J Vis (2019) 22:79–93
https://doi.org/10.1007/s12650-018-0518-y

REGULAR PAPER

Tan Tang · Kaijuan Yuan · Junxiu Tang · Yingcai Wu

Toward the better modeling and visualization of uncertainty for streaming data

Received: 21 July 2018 / Accepted: 7 September 2018 / Published online: 6 October 2018

Abstract Streaming data can be found in many different scenarios, in which data are generated and arriving continuously. Sampling approaches have been proven as an effective means to cope with the sheer volume of the streaming data. However, sampling methods also introduce uncertainty, which can affect the reliability of subsequent analysis and visualization. In this paper, we propose a novel model called PDm and visualization named uncertainty tree to present uncertainty that arises from sampling streaming data. PDm is first introduced to characterize uncertainty of streaming data, and an optimization method is then proposed to minimize uncertainty. Uncertainty tree is further developed to enhance data understanding by visualizing uncertainty and revealing temporal patterns of streaming data. Lastly, a quantitative evaluation and real-world examples have been conducted to demonstrate the effectiveness and efficacy of the proposed techniques.

Keywords Uncertainty visualization · Streaming data · Optimization · Time-series data

1 Introduction

Advances in computing infrastructure have brought a variety of streaming data, which are generated continuously. Effectively monitoring streaming data and quickly analyzing the spotted patterns become increasingly important for various applications, such as social stream analytics (Liu et al. 2016b) and real-time monitoring in smart factories (Xu et al. 2017). However, the sheer volume and high update frequency do not allow for storing, processing, and analyzing the comprehensive streaming data (Crouser et al. 2017). Thus, sampling techniques are widely and successfully adopted in numerous applications to cope with streaming data (Vitter 1985; Efraimidis and Spirakis 2006). Nevertheless, sampling streaming data inevitably produce uncertainty which can significantly affect the subsequent processing, analysis, and visualization (Wu et al. 2012). Without the proper assessment and visualization of uncertainty, unreliable or even erroneous conclusions can be drawn, resulting in undesired consequences (Thomson et al. 2005).

Let us consider an example to understand this problem. Suppose a marketing analyst wants to track and analyze the popularity of two commercial electronic products, A and B, according to people's comments on Twitter (see Fig. 1). Direct access to Twitter Firehose allows him/her to continuously receive a huge volume of tweets in real time. Given the extreme data size but limited computing resources (e.g., laptop), she/he has to resort to sampling techniques (Vitter 1985; Efraimidis and Spirakis 2006) to track and analyze the sampled, instead of the complete data. Suppose 60,000 tweets discuss product A, while only 40,000 tweets comment on product B in the complete dataset, which means product A is hotter than B on Twitter. But

T. Tang · K. Yuan · J. Tang · Y. Wu (✉)
State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China
E-mail: ycwu@zju.edu.cn

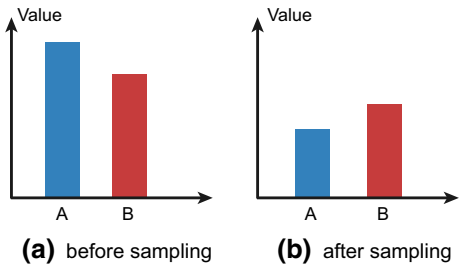

Fig. 1 Uncertainty arising from sampling streaming data can affect subsequent analysis and visualization. The opposite conclusions may be generated from **a** the complete dataset and **b** the sampled dataset

there are only 24,000 tweets concerning with product A, while 26,000 tweets discuss product B in the sampled dataset, as shown in Fig. 1. Without considering uncertainty of sampling the tweet stream, the analyst may draw an erroneous conclusion that product B is hotter than product A. Thus, how to quantify, minimize and make users aware of uncertainty is a key issue to the application of streaming data.

To address this problem, we have to overcome two obstacles. The first challenge is how to quantify and minimize uncertainty of sampled data. Various models have been introduced to characterize or quantify uncertainty arising in different disciplines (Gosink et al. 2013; Mirzargar et al. 2014). Although these methods can be applied to temporal data simultaneously, they cannot reduce uncertainty or relieve the impact on the subsequent analysis in the meantime. Thus, analysts have to employ extra methods to minimize uncertainty that arises from sampling streaming data, which increases computational burden. Limited by computational resources, it is necessary for researchers to develop a new model that can both quantify and minimize uncertainty of sampled data efficiently.

The second challenge is how to make users aware of uncertainty and facilitate reliable analysis. The most intuitive way is to develop uncertainty glyphs or encode uncertainty through visual channels, which can assist analysts in maintaining a high-level awareness of uncertainty. In recent years, researchers have introduced two visual metaphors, namely river flow (Liu et al. 2016b) and sedimentation (Huron et al. 2013), to visualize streaming data. These metaphors success in revealing streaming patterns and smoothing transitions between incoming and aging data. But they ignore uncertainty and cannot assess its impact on the subsequent analysis. Simply combining uncertainty glyphs or channels into existing visualizations may increase visual complexity and even produce heavy visual clutter. Thus, a novel visualization that can present both uncertainty and streaming patterns is required, which is also a non-trivial challenge.

This work focuses on the modeling and visualization of uncertainty that arises from sampling streaming data. Specifically, we have proposed a novel model to quantify and minimize uncertainty in the meantime. We incrementally extract sampled data from data streams and aggregate samples to generate the samples distribution. We define uncertainty of sampled data as the dispersion among the new and aging samples distributions (ISO 2008). This definition can be also regarded as a temporal generalization of variance, which is flexible and can be applied in different disciplines. Considering the influence of uncertainty, we cannot fully believe the generated samples distribution. Thus, we employ a linear model to estimate the precise samples distribution. The linear model is easy to understand and can be simultaneously integrated into the uncertainty model. We then acquire the estimated distribution of new samples using an optimization method (Wan et al. 2016) which aims to minimize the global uncertainty. Inspired by Bayesian surprise model (Correll and Heer 2016), we quantify the trustworthiness of aggregated samples and develop a novel visualization called *uncertainty tree* to demonstrate the individual-level uncertainty. In the end, we evaluate the model and visualization through a quantitative analysis and real-world examples.

Our contributions are summarized as follows:

- A novel model to characterize uncertainty that arises from sampling streaming data and an optimization method to alleviate the impact of uncertainty.
- An uncertainty-aware visualization to help analysts facilitate trustworthy analysis and exploration of streaming data.
- A quantitative evaluation and real-world examples to demonstrate the usage and validate the effectiveness of proposed techniques.

2 Related work

2.1 Uncertainty quantification

Uncertainty can arise from any stage in a data processing pipeline including data acquisition, transformation and visualization (Pang et al. 1997). Thus, how to model, quantify and visualize uncertainty information becomes an increasingly important topic across various domains (Skeels et al. 2008; Potter et al. 2012). Pang et al. (1997) presented a comprehensive survey of techniques including adding glyphs, adding geometry, modifying geometry, modifying attributes, animation and psycho-visual approaches that visually represent scientific data together with uncertainty. Based on frameworks for scientific visualization, Thomson et al. (2005) presented a general topology to describe uncertainty related to intelligence analysis. Zuk and Carpendale (2007) extended topology to include the uncertainty of reasoning. MacEachren (1992) introduced a conceptual model that related to geographic visualizations and addressed the difference between data quantify and uncertainty. Furthermore, they outlined general principles from three perceptual and cognitive theories (Bertin, Tufte and Ware) to analyze different uncertainty visualizations (Zuk and Carpendale 2006). Recently, Potter et al. (2012) identified frequently occurring types of uncertainty and connected them with common visual representations. We follow these well-established theories to analyze the sampling of streaming data and characterize uncertainty through probability distribution and uncertainty score.

