the most effective techniques, respectively.

IRLS: a method to solve certain optimization problems (solver) with objective functions of the form of a p-norm by an iterative method

https://en.wikipedia.org/wiki/Iteratively_reweighted_least_squares
https://github.com/PyLops/pylops

5 RESULTS

The source code and data for our framework and evaluation is available at <https://github.com/USFDataVisualization/LineSmooth>, and an interactive version of our evaluation is at <https://usfdatavisualization.github.io/LineSmoothDemo>.

5.1 Data Sources

We evaluate 80 datasets in 13 categories from 8 data sources (see Fig. 6). Data were selected that contained typical qualities, such as long-term trends (e.g., stock prices), cyclical behaviors (e.g., daily temperatures), or important spikes (e.g., a stock market crash).

- 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 [11].
- 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 [16].
- 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 [32].
- US Domestic Flights (*flights*) data (see Fig. 6(d)) contains the number of daily, weekly, and monthly (7671, 1095, and 252 samples, respectively) number of US flights from January 1, 1988 through December 31, 2008. Data collected from Observable [9].
- 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 [54].
- 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 [31].

- 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 [51].

- Radio Astronomy (*astro*) data (see Fig. 6(h)) 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 [3].

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.

Similar conclusions with "5.2.4 Characterize Distribution / Cluster: Trends" and "5.2.5 Sort / Cluster: Points"

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_tmax*. Among the worst performing techniques were all rank-based approaches, all frequency domain-based approaches, and the UNIFORM and DOUGLAS-PEUCKER subsampling approaches.

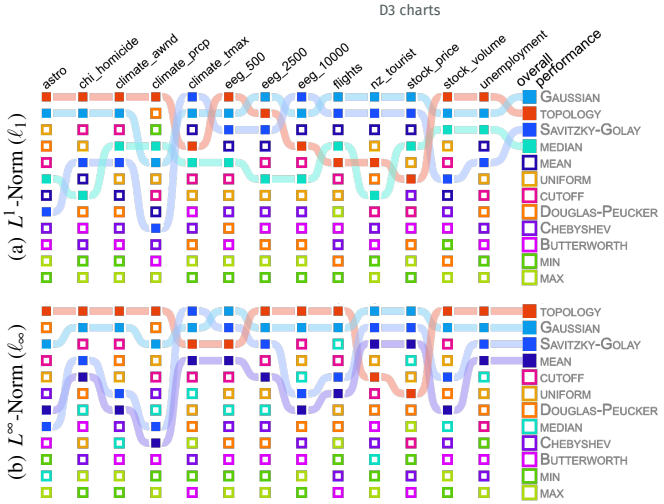


Fig. 8. Average ranking for the **Retrieve Value** and **Determine Range** tasks across all datasets using the (a) L^1-norm and (b) L^∞-norm. Both metrics show that TOPOLOGY and GAUSSIAN were primarily the best methods to use, while SAVITZKY-GOLAY occasionally performs well.

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.

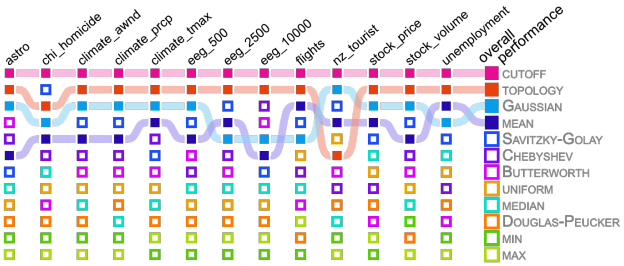


Fig. 9. Average ranking for the **Compute Derived Value** task across all datasets using the area preservation metric, δa. The results show that CUTOFF is clearly the top method, followed by TOPOLOGY. GAUSSIAN, MEAN, and SAVITZKY-GOLAY occasionally perform well.

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 for some levels of smoothing, peaks will be included in the output. Among the techniques that performed poorly were once again rank-based, frequency domain-based, and convolutional techniques, excluding GAUSSIAN.

