



Sampling for Scalable Visual Analytics

Bum Chul Kwon and Janu Verma

IBM T.J. Watson Research Center

Peter J. Haas

IBM Almaden Research Center

Çağatay Demiralp

IBM T.J. Watson Research Center

The ability to scale interactive visual analysis to massive datasets is becoming increasingly important. For example, almost one-quarter of the 459 respondents in a 2015 KDnuggets poll analyzed datasets with sizes ranging from 1 terabyte to more than 100 petabytes.¹ Sampling is the canonical method for quickly and flexibly inferring patterns in large data, because approximate answers based on samples are often as useful as exact answers, and sampling can also reduce the cognitive burden of visual clutter. Prior research in the database community has yielded valuable insights into sampling and visual analytics, but this work has focused primarily on aggregation queries and on systems issues related to accessing the sample data.

Here, we make a case for sampling as an essential tool for scalable interactive visual analysis. We first outline prior work by the database community on sampling for visualizing aggregation queries and then consider how these results might be improved and extended to a broader setting. In particular, we discuss issues important to sampling-based visual analytics and delineate three future research directions: understanding the interplay between sampling and perception, assessing and visually representing sampling-induced uncertainty, and giving nonexpert users interactive control over the sampling process. More generally, we need to better understand how users interact with sampling to enable its wider adoption for scalable visual analytics.

Why Is Sampling Needed?

Visual analytics is a powerful tool for exploring and understanding data, as it augments human cognition by leveraging visual perception and facilitates interactive, iterative analysis workflow, which is essential for data experimentation. The visualization of large datasets, however, poses several challenges. First, displaying a large number of data items can create visual clutter, challenging perception and, hence, visual analysis. For example, as Danyel Fisher pointed out, a scatterplot based on numerous data points will typically appear as a dark mass that obscures any fine data structure.² Equally challenging is the problem of seamlessly exploring large datasets at interactive rates; such exploration typically involves coordinating multiple visualizations using brushing, linking, panning, and zooming, which compounds performance challenges. Since the late 1990s, researchers have recognized that the increasing volumes of data make it difficult to achieve interactivity by merely applying raw processing power.³

The two basic approaches to scalable interactivity are precomputation and sampling.⁴ *Precomputation* refers to processing data into formats such as prespecified tiles or cubes to interactively answer queries via zooming, panning, brushing, and so on. This approach has been prevalent in both the visualization and database communities, from which most of the current techniques originate. Pure precomputation is not always desirable or feasible, however. Clearly, precomputation alone does not

address visual clutter. Moreover, precomputation is inflexible because it restricts a user's ability to run ad hoc queries for interactively generating and testing hypotheses. For massive datasets, it is time-consuming to access and render even a single tile. Recent work has therefore relied on prefetching tiles based on real-time models of user intent,⁵ which are nontrivial to build. Even more problematically, precomputation is infeasible for high-dimensional data because it requires a huge set of cubes or tiles. For this reason, most scalable visualization tools based on precomputation have been restricted to low-dimensional datasets such as maps.

In contrast, *sampling*, or choosing a subset of elements to estimate the properties of the entire set, offers an attractive alternative and/or supplement to precomputation for visual analytics tools. Sampling is fast and flexible, and it works well in high dimensions. It addresses both computational and perceptual/cognitive problems at once, which makes it a viable, practical approach to scalable visual analysis.

Sampling Challenges

A major issue that has impeded the adoption of sampling for visualizations is the concern that the uncertainties induced by sampling can amplify perceptual or cognitive biases. For example, the sampling process might omit rare but important data items, leading to erroneous perceptions. Such sampling-based uncertainties, coupled with other sources of uncertainty from data processing and mapping, can adversely affect user trust levels or can lead to misconceptions if the user is overly trusting.⁶

Thus, we must address a key question: How can visualization help mitigate such biases both by ensuring that the sample reveals important structural features of the entire dataset and by communicating the uncertainties arising from the sampling process? Because a given sample may be used for multiple visualizations, we hypothesize that users will sometimes want to directly interact with the sample itself, which will require both highly controllable sampling mechanisms and feedback on sample quality. More generally, the visualization community lacks an adequate understanding of how users interact, or might interact, with sampling in visual analytics tools and how sampling affects the user's experience and comprehension.

Two decades ago, researchers in the database community began to address some of these challenges. Here, we begin by reviewing some key results from this work. We then place these results

in the context of a general framework for visual analytics and point to research directions for the visualization community.