Researchers have proposed various models and methods (Gosink et al. 2013; Mirzargar et al. 2014; Chen et al. 2015; Wu et al. 2010; Liu et al. 2016a; Cao et al. 2016) to alleviate the influences of uncertainty and improve the trustworthiness of analysis. These works can be classified into two groups. One group demonstrates uncertainty with scientific data, particularly ensemble data. Gosink et al. (2013) proposed a Bayesian model to characterize and visualize predictive uncertainty in numerical ensembles, while Chen et al. (2015) moved a step forward. They introduced an uncertainty-aware framework to visualize and explore multidimensional ensembles. The other group demonstrates uncertainty behind abstract information. For example, Wu et al. (2010) developed a circular wheel visualization to convey the uncertainty in the user feedback analysis. Liu et al. (2016a) introduced a variance/mean ratio(VMR) method to model uncertainty in the microblog retrieval and proposed a graph layout integrated with a circular box plot. Instead of employing the graph layout, Cao et al. (2016) proposed a triangle mesh to present the relationships between uncertain data and their probabilistic labels. However, these models cannot be employed directly to process streaming data due to the extremely large volume and fast updating rate. Hence, we introduce a novel model which can not only decrease the reducible uncertainty but also reveal the uneliminated uncertainty behind streaming data.

2.2 Uncertainty visualization

In addition to these well-established models, many intuitive visualizations were developed by researchers, such as density plots (Feng et al. 2010), summary plot (Potter et al. 2010) and curved box plot (Mirzargar et al. 2014). Feng et al. (2010) introduced a blurring method based on density plots to visualize uncertain data in scatterplots and parallel coordinates. And Potter et al. (2010) proposed a summary box plot to communicate descriptive statistics of uncertainty information. After that, Mirzargar et al. (2014) further enhanced the box plot to visualize main features of curve ensembles in simulations. Uncertain graph and network also attracted considerable attention from many researchers. Whitaker et al. (2013) introduce a multi-level visualization based on node-link diagrams to support the exploration of fuzzy overlapping communities in networks. Similarly, Schulz et al. (2017) investigated the probabilistic information of sampled graphs and utilized splatting, boundary shapes and bundled edges to enhance node-link diagrams. Park et al. (2016) introduced a visualization-aware sampling method to reduce the data size and guarantee the visualization quality. Kim et al. (2015) proposed a rapid sampling method to preserve the visual properties of visualizations, such as ordering. However, their methods focus on visualizing data directly instead of integrating uncertainty into the visualizations. Inspired by the successful visual idioms, we introduce a novel uncertainty visualization called *uncertainty tree* to reveal both uncertainty and data patterns.

2.3 Temporal visualization

Proper visualization techniques (Pak et al. 2003; Huron et al. 2013; Tanahashi et al. 2015) can facilitate analysis and exploration of streaming data, which is a continuously updated, unbounded data sequence. Wong et al. Pak et al. (2003) combined an adaptive visualization technique based on data stratification and an incremental visualization technique based on data fusion to visualize transient data streams. Inspired by physical sedimentation processes, Huron et al. (2013) proposed a novel visual metaphor called visual sedimentation to depict the incoming data as falling objects. Tanahashi et al. (2015) applied storylines to visualize streaming data with an efficient framework which consists of data management, layout construction and refinement. In recent years, text streams have attracted considerable attention. Various models and visualizations have been proposed to help people visually analyze topics evolution patterns (Cui et al. 2014; Liu et al. 2016b). Cui et al. (2014) introduced an incremental evolutionary tree cut algorithm to represent each tree with a tree cut according to users' interests, which supports a progressive exploration and analysis of hierarchical topics in large text corpora. However, the relationship between two consecutive topics remains unclear in the visualization. Based on a sedimentation metaphor, Liu et al. (2016b) aligned existing topics with new representative topics and developed TopicStream which supports exploration of hierarchical topic evolution. In this work, we have implemented a prototype to reveal both the temporal pattern and uncertainty of streaming data and integrated a set of interactions to facilitate level-of-detail exploration.

3 Problem and approach overview

This work is trying to mitigate the influence of uncertainty that arises from sampling streaming data. Specifically, our approach is established on the multivariate data, such as tweets stream, Internet flow and urban trajectories. Given N samples extracted at time span t, we want to estimate the samples distribution $\hat{f}_t$ and calculate the overall uncertainty U_t. The samples distribution can assist analysts in gaining valuable insights toward streaming data. For example, the geolocation is one attribute of tweets stream. We estimate the region distribution of sampled tweets so that we can understand which city has more or less tweets activities. Then the Twitter cooperation can reduce advertisements in cities which have more active users and put more on other places. Instead of showing numerical results of the model, we have also developed a novel visualization called *uncertainty tree* to demonstrate uncertainty and facilitate level-of-detail exploration of streaming data (Fig. 2).

Our approaches have the following steps:

Step 1 Sampling and aggregating streaming data. Direct processing and visualizing streaming data may come with two obstacles. One is that streaming data update continuously and appear at the unpredictable time. The other is that the sheer volume of streaming data increases the visual complexity heavily. Thus, we employ a moving time window to sample streaming data. At time span t, we aggregate samples s_i and the aggregate data are denoted as $a_j(t) = \{s_1, s_2, \ldots, s_n\}$.

Step 2 Estimating the samples distribution and reducing uncertainty. Given the aggregate data a_j, we can generate the initial samples distribution f_t which may not be accurate enough due to uncertainty. Thus, we employ a linear model to estimate the precise samples distribution $\hat{f}_t$, which closes to the distribution of complete data. Moreover, we define uncertainty U_t based on a temporal generalization of variance and

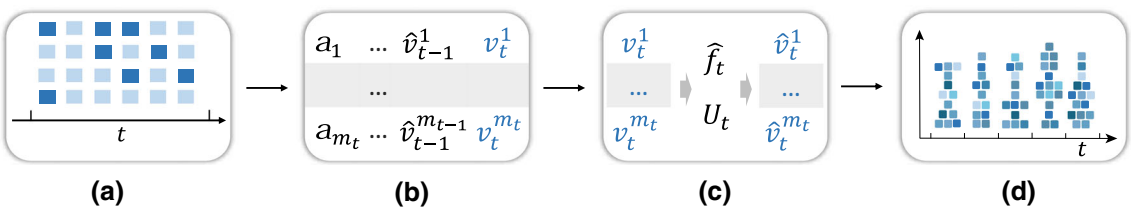

Fig. 2 Approach overview: **a** collecting and sampling streaming data at time span t. Each node represents one data item, and darker nodes represent samples. **b** Aggregating samples into a_j and generating the samples distribution $f_t(a_j), j = 1, \ldots, m_t$. **c** Estimating the precise distribution $\hat{f}_t$ and reducing uncertainty U_t. **d** Visualizing uncertainty and supporting level-of-detail exploration. Each node represents aggregated samples to reduce visual complexity

Table 1 Mathematical notations

Notation	Definition
t	The tth time span
s_i	The ith sample of streaming data
a_j	The jth aggregate data
$v_j(t)$	The number of samples in a_j
$\hat{v}_j(t)$	The estimated number of samples in a_j
f_t	The initial samples distribution at time span t
$\hat{f}_t$	The estimated samples distribution at time span t
U_t	The overall uncertainty of streaming data
c_t	The reliability score at time span t
w_t	The reliability weight at time span t
N_t	The total number of samples at time span t
$T_t(a_j)$	The trustworthiness of the aggregate data a_j

integrate $\hat{f}_t$ into the definition. We formulate the two tasks (estimation and reduction) into one optimization problem so that we can estimate $\hat{f}_t$ and reduce U_t at meantime.