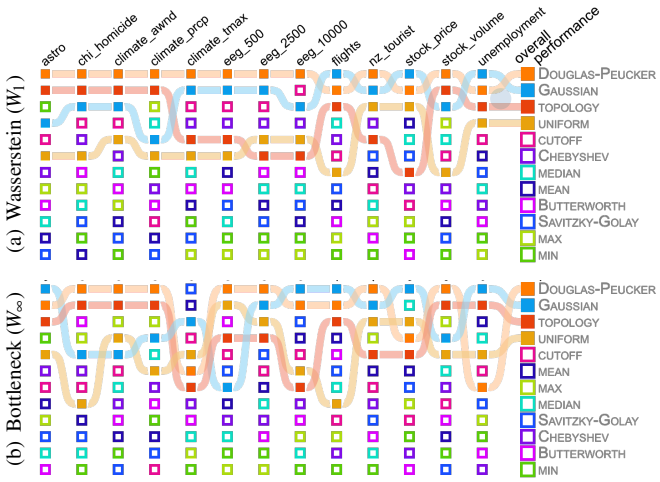


Fig. 10. Average ranking for the **Find Extrema** and **Find Anomalies** tasks across all datasets using the (a) average case, Wasserstein distance, and (b) worst case, Bottleneck distance. Both metrics show DOUGLAS-PEUCKER performing best, followed by GAUSSIAN, TOPOLOGY, and UNIFORM, in order.

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.

Similar conclusions with "5.2.1 Retrieve Value / Determine Range"

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

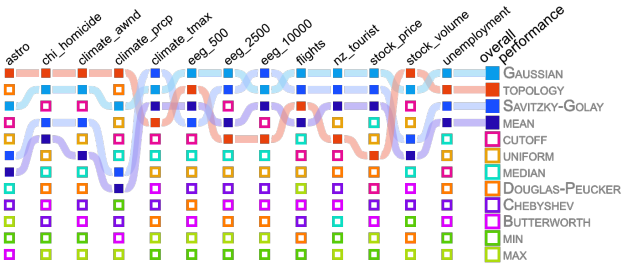


Fig. 11. Average ranking for the **Characterize Distribution** and **Cluster: Trends** tasks across all datasets using the frequency preservation metric, $\mathscr{F}$. The results show that depending upon the type of data, GAUSSIAN, TOPOLOGY, and SAVITZKY-GOLAY methods produce the best results.

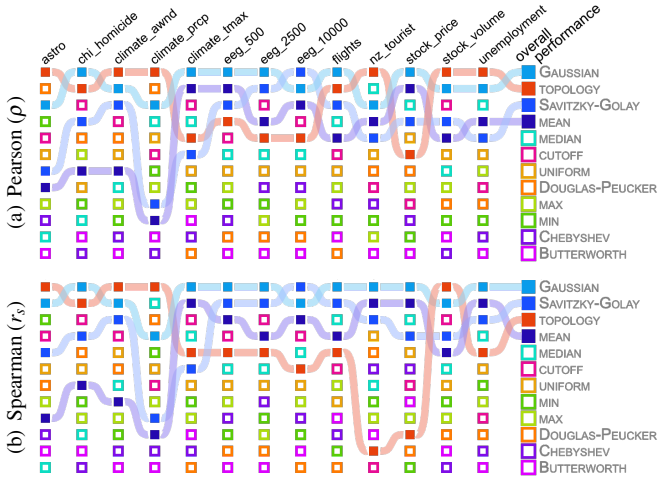


Fig. 12. Average ranking for the **Sort** and **Cluster: Points** tasks across all datasets using the (a) Pearson Correlation Coefficient, ρ, and (b) Spearman Rank Correlation, r_s. Both metrics show GAUSSIAN performing best, followed by TOPOLOGY and SAVITZKY-GOLAY.

+ MEAN occasionally performing well

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-PEUCKER 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-PEUCKER.

Table 3. Grades for the efficacy of different smoothing methods are calculated by the frequency of being ranked in the top 3 for all 80 datasets. A: $> 75\%$; B: $50\% - 75\%$; C: $25\% - 50\%$; D: $5\% - 25\%$

	Retrieve Value	Determine Range	Compute Derived Value	Find Extrema	Find Anomalies	Characterize Distribution	Sort	Cluster: Trends	Cluster: Points
MEDIAN	D	D	–	D	D	D	C	D	C
MIN	–	–	–	–	–	–	–	–	–
MAX	–	–	–	D	D	–	–	–	–
GAUSSIAN	**A**	**A**	C	**B**	**B**	**A**	**A**	**A**	**A**
SAVITZKY-GOLAY	C	C	D	D	D	**B**	C	**B**	C
MEAN	C	C	D	D	D	C	C	C	C
CUTOFF	D	D	**A**	D	D	C	D	C	D
BUTTERWORTH	–	–	D	D	D	–	–	–	–
CHEBYSHEV	–	–	D	–	–	–	–	–	–
UNIFORM	–	–	–	D	D	–	–	–	–
DOUGLAS-PEUCKER	D	D	–	**A**	**A**	D	D	D	D
TOPOLOGY	**B**	**B**	**A**	**B**	**B**	**B**	**B**	**B**	**B**

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 [22].