Sampling and Visual Analytics in Database Research

The 1990s-era Control project at the University of California, Berkeley,³ in collaboration with IBM and Informix, focused on providing the user with both feedback and control during query processing (see control.cs.berkeley.edu). Standard practice was for a user to submit a query and then wait for a completely accurate result to be returned at the end of query execution. For large datasets, this could take a long time, and in the interim, the user was provided with no useful information. For aggregation queries such as SUM, COUNT, and AVG, on-line analytic processing (OLAP) systems attempted to reduce the processing delay using a pure pre-computation approach, but as a consequence, they could only support a rigidly defined set of queries.

The key observation behind the Control project was that approximate query results often suffice for planning purposes—for example, it suffices to know that annual West Coast sales are roughly \$300 million, rather than knowing the exact value of \$299,685,422.

Online Aggregation

The first result from the Control project was a system for online aggregation.⁷ Primarily, this provided a GUI that allows the user to observe a query's progress and control the execution. The idea is to compute ever-more precise results based on a continually growing data sample; to this end, data is accessed in random order.

Figure 1 shows two screen shots of an online aggregation system executing a query of the form `SELECT AVG(TEMPERATURE) FROM READINGS GROUP BY SENSOR_ID` over an input table with 327,296 rows. Figure 1a shows the running query results after 74 rows of the input table have been scanned. The dots represent statistical point estimates of the final query results—that is, of the true average temperatures for the five sensors based on all the data. Each point estimate is statistically unbiased in that its expected value equals the true average temperature for the corresponding sensor. The uncertainty due to sampling is indicated by the error bars bracketing the dots. These vertical lines represent simultaneous 99 percent statistical confidence intervals (CIs). That is, with a probability of 99 percent, the true average temperatures for the sensors all lie between the upper and lower endpoints of their respective

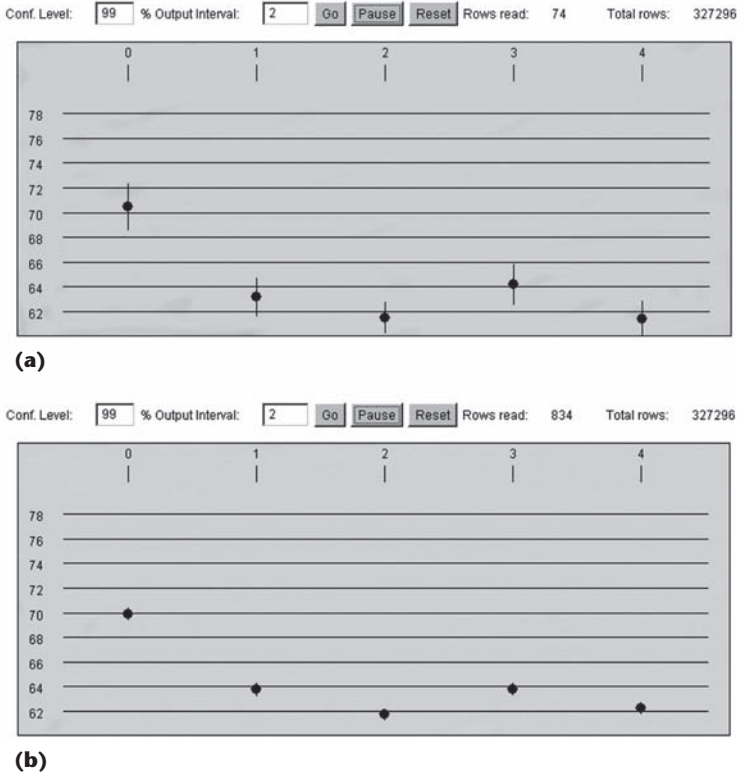


Figure 1. Screenshots of an online aggregation user interface: query results (a) after 74 rows (0.02 percent) and (b) after 834 rows (0.25 percent) of the input table have been scanned. The dots represent statistical point estimates of the final query results, and the error bars bracketing the dots indicate the uncertainty due to sampling.

only 74 rows (0.02 percent of the data) have been scanned, the CIs for sensors 1–4 overlap, so the relative ordering of the sensors by average temperature is uncertain. It is clear, however, that sensor 0 has the highest average temperature (with 99 percent confidence). If this information suffices for the analyst, then she can immediately terminate the query and move on to the next query, perhaps drilling down on the data for sensor 0 in search of an explanation. If the relative ordering of all sensors is of primary interest, then the analyst can let the sampling process continue. As Figure 1b shows, after 834 rows (0.25 percent) have been processed, the CIs are extremely short, and the relative ordering of average temperatures is now apparent. Thus, the user can trade off time and precision on the fly.