Step 3 Visualizing uncertainty and facilitating level-of-detail exploration. We develop a new tree-shape visual idiom called uncertainty tree to demonstrate both individual- and global-level uncertainty. At the individual level, we assess the trustworthiness of the aggregate data a_j using Bayesian surprise model (Correll and Heer 2016) according to the estimated distribution $\hat{f}_t$. At the global level, we obtain the uncertainty U_t. In order to reveal temporal patterns of streaming data, uncertainty tree integrates a set of flexible interactions which help analysts facilitate level-of-detail exploration (Table 1).

4 Data sampling and aggregation

This section describes the process of sampling and aggregating streaming data. The streaming data are difficult to be processed and visualized due to three characteristics: high updating frequency, unpredictable incoming time and the sheer volume. Sampling, such as reservoir techniques (Vitter 1985; Efraimidis and Spirakis 2006), becomes a necessary pre-processing step for analysts who are limited by computational resources to cope with streaming data. The sampling process of streaming data is little bit different from sampling static datasets, as shown in Fig. 3. The sampling size of streaming data is fixed, while the sampling ratio is dynamic and decided by the total size of the complete dataset. For a newly arriving data item, the sampler decides to take it or not according to specific sampling methods. In our application, we choose the standard reservoir method (Vitter 1985) to sample streaming data. Notably, our model is established on the sampled dataset instead of specific sampling methods, which makes the model applicable to various sampling techniques.

We propose a moving time window to sample and aggregate streaming data at time span t. Samples can be aggregated by their attributes. For example, we can aggregate tweet samples according to the country tag. Each of the aggregate data a_j represents one country, and its value is defined as the number of tweets posted in that country at time span t. At the end of the time window, our model is invoked to process aggregate data and then the visualization updates. Besides, we can set the length of the time window so that we can explore

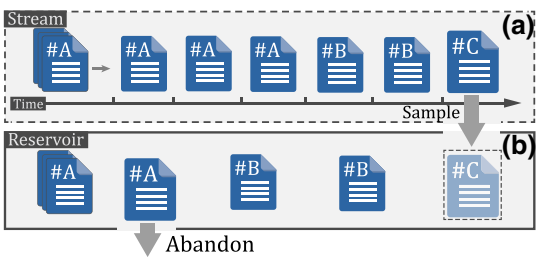

Fig. 3 The illustration of reservoir sampling (Vitter 1985): **a** the data stream. **b** The reservoir which stores samples. The newly arriving datum #C is sampled according to a random probability. Once acquiring a new sample, the sampler would abandon an existing one (#A) in the reservoir randomly. Thus, the volume of the reservoir is fixed, but the sampling ratio is dynamic

uncertainty and reveal temporal patterns of streaming data at different times granularity (e.g., minutes, hours, days).

5 Estimation and uncertainty model

This section describes how to estimate the samples distribution $\hat{f}_t$ and minimize the overall uncertainty U_t of streaming data.

5.1 Samples distribution

Given the aggregate data $a_j, j = 1, \ldots, m_t$, the initial samples distribution f_t can be acquired by

$$f_t(a_j) = \frac{v_j(t)}{\sum_{k=1}^{m_t} v_k(t)} \tag{1}$$

The samples distribution can help users understand the numerical relationships among samples, which is the foundation of various analytic tasks. Nevertheless, the initial distribution generated from sampled data cannot precisely describe the data distribution in the complete dataset due to uncertainty arising from sampling. We define the accuracy as the distance between the distributions extracted from sampled data and complete data. The inaccurate initial samples distribution cannot maintain the relative relationships among data items in the complete dataset, which has negative effects on the subsequent analysis and visualization. Back to the example in Sect. 1, we can calculate the tweets distribution in sampled dataset ($f(A) = 0.48, f(B) = 0.52$) and complete data ($f(A) = 0.6, f(B) = 0.4$). Obviously, the initial samples distribution describes the wrong numerical relationships among products A and B. Thus, we estimate the samples distribution by applying a weighting scheme (Wan et al. 2016) which combines historical and current information extracted from aging and incoming data, respectively.

$$\hat{f}_t = \sum_{i=0}^{m-1} w_{t-i} f_{t-i} \tag{2}$$

where w_{t-i} is the reliability weight of the initial samples distribution f_{t-i} and subject to $\sum_{i=0}^{m-1} w_{t-i} = 1$. m is the number of aging time spans. Equation 2 can be regarded as a linear regression model which is widely used in machine learning and statistics (Freedman et al. 2007). Typically, the parameters of a linear regression model can be obtained using the least-square method or the maximum likelihood method which minimizes the errors between estimated and labeled data. However, the labels $f_1, \ldots, f_t$ in our application are generated from sampled data directly and may not be accurate due to uncertainty. Thus, we define uncertainty based on Eq. 2 and estimate the samples distribution by formulating an optimization problem to minimize uncertainty.

5.2 Model

5.2.1 Uncertainty definition

Uncertainty refers to the phenomenon that people cannot precisely describe objects or states, which can be measured by various methods in different disciplines. We follow the formal definition of uncertainty, which can be found in the ISO guide (ISO 2008), to define uncertainty of sampled streaming data.

$$U_t = \frac{1}{m} \sum_{i=0}^{m-1} \| \bar{f}_t - f_{t-i} \|^2 \tag{3}$$

$$\bar{f}_t = \frac{1}{m} \sum_{i=0}^{m-1} f_{t-i} \tag{4}$$

Equations 3 and 4 can be regarded as a temporal generalization of variance which is widely used in statistics to calculate the dispersion of random phenomenon. Thus, our uncertainty definition can be also applied in other applications, such as the time-series ensemble. Nevertheless, Eqs. 3 and 4 assume that the reliability

of the information at a different time spans is the same and equal to $\frac{1}{m}$, which is not rationale according to Wan et al. (2016). We revise the uncertainty definition as:

$$U_t = \sum_{i=0}^{m-1} c_{t-i} \|\hat{f}_t - f_{t-i}\|^2, c_{t-i} > 0 \tag{5}$$

where c_t is the reliability score of the samples distribution f_t. In Eq. 5, $\overline{f}_t$ is replaced by $\hat{f}_t$ because the reliability scores at different times may not be equal. We can also obtain the relationships between the reliability score c_t and reliability weight w_t by combining Eqs. 2 and 5.

$$w_t = \frac{c_t}{\sum_{i=0}^{m-1} c_{t-i}} \tag{6}$$

Equation 5 would collapse to Eq. 3 when reliability scores are the same and equal to $\frac{1}{m}$. Thus, we conclude that Eq. 5 is a rational generalization of Eq. 3 by considering the reliability of the information at different times span may not be equal.