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 while 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 [8]. 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 input on this project. This work was supported in part by a grant from the National Science Foundation (IIS-1845204).

REFERENCES

[1] M. Adnan, M. Just, and L. Baillie. Investigating time series visualisations to improve the user experience. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 5444–5455, 2016. doi: 10.1145/2858036.2858300

[2] D. Albers, M. Correll, and M. Gleicher. Task-driven evaluation of aggregation in time series visualization. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 551–560, 2014. doi: 10.1145/2556288.2557200

[3] Alma Science Archive. `http://almascience.nrao.edu/aq/`, Apr 2020.

[4] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization (InfoVis)*, pp. 111–117, 2005. doi: 10.1109/INFVIS.2005.1532136

[5] C. Arbesser, F. Spechtenhauser, T. Mühlbacher, and H. Piringer. Visplause: Visual data quality assessment of many time series using plausibility checks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):641–650, 2017. doi: 10.1109/TVCG.2016.2598592

[6] G. R. Arce. *Nonlinear signal processing: a statistical approach*. John Wiley & Sons, 2005. doi: 10.1002/0471691852

[7] L. A. Best, L. D. Smith, and D. A. Stubbs. Perception of linear and nonlinear trends: using slope and curvature information to make trend discriminations. *Perceptual and Motor Skills*, 2007. doi: 10.2466/pms.104.3.707-721

[8] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister. What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2306–2315, 2013. doi: 10.1109/TVCG.2013.234

[9] M. Bostack. Observable: Zoomable Area Chart. `https://observablehq.com/@d3/zoomable-area-chart`, Apr 2020.

[10] S. Butterworth et al. On the theory of filter amplifiers. *Wireless Engineer*, 7(6):536–541, 1930.

[11] Chicago Police Department. Crimes - 2001 to present: City of Chicago: Data Portal. `https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2`, Apr 2020.

[12] R. T. Chin and C.-L. Yeh. Quantitative evaluation of some edge-preserving noise-smoothing techniques. *Computer Vision, Graphics, and Image Processing*, 23(1):67–91, 1983. doi: 10.1016/0734-189X(83)90054-3

[13] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. doi: 10.1090/S0025-5718-1965-0178586-1

[14] M. Correll, D. Albers, S. Franconeri, and M. Gleicher. Comparing averages in time series data. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 1095–1104, 2012. doi: 10.1145/2207676.2208556

[15] M. Correll and J. Heer. Regression by eye: Estimating trends in bivariate visualizations. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 1387–1396, 2017. doi: 10.1145/3025453.3025922

[16] A. Delorme. EEG / ERP Public Database. `https://sccn.ucsd.edu/~arno/fam2data/publicly_available_EEG_data.html`, Apr 2020.

[17] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973. doi: 10.3138/FM57-6770-U75U-7727

[18] C. Eaton, C. Plaisant, and T. Drizd. Visualizing missing data: Graph interpretation user study. In *IFIP Conference on Human-Computer Interaction*, pp. 861–872. Springer, 2005. doi: 10.1007/11555261_68

[19] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, Providence, RI, USA, 2010. doi: 10.1090/mbk/069

[20] J. Gil and M. Werman. Computing 2-d min, median, and max filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):504–507, 1993. doi: 10.1109/34.211471

[21] A. Gogolou, T. Tsandilas, T. Palpanas, and A. Bezerianos. Comparing similarity perception in time series visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):523–533, 2019. doi: 10.1109/TVCG.2018.2865077

[22] C. D. Hansen and C. R. Johnson. *Visualization handbook*. Elsevier, 2011.