Although not visible in Figure 1, the GUI provides the user with additional control. By hovering the cursor over and then right-clicking a given dot, the user can abort the average-temperature calculation for a given sensor, allowing more system resources to be devoted to speeding up the CI shrinkage rate for the other sensors. The user can also explicitly speed up or slow down the sampling rates for individual sensors. To support this functionality,

some precomputation is required. Specifically, the system needs to build an index on the grouping attribute (the sensor ID) in order to use an *index-striding technique*, which is essentially a form of stratified sampling. Such an index also addresses the problem of sampling from small groups.

There have been many improvements on this basic method since its introduction. For example, researchers have extended the standard statistical confidence formulas used in Figure 1 to handle queries that involve relational operations (such as selection, projection, and join),⁸ relaxed the requirement for random data access using Bayesian estimation techniques,⁹ and in the DBO database system,¹⁰ improved scalability via both sophisticated systems engineering for Hadoop platforms and improved CI formulas.

Even given these improvements, the overall approach still assumes that the user understands CIs. However, a subsequent user study of a (simulated) online aggregation system indicated that many, but not all, users are comfortable with the notion of shrinking CIs and that there is substantial room for improvement so that such a GUI can handle scaling issues, dirty data, and so on.¹¹

Combining Sampling and Precomputation

A critical drawback of pure online aggregation is that the presence of small groups can seriously degrade performance; indeed, it might be necessary to scan almost the entire table to achieve a desired CI width. One method for dealing with this issue is to combine sampling with precomputation, as with the use of indexes we mentioned previously. This approach trades off flexibility for performance. An early example of this idea is the Bell Labs AQUA project,¹² which developed sampling-based data synopses that could be used to obtain quick approximate query answers. Unlike online aggregation, where the user can adjust precision on the fly, the precision of an approximate answer in AQUA is fixed at the time of synopsis creation.

A more recent example of a hybrid sampling and precomputation approach is BlinkDB,¹³ which uses ideas from OLAP and sampling to provide an approximate, distributed system for efficient query processing. BlinkDB requires that the columns used by queries in `WHERE`, `GROUP BY`, and `HAVING` clauses are fixed over time. Such columns are called query column sets. In an initial sample creation step, a collection of disjoint samples is generated that will support a range of queries. In essence, entries from the query column sets are selected such that the rare groups are over-represented while optimizing for a user-specified

storage size. The samples created can efficiently provide approximate answers to queries involving both rare and common groups. At query time, a sample subset is dynamically and heuristically selected to compute the query, optimizing with respect to user-specified constraints on sampling error and processing time. Like DBO, BlinkDB is implemented on the top of Hadoop and employs a MapReduce-based distributed file system. BlinkDB is more flexible than AQUA because precision can vary from query to query. However, it is less flexible than online aggregation, where error bounds can be adjusted on the fly based on user perception, and there is no a priori limit on attainable precision.

More recently, Muhammad El-Hindi and his colleagues combined online aggregation with a novel VisTree multidimensional index structure designed to speed up brushing, linking, and zooming for histogram-like data summaries.¹⁴ When reacting to a brushing operation, for example, the system uses partially loaded VisTree information to guide sampling in order to provide estimates for regions in which exact distribution data is not available. The resulting uncertainty is communicated to the user via error bars on the histogram.

Other work in a related vein optimizes the sampling process to efficiently satisfy a prespecified perceptual requirement. For example, Uwe Jugel and his colleagues choose the sampling rate such that sampling-induced fuzziness in the visualization is smaller than the resolution of a screen pixel, so the user cannot discern the difference between the exact and approximate answer.¹⁵ Given the high resolution of modern displays, this approach may be too costly unless the visualization only occupies a small fraction of the screen area. Daniel Alabi and Eugene Wu went further, proposing direct exploitation of perceptual models to determine the sample size at which errors are imperceptible.¹⁶ Finally, Albert Kim and his colleagues proposed a sampling algorithm for scenarios in which the visual property of interest is the ordering of some

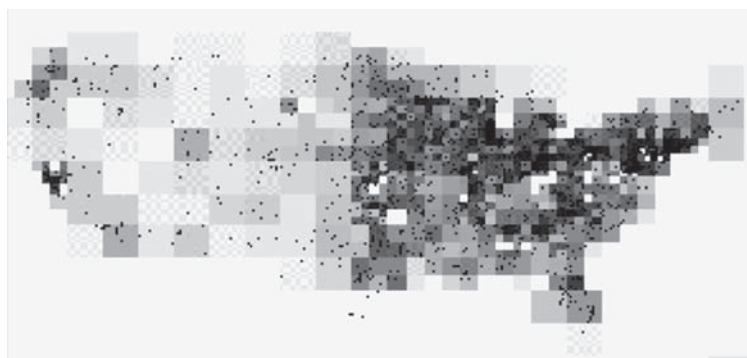


Figure 2. CLOUDS visualization of US cities. The dots indicate the cities that have been rendered so far, and the shading approximates the final density of different areas.

quantity among different groups, as in a bar chart of group counts.¹⁷ Their algorithm preserves the desired ordering at all times in the course of the execution and is very fast. Instead of picking a sample from each group in a naïve round-robin manner, the algorithm draws samples from groups with overlapping CIs.