5.2.2 *Uncertainty minimization*

In this section, we cast an optimization problem to estimate the samples distribution by minimizing uncertainty and demonstrate how to solve it.

$$\min_{c_t, c_{t-1}, \ldots, c_{t-m+1} > 0} U_t = \sum_{i=0}^{m-1} c_{t-i} \|\hat{f}_t - f_{t-i}\|^2 \tag{7}$$

This problem can be efficiently solved by a two-stage iterative optimization model named KDEm (Wan et al. 2016).

Algorithm 1: A Two-stage Iterative Model: KDEm

Input: $f_{t-i}, i = 0, \ldots, m-1$
Output: Samples distribution $\hat{f}_t$ and Uncertainty U_t
Initialize $c_t^0 = \ldots = c_{t-i}^0 = \ldots = c_{t-m+1}^0 = \frac{1}{m}$;
repeat
  (a) Update $\hat{f}_t$ by $\hat{f}_t^{(k+1)} = \sum_{i=0}^{m-1} w_{t-i}^{(k)} f_{t-i}$
  where $w_{t-i}^{(k)} = \frac{c_{t-i}^{(k)}}{\sum_{j=0}^{m-1} c_{t-j}^{(k)}}, i = 0, \ldots, m-1$;
  (b) Update $c_{t-i}^{(k)}$ by
  $c_{t-i}^{(k+1)} = -log(\frac{\|\hat{f}_t^{(k+1)} - f_{t-i}\|}{\sum_{j=0}^{m-1} \|\hat{f}_t^{(k+1)} - f_{t-j}\|}), i = 0, \ldots, m-1$;
until U_t *cannot be reduced*;

Algorithm 1 demonstrates the pseudocode. The algorithm first initializes all reliability scores $\{c_{t-i}\}_{i=0}^{m-1}$ by assigning $\frac{1}{m}$. In the first stage, the algorithm updates the reliability weights $\{w_{t-i}\}_{i=0}^{m-1}$ using Eq. 6 and obtains $\hat{f}_t$ using Eq. 2. In the second stage, once the new $\hat{f}_t$ is obtained, the algorithm updates reliability scores $\{c_{t-i}\}_{i=0}^{m-1}$ by

$$c_{t-i}^{(k+1)} = -\log\left(\frac{\|\hat{f}_t^{(k+1)} - f_{t-i}\|}{\sum_{j=0}^{m-1} \|\hat{f}_t^{(k+1)} - f_{t-j}\|}\right) \tag{8}$$

where $i = 0, \ldots, m-1$ and k is the iteration times. If U_t, which is derived from Eq. 5, changes considerably, then steps (a) and (b) of Algorithm 1 will be repeated.

KDEm makes sure that $\hat{f}_t$ converges to the accurate samples distribution to the largest extent. If an initial samples distribution f_{t-j} is close to the estimated $\hat{f}_t$, then the associated reliability score c_{t-j} will update to a higher value according to Eq. 8. If the updated reliability scores c_{t-j} become more trustworthy, which indicates that f_{t-j} is more similar to the samples distribution of the complete data, then the obtained $\hat{f}_t$ will be

more accurate. Thus, trustworthy reliability scores $c_t, \ldots, c_{t-m+1}$ and accurate $\hat{f}_t$ can mutually enhance each other by iterating step (a) and (b) until uncertainty U_t cannot be reduced.

In the literature (Wan et al. 2016), the distance norm $\|\cdot\|$ between two samples distributions $\hat{f}_t$ and f_t is defined as:

$$\|\hat{f} - f\|^2 = K_h(\hat{f},\hat{f}) - 2K_h(f,\hat{f}) + K_h(f,f) \tag{9}$$

where $K_h(\cdot,f)$ is a kernel function (e.g., Gaussian kernel), and h is a bandwidth. Equation 9 is developed to measure the distance between distributions of numerical variables and cannot be applied in our applications. Thus, we employ the KL divergence to measure the distance between $\hat{f}_t$ and f_t, both of which are distributions of discrete variables $a_j, j = 1, \ldots, m_t$.

$$\|\hat{f} - f\| = \sum f(i) \log \frac{f(i)}{\hat{f}(i)} \tag{10}$$

In statistic, KL divergence is widely used to measure the similarity between two distributions. If f_t is similar to $\hat{f}_t$, then $\|\hat{f}_t - f_t\|$ will be decreased according to Eq. 10 and the associated reliability score c_t will be increased according to Eq. 8. Thus, KL divergence is also a rational measure to calculate the distance between two discrete distributions. We name the improved model as PDm, and the comparison between KDEm and PDm is illustrated by a quantitative evaluation in Sect. 7.

The obtained uncertainty U_t and the reliability score c_t provide two important signals that can detect major changes of the data stream in which analysts may be interested. If the data stream changes abruptly, then the initial samples distribution f_t will be quite different from previous distributions. We describe this phenomenon as diversity. According to algorithm 1, our model will turn down the reliability score c_t automatically. But the minimized uncertainty may be still a little higher than before. Thus, the higher uncertainty U_t and the lower reliability score c_t indicate that some major events may appear and change the data stream abruptly. In the next section, we visualize these valuable cues to make analysts be aware of uncertainty when exploring streaming data.

6 Uncertainty visualization

This section introduces a novel uncertainty visualization called *uncertainty tree* to facilitate exploration and analysis of streaming data and make users be aware of uncertainty. The visualization demonstrates uncertainty of sampled streaming data at two levels, namely individual and global levels. At the individual level, we propose the Bayesian surprise model to assess the trustworthiness of the aggregate data a_j and present the trustworthy level using color encodings. At the global level, we present the relative changes of the overall uncertainty U_t using diverging color encodings to reveal the overall evolution of uncertainty. Furthermore, we integrate uncertainty tree with a set of interactions to facilitate level-of-detail exploration and analysis.

6.1 Surprise node

The sheer volume of streaming data increases the heavy burden of the human cognitive system and makes it difficult for users to search a pattern visually. Therefore, we visualize the aggregate data $a_1, a_2, \ldots a_{m_t}$ to decrease the visual complexity. For each aggregate data a_j, we obtain the value $v_j(t)$ and distribution f_t by solving Eq. 7. As discussed above, the value $v_j(t)$ may not reflect the precise ratio of a_j in the complete dataset due to uncertainty, which has negative effects in the subsequent analysis. Thus, we employ Bayesian surprise model (Correll and Heer 2016) to assess the trustworthy level of aggregate data $a_j, j = 1, \ldots m_t$.

Bayesian surprise model aims to measure the changes in beliefs by comparing the prior and posterior information, of which the bigger differences indicate higher-level untrustworthiness of data. In general, the prior information can be obtained from historical data, while the posterior information is usually collected through observation or estimation (Correll and Heer 2016). Thus, the value $v_j(t)$ is defined as the posterior information in our application and the estimated value $\hat{v}_j(t)$ can be regarded as the prior information since it is estimated from historical values $v_j(t-j), j = 1, 2, 3 \ldots$.

$$\hat{v}_j^i(t) = N_t * \hat{f}_t(a_j^i) \tag{11}$$

where N_t is the total number of samples at time span t, and $\hat{f}_t(a_j)$ is the estimated distribution of aggregate data $a_j, j = 1, \ldots, m_t$. The distribution $\hat{f}_t$ is obtained by iteratively updating the weighting scheme which balances information at different time spans until uncertainty is minimized (see Sect. 5). Thus, $\hat{f}_t$ relieves the impact of uncertainty to a large extent and can estimate the precise value $\hat{v}_j(t)$.