[23] L. Harrison, F. Yang, S. Franconeri, and R. Chang. Ranking visualizations of correlation using weber's law. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1943–1952, 2014. doi: 10.1109/TVCG.2014.2346979

[24] J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: The effects of chart size and layering on the graphical perception of time series visualizations. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 1303–1312, 2009. doi: 10.1145/1518701.1518897

[25] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977. doi: 10.1080/03610927708827533

[26] W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):927–934, 2010. doi: 10.1109/TVCG.2010.162

[27] M. Kay and J. Heer. Beyond weber's law: A second look at ranking visualizations of correlation. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):469–478, 2016. doi: 10.1109/TVCG.2015.2467671

[28] M. Kerber, D. Morozov, and A. Nigmetov. Geometry helps to compare persistence diagrams. *Journal of Experimental Algorithmics (JEA)*, 22:1–4, 2017. doi: 10.1145/3064175

[29] S. K. Kopparapu and M. Satish. Identifying optimal gaussian filter for gaussian noise removal. In *Conference on Computer Vision, Pattern Recognition, Image Processing, and Graphics*, pp. 126–129, 2011. doi: 10.1109/NCVPRIPG.2011.34

[30] S. Kosslyn. Understanding charts and graphs. *Applied Cognitive Psychology*, 3:185 – 225, 1989. doi: 10.1002/acp.2350030302

[31] National Centers for Environmental Information. Climate Data Online: Web Services Documentation. `https://www.ncdc.noaa.gov/cdo-web/webservices/v2`, Apr 2020.

[32] New Zealand Tourist Arrivals. `https://tradingeconomics.com/new-zealand/tourist-arrivals`, Apr 2020.

[33] M. R. Nourbakhsh and K. J. Ottenbacher. The statistical analysis of single-subject data: a comparative examination. *Physical Therapy*, 74(8):768–776, 1994. doi: 10.1093/ptj/74.8.768

[34] B. Ondov, N. Jardine, N. Elmqvist, and S. Franconeri. Face to face: Evaluating visual comparison. *IEEE Transactions on Visualization and Computer Graphics*, 2018. doi: 10.1109/TVCG.2018.2864884

[35] A. V. Pandey, K. Rall, M. L. Satterthwaite, O. Nov, and E. Bertini. How deceptive are deceptive visualizations? an empirical analysis of common distortion techniques. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 1469–1478, 2015. doi: 10.1145/2702123.2702608

[36] S. Perreault and P. Hébert. Median filtering in constant time. *IEEE Transactions on Image Processing*, 16(9):2389–2394, 2007. doi: 10.1109/TIP.2007.902329

[37] W. Playfair. *The commercial and political atlas: representing, by means of stained copper-plate charts, the progress of the commerce, revenues, expenditure and debts of england during the whole of the eighteenth century*. T. Burton, 1801.

[38] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, 1972. doi: 10.1016/S0146-664X(72)80017-0

[39] J. D. Rhodes and S. Alseyab. The generalized chebyshev low-pass prototype filter. *International Journal of Circuit Theory and Applications*, 8(2):113–125, 1980. doi: 10.1002/cta.4490080205

[40] P. Rosen, A. Suh, C. Salgado, and M. Hajij. TopoLines: Topological Smoothing for Line Charts. In *EuroVis 2020 - Short Papers*, 2020. doi: 10.2312/evs.20201053

[41] G. Ryan, A. Mosca, R. Chang, and E. Wu. At a glance: Pixel approximate entropy as a measure of line chart complexity. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):872–881, 2019. doi: 10.1109/TVCG.2018.2865264

[42] T. Saito, H. N. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda. Two-tone pseudo coloring: Compact visualization for one-dimensional data. In *IEEE Symposium on Information Visualization (InfoVis)*, pp. 173–180, 2005. doi: 10.1109/INFVIS.2005.1532144

[43] B. Saket, A. Endert, and A. Demiralp. Task-based effectiveness of basic visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(7):2505–2512, 2019. doi: 10.1109/TVCG.2018.2829750

[44] A. Savitzky and M. J. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964. doi: 10.1021/ac60214a047

[45] C. E. Shannon. Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers*, 37(1):10–21, 1949. doi: 10.1109/

JRPROC.1949.232969

[46] 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

[47] 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

[48] 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

[49] 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

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

[51] U.S. Bureau of Labor Statistics. `https://www.bls.gov/`, Apr 2020.

[52] 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

[53] 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

[54] Yahoo Finance. `http://finance.yahoo.com/`, Apr 2020.

[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