Approximate Visualizations

The database community has primarily focused on the visual analysis of aggregate quantities such as sums and averages. One exception is the CLOUDS interface from the Control project. Figure 2 shows an intermediate phase in an online visualization of US cities. The dots correspond to cities that have been rendered so far, and the shading approximates the final density of different areas, based on an online-aggregation-style computation. Such visualizations let users pan and zoom, under the assumption that the accuracy of what is seen is not as important as the rough sense of the moving picture. When the user ceases moving over the visual canvas, more data is fetched and rendered over time for the visualized region, causing the clouds to gradually “lift.”

General Issues in Sampling for Visual Analysis

Although a source of many interesting ideas, prior database research on sampling and visual analysis raises many questions and leaves much

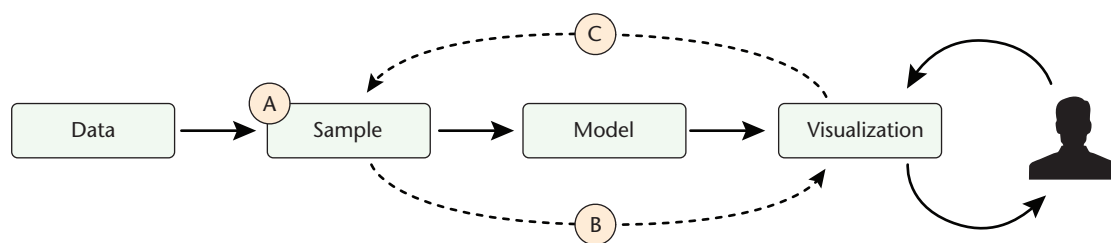
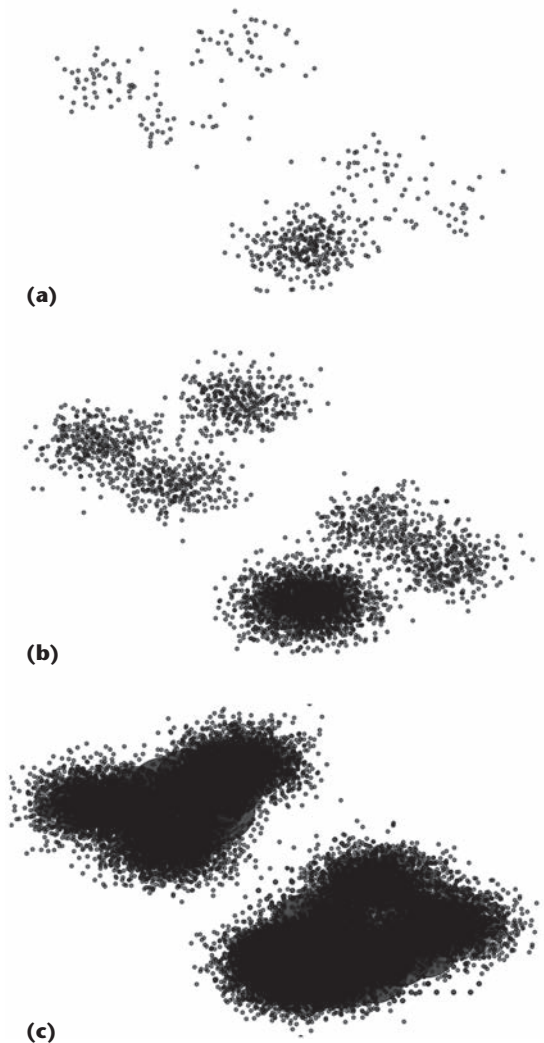


Figure 3. A visual analytics framework with sampling. We can augment the knowledge generation model by adding (A) sampling methods, (B) sample visualizations, and (C) interactivity between the user and the sampling process.

Figure 4.
Visualizing
point clusters
using sampling:
(a) 1 percent
sample,
(b) 10 percent
sample, and
(c) 100 percent
sample.



room for generalization and improvement. In this section we explore some of these ideas from a broader perspective. In general, we can augment the knowledge generation model