Correll et al. Correll and Heer (2016) have developed an efficient method based on Bayesian theory to calculate the surprise distribution of temporal events data. We follow this definition and define the trustworthy level of each aggregate data a_j as

$$T_t(a_j) = \sum_{k=0}^{m-1} w_{t-k} * (1 - |E_j - O_j|) * \log(1 - |E_j - O_j|) \tag{12}$$

where E_j and O_j are distributions of the estimated value $\hat{v}_j(t)$ and observed value $v_j(t)$ at the temporal side, respectively. Bayesian surprise captures the notion that the changes between prior and posterior information characterize the trustworthiness of aggregate data where the larger difference indicates the lower confidence level.

As shown in Fig. 4a, each of the aggregate data is visualized using a rectangle whose color encodes the trustworthy levels. The more trustworthy the aggregate data is, the more salient the associated node appears to be, and vice verse. Thus, we name our design as surprise node which provides a visual cue for people to be aware of the untrustworthy data. Moreover, we employ two kinds of contours, namely dashed contour and solid contour, to represent the dynamic changes of values at consecutive time spans. A rectangle with a dashed contour or solid contour represents the item that has a larger or lower value than before, respectively. Thus, the contour encoding helps users understand how the value of aggregate data evolves.

6.2 Uncertainty tree

We design a new visual metaphor, *uncertainty tree*, to lay out surprise nodes and visualize the overall uncertainty U_t. Uncertainty tree is the core component of the visualization system, providing an intuitive visual summary of uncertainty information, which help analysts develop a growing understanding of how and where uncertainty was introduced. Uncertainty tree can also help users control the confidence level of datasets through some flexible interactions and facilitate more trustworthy exploration and analysis.

The uncertainty tree is stacked by surprise nodes at time span t, and the layout is oriented from bottom to top. For each aggregate data a_j, the associated surprise node is positioned according to its value $v_j(t)$. Specifically, surprise nodes who have larger values can occupy higher positions along the y-axis and those with the same values are placed at the same level, as shown in Fig. 4b. The order of surprise nodes along the x-axis within a time span has no actual meaning. We provide users two layout options to decide the overall shape of uncertainty tree, namely symmetric and one-sided (see Fig. 4c, d), both of which present the overall data distribution. However, stacking surprise nodes directly may lead to sparse layout, as shown in Fig. 4b. Thus, we bin y-axis to make uncertainty tree more scalable (see Fig. 4c). Furthermore, we place uncertainty trees horizontally to reveal temporal patterns and help analysts compare multiple trees easily.

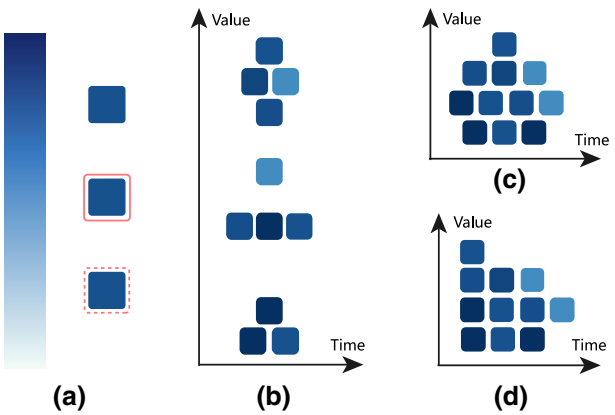

Fig. 4 Design considerations of uncertainty tree: **a** encoding the trustworthiness of the surprise node using sequential colors, **b** stacking surprise nodes according to their values, **c** binning y-axis to reduce visual sparsity, **d** providing two kinds of layouts

We next visualize the global-level uncertainty U_t which assists analysts to understand the overall variation of streaming data. As shown in Fig. 6, we encode the relative change of U_t using the background color of uncertainty tree. The diverging color legend is used since the relative change of U_t ranges from negative to positive. Compared to visualize the absolute value of U_t, the relative change provides a more intuitive visual cue for analysts to understand the evolution of the overall uncertainty.

6.3 Interaction

Besides general interactions (i.e., panning), uncertainty tree integrates a set of other interactions to support level-of-detail exploration and analysis, such as the following:

- *Hover* When users hover a node, a pop-up card displays the detail information about the node and the contour would be changed to reveal the difference between the current and previous values.
- *Highlight* When users click a node, the associated nodes at different time spans would be connected to reveal the temporal pattern. Moreover, the width of the link will be decreased/increased according to its value change.
- *Filter* The visualization provides visual cues for analysts to be aware of uncertainty. The filtering interaction helps users exclude untrustworthy data and facilitate reliable analysis.
- *Relayout* In order to assist users to understand the distribution of streaming data, we provide users two options to reshape uncertainty tree, namely symmetric and one-sided. An animation will be invoked to present a smooth transition when layout changes.

7 Quantitative evaluation

In this section, we conduct a quantitative evaluation to validate whether the improvement to KDEm is efficient by comparing PDm and KDEm, which is the baseline method. We test PDm and KDEm on artificial datasets and measure the accuracy of their results and the average implementation time. In this experiment, we first construct several datasets on the basis of two ground-truthed distributions, namely uniform distribution (UD) and geometric distribution (GD). We then test PDm and KDEm, both of which can estimate ground-truthed distributions from these artificial datasets, and employ *rooted-mean-square errors* (*RMSE*) to measure the accuracy of their results. We also record the time when PDm and KDEm are tested on different datasets. Furthermore, we repeat the experiment five times to eliminate the interference of other factors, such as CPU scheduling. Finally, we compare PDm and KDEm according to their average implementation time and RMSE to validate the efficiency of PDm.

Datasets are generated as follows. We first generate n artificial distributions whose dimension is m for each ground-truthed distribution (UD or GD). Generating the artificial distribution is interfered by a random error who follows Gaussian distribution $N(0, \sigma^2)$. Therefore, errors exist between artificial distributions and ground-truthed distributions. We test both PDm and KDEm on two ground-truthed distributions for $\sigma = 0.1, m = 100$ and $n = 100, 1000, 10{,}000$. In total, two small-size ($n = 100$), medium-size ($n = 1000$) and large-size ($n = 10{,}000$) artificial datasets are generated according to ground-truthed distributions.

Measures For each dataset, we have one ground-truthed distribution (UD or GD) and one estimated distribution, which is obtained using PDm or KDEm. We can use RMSE to measure the accuracy of results.

$$\mathrm{RMSE} = \sqrt{\frac{1}{r}\sum_{i=1}^{r} \|f_e - f_g\|^2} \tag{13}$$

where r is the repetition times, and f_e and f_g are the estimated and ground-truthed distribution, respectively. We use the average implementation time to measure the computational performance of algorithms, which is defined as

$$\bar{t} = \frac{1}{r}\sum_{i=1}^{r} t_i \tag{14}$$

where t_i is the implementation time of the ith experiment. Smaller RMSE and average implementation time indicate better performance.

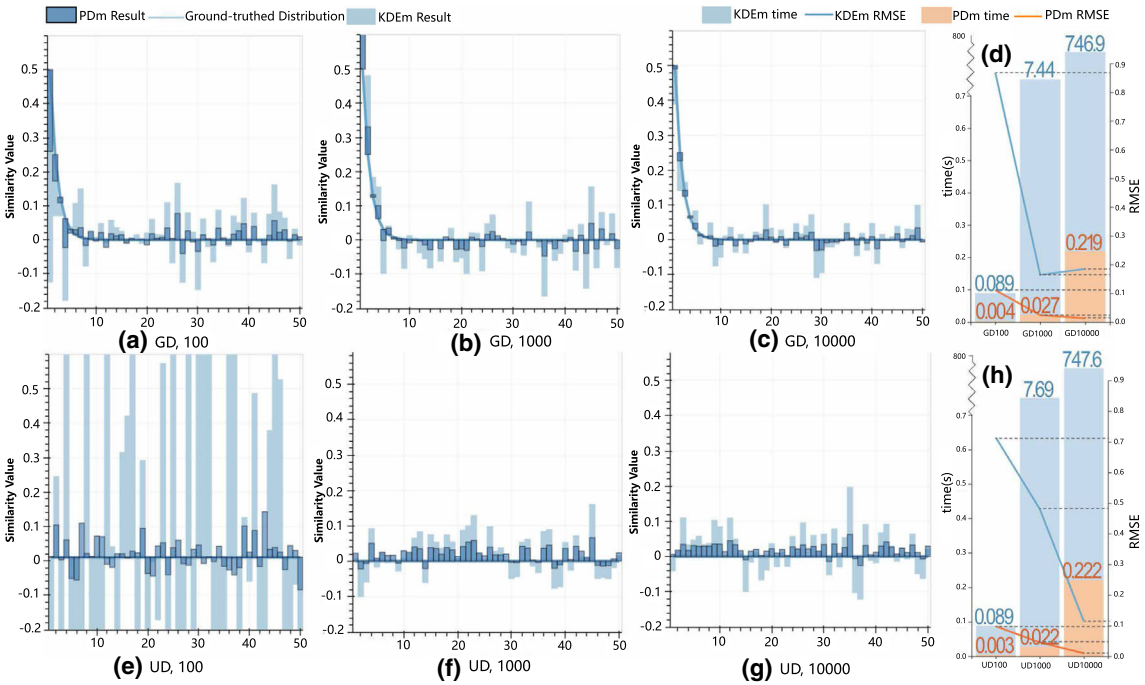

Fig. 5 Model comparison. **a**–**c** result on the uniform distribution, whereas **e**–**g** result on the geometric distribution. The blue line represents the ground-truthed distribution. The darker blue bar represents the estimated distribution of PDm, and the lighter blue bar represents that of KDEm. **d**, **h** summarize results on two ground-truthed distributions. The *x*-axis represents the size of the datasets. The bar chart represents the average implementation time that is measured by the left *y*-axis. The line chart represents the RMSE that is measured by the right *y*-axis. The blue color presents results of KDEm, and the orange color presents results of PDm

Results are shown in Fig. 5. Figure 5a–c presents the ground-truthed distribution, UD and estimated distributions of PDm and KDEm. We notice that the estimated distribution of PDm is much closer to the ground-truthed distribution compared to that of KDEm among the three datasets, which have different data size. We notice the similar phenomenon through observing Fig. 5e–g, which present the ground-truthed distribution, GD and results of both PDm and KDEm. Thus, we conclude that PDm has better performance than KDEm from the aspect of accuracy. Figure 5d, h validates this conclusion because the RMSE values of KDEm (blue line) are higher than those of PDm (orange line). Furthermore, we also notice that estimated distributions become more accurate with the increase in data size. This is because increasing knowledge about the truth can decrease uncertainty (Potter et al. 2012).

Figure 5d, h also presents the average implementation time of both PDm (orange bar) and KDEm (blue bar), which grows considerably with the increase of data size. Nevertheless, a significant improvement is observed in the computational performance of PDm, which can be a hundred times, even a thousand times faster than KDEm when data size is relatively large. Therefore, PDm can achieve better performance than KDEm in both the implementation time and accuracy of results. Finally, we conclude that the improvement to KDEm is efficient according to the results of this experiment.

8 Usage scenario

This section introduces real-world examples to demonstrate the usage and efficacy of the visualization. Social media has become an inevitable role in modern life. People tend to present life moments, comment on public issues and follow the opinion leader on social media, such as Twitter (Wu et al. 2016). Therefore, we pick this application as our examples. The data were collected through Twitter API.[1] We collected tweets posted on January 1, 2016, and each tweet contains 31 attributes including ID, timestamp, hashtag and content. The entire dataset contains around 8 million tweets and occupies 3 GB of disk space. We then

[1] https://developer.twitter.com/.

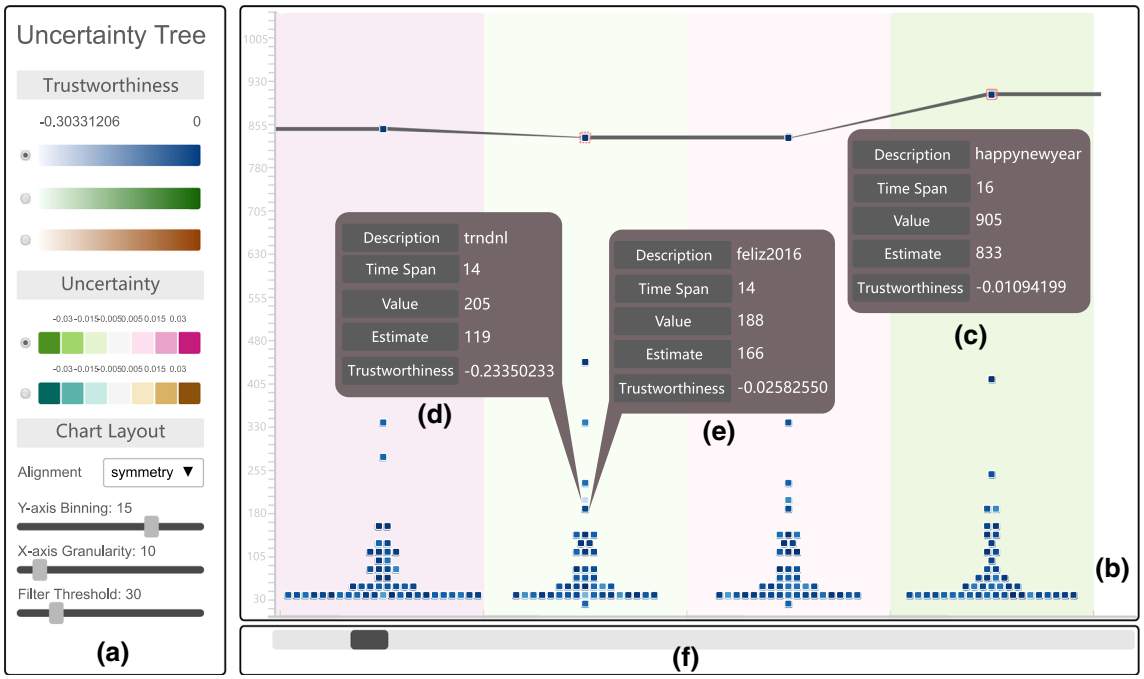

Fig. 6 Our system includes (**a**) a panel which controls parameters of the model and visualization, (**b**) uncertainty tree which integrates level-of-detail information (**c**), (**d**) and (**e**), as well as (**f**) a timeline slider which tracks the evolution of the streaming data

employed the reservoir technique (Vitter 1985) to sample tweets stream to simulate the sampling of streaming data in the real world. The volume of the reservoir is fixed at 1.5 million, and the sampling rate is approximately a quarter.

The data are visualized in Fig. 6. Each surprise node represents one topic discussed on Twitter, and the value of node equals to the counting number of tweets concerning the topic. We set the time granularity to 10 min and bin *y*-axis by 15 to decrease the visual complexity. We also filter topics whose value is less than 30 so that we can focus on well-received topics on Twitter. After clicking the top node, analysts can immediately see that the hottest topic is *happynewyear*. Meanwhile, analysts can easily understand the evolution of the hottest topic with the enhancement of links and contours of surprise nodes. Moreover, analysts can observe that the hottest topic is filled with darker color which indicates its trustworthy level is high. The visualization is validated by the truth that most people celebrate the new year on January 1, 2016, so that *happynewyear* becomes the hottest topic and the conclusion is reliable.

Analysts can also observe one "different" topic *trndnl* in Fig. 6d whose color is clearly lighter than neighboring nodes. This pattern indicates that the estimated value of the node is much less than the sampled value, which makes the trustworthy level of the node lower than that of others. Analysts would conclude that the topic *trndnl* is hotter than topic *feliz2016* according to their value difference without considering uncertainty. However, the conclusion is wrong according to the complete dataset which validates that uncertainty can affect subsequent analysis and visualization. Thus, analysts can easily discover untrustworthy data and be aware of uncertainty with uncertainty tree. However, uncertainty does not equal to errors (Freedman et al. 2007). The primary object of uncertainty tree is to make users be aware of uncertainty instead of erasing erroneous data. Thus, uncertainty tree provides the filtering interaction for users so that they can decide to remove data or not. Moreover, uncertainty tree provides estimated values to help analysts make decisions.

Besides, we can also visualize the tweets distribution among countries all over the world. The time granularity is set to 10 min, and the *y*-axis is binned by 50 to generate legible visualizations. Each surprise node represents a country, and the *y*-axis represents tweets number. According to Fig. 7c, analysts can observe that the Philippines has the most Twitter fans since it always occupies the top level. Moreover, the trustworthiness of surprise nodes indicates that this observation is reliable. However, there exist countries whose position in uncertainty tree is untrustworthy (see Fig. 7a, b). According to estimated values, the tweets posted in the State of Israel may be undersampled, while the tweets posted in Kazakhstan may be

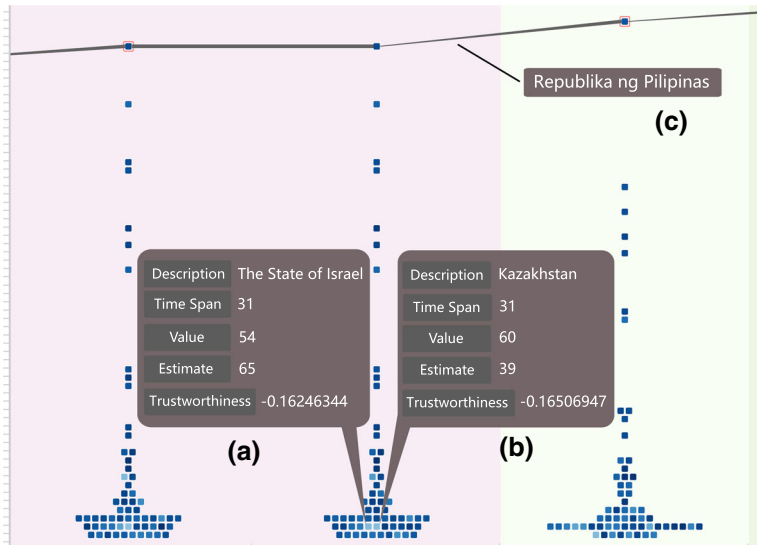

Fig. 7 Tweets distribution among countries: **a** the State of Israel. **b** Kazakhstan. **c** the Philippines

oversampled. These untrustworthy samples also increase the overall uncertainty at time span 31 where the background color of uncertainty tree becomes red.

9 User feedback

To understand the usability of uncertainty tree, we conduct semi-structured interviews with two visualization designers (P1 and P2). Both of them have more than two years of experiences in visualization design and research. The interviews begin with an introduction about uncertainty tree. The interviewees can freely explore our system and raise any questions. Then, we ask them to complete the following tasks:

Task 1 Locate the uncertain time span and find the untrustworthy data.

Task 2 Analyze the semantic information of the untrustworthy data.

We use the same dataset as mentioned above. The default time granularity is set to 10 min, and the *y*-axis is binned by 50. In task 1, both interviewees can promptly find the uncertain time window since its background is an apparent visual cue for users. They can also successfully locate the untrustworthy node whose color is clearly lighter than surrounding nodes. The pattern in Fig. 6 is also validated by the two participants. In task 2, both participants can acquire the detail information of the aggregate data using our interactions. However, P2 mentioned that "it is not easy to compare two neighboring nodes at the same level." The reason is that users have to click nodes twice to obtain the values. We plan to employ an adaptive layout method to solve this issue. The nodes at the same level can be re-layout according to their values in the *y*-axis when users invoke interactions. P1 also gave us valuable suggestions that "it would be useful to engage users using animations to update nodes."

10 Discussion

The analysis of streaming data is hindered by the large volume and high update frequency. To address these challenges, sampling techniques are widely employed to enable analysts to explore incomplete datasets. However, sampling inevitably introduces uncertainty. Direct visualization of sampled streaming data may lead to undesired results or even erroneous conclusions. Thus, we develop a novel model called PDm and new visualization named uncertainty tree to characterize uncertainty that arises from sampling streaming data. Our approaches can help analysts perform more trustworthy analysis and draw more reliable conclusions from streaming data.

Although PDm is established on streaming data, it can be extended to other temporal datasets, such as the time-series ensemble. On the one hand, the uncertainty quantification method in PDm follows the formal

definition of uncertainty in the ISO guide (ISO 2008), which is accepted in various disciplines. On the other hand, the uncertainty formula (see Eq. 5) is an extension to traditional variance formula (Freedman et al. 2007) whose mathematical form can be flexibly applied in different problems. Thus, PDm is a general model to quantify uncertainty on temporal datasets.

Uncertainty tree is a novel visualization method which can both reveal temporal patterns of streaming data and make people aware of uncertainty. Uncertainty tree visualizes the aggregated form of streaming data and uses binned y-axis to stack aggregate data which makes it scalable to the different size of the dataset. Two real-world examples indicate that uncertainty tree can be employed in different analysis situations. Thus, it can be integrated as an attached view into other applications to facilitate trustworthy exploring and analysis.

Although the evaluation demonstrates that PDm and uncertainty tree can be successfully applied in practical applications, there exist some limitations. Equation 5 indicates that the abrupt change of streaming data may also increase the global uncertainty U_t. PDm then reduces uncertainty and alleviates its impact to the newly arriving data. However, two possibilities can account for the abrupt changes of streaming data. One scenario is that uncertainty produces erroneous patterns, which analysts must ignore. The other scenario is that novel patterns appear in the newly arriving data, which is caused by the diversity of data. PDm cannot distinguish between the two different scenarios, namely uncertainty and diversity. Therefore, we plan to incorporate user knowledge to maintain the balance between diversity and uncertainty. For example, a slider widget can be used in the system which enables users to adjust the parameters of PDm and control the results.

11 Conclusion

In this work, we present a new model called PDm to assist analysts to perform more trustworthy exploration and analysis of streaming data. PDm first quantifies the global uncertainty of streaming data, and then an optimization method is proposed to minimize it. We then develop a novel visualization named uncertainty tree to enhance data understanding by presenting uncertainty at the different levels. The individual-level uncertainty is revealed by Bayesian surprise model (Correll and Heer 2016) and visualized by surprise nodes. We integrate uncertainty tree with a set of interactions to reveal temporal patterns of streaming data and facilitate level-of-detail exploration. In the end, we conduct a quantitative evaluation and demonstrate real-world examples to validate the effectiveness of our approaches.

Acknowledgements The work was supported by NSFC (61761136020, 61502416), NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (U1609217), Zhejiang Provincial Natural Science Foundation (LR18F020001) and the 100 Talents Program of Zhejiang University. This project was also partially funded by Microsoft Research Asia.

References

Cao N, Lin YR, Gotz D (2016) Untangle map: visual analysis of probabilistic multi-label data. IEEE Trans Vis Comput Graph 22(2):1149–1163

Chen H, Zhang S, Chen W, Mei H, Zhang J, Mercer A, Liang R, Qu H (2015) Uncertainty-aware multidimensional ensemble data visualization and exploration. IEEE Trans Vis Comput Graph 21(9):1072–1086

Correll M, Heer J (2016) Surprise! bayesian weighting for de-biasing thematic maps. IEEE Trans Vis Comput Graph 23(1):651–660

Crouser RJ, Franklin L, Endert A, Cook K (2017) Toward theoretical techniques for measuring the use of human effort in visual analytic systems. IEEE Trans Vis Comput Graph 23(1):121–130

Cui W, Liu S, Wu Z, Wei H (2014) How hierarchical topics evolve in large text corpora. IEEE Trans Vis Comput Graph 20(12):2281–2290

Efraimidis PS, Spirakis PG (2006) Weighted random sampling with a reservoir. Inf Process Lett 97(5):181–185

Feng D, Kwock L, Lee Y, Taylor R (2010) Matching visual saliency to confidence in plots of uncertain data. IEEE Trans Vis Comput Graph 16(6):980–989

Freedman D, Robert P, Purves R (2007) Chance Errors in Sampling. In: Statistics, 4th edn. pp 355–374

Gosink L, Bensema K, Pulsipher T, Obermaier H, Henry M, Childs H, Joy KI, Owhadi H, Scovel C, Sullivan T (2013) Characterizing and visualizing predictive uncertainty in numerical ensembles through bayesian model averaging. IEEE Trans Vis Comput Graph 19(12):2703–2712

Huron S, Vuillemot R, Fekete JD (2013) Visual sedimentation. IEEE Trans Vis Comput Graph 19(12):2446–2455

ISO (2008) Evaluation of measurement data—guide to the expression of uncertainty in measurement

Kim A, Blais E, Parameswaran A, Indyk P, Madden S, Rubinfeld R (2015) Rapid sampling for visualizations with ordering guarantees. Proc VLDB Endow 8(5):521–532
Liu M, Liu S, Zhu X, Liao Q, Wei F, Pan S (2016a) An uncertainty-aware approach for exploratory microblog retrieval. IEEE Trans Vis Comput Graph 22(1):250–259
Liu S, Yin J, Wang X, Cui W, Cao K, Pei J (2016b) Online visual analytics of text streams. IEEE Trans Vis Comput Graph 22(11):2451–2466
MacEachren AM (1992) Visualizing uncertain information. Cartogr Perspect 13:10–19
Mirzargar M, Whitaker RT, Kirby RM (2014) Curve boxplot: generalization of boxplot for ensembles of curves. IEEE Trans Vis Comput Graph 20(12):2654–2663
Pak CW, Foote H, Adams D, Cowley W, Thomas J (2003) Dynamic visualization of transient data streams. In: Proceedings of the IEEE symposium on information visualization, pp 97–104
Pang AT, Wittenbrink CM, Lodha SK (1997) Approaches to uncertainty visualization. Vis Comput 13(8):370–390
Park Y, Cafarella M, Mozafari B (2016) Visualization-aware sampling for very large databases. In: 2016 IEEE 32nd international conference on data engineering (ICDE). IEEE, pp 755–766
Potter K, Kniss J, Riesenfeld R, Johnson C (2010) Visualizing summary statistics and uncertainty. Comput Graph Forum 29(3):823–832
Potter K, Rosen P, Johnson CR (2012) From quantification to visualization: a taxonomy of uncertainty visualization approaches. IFIP Adv Inf Commun Technol 377:226–249
Schulz C, Nocaj A, Goertler J, Deussen O, Brandes U, Weiskopf D (2017) Probabilistic graph layout for uncertain network visualization. IEEE Trans Vis Comput Graph 23(1):531–540
Skeels M, Lee B, Smith G, Robertson G (2008) Revealing uncertainty for information visualization. In: Proceedings of the working conference on advanced visual interfaces, pp 376–379
Tanahashi Y, Hsueh CH, Ma KL (2015) An efficient framework for generating storyline visualizations from streaming data. IEEE Trans Vis Comput Graph 21(6):730–742
Thomson J, Hetzler E, MacEachren A, Gahegan M, Pavel M (2005) A typology for visualizing uncertainty. Proc SPIE Vis Data Anal 5669:146
Vitter JS (1985) Random sampling with a reservoir. ACM Trans Math Softw 11(1):37–57
Wan M, Chen X, Kaplan L, Han J, Gao J, Zhao B (2016) From truth discovery to trustworthy opinion discovery: an uncertainty-aware quantitative modeling approach. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp 1885–1894
Whitaker RT, Mirzargar M, Kirby RM (2013) Contour boxplots: a method for characterizing uncertainty in feature sets from simulation ensembles. IEEE Trans Vis Comput Graph 19(12):2713–2722
Wu Y, Wei F, Liu S, Au N, Cui W, Zhou H, Qu H (2010) OpinionSeer: interactive visualization of hotel customer feedback. IEEE Trans Vis Comput Graph 16(6):1109–1118
Wu Y, Yuan GX, Ma KL (2012) Visualizing flow of uncertainty through analytical processes. IEEE Trans Vis Comput Graph 18(12):2526–2535
Wu Y, Cao N, Gotz D, Tan YP, Keim DA (2016) A survey on visual analytics of social media data. IEEE Trans Multimed 18(11):2135–2148
Xu P, Mei H, Ren L, Chen W (2017) ViDX: visual diagnostics of assembly line performance in smart factories. IEEE Trans Vis Comput Graph 23(1):291–300
Zuk T, Carpendale S (2006) Theoretical analysis of uncertainty visualizations. In: Proceedings of the SPIE visualization and data analysis, vol 6060, p 606007-14
Zuk T, Carpendale S (2007) Visualization of uncertainty and reasoning. Proc Int Symp Smart Graph 4569:164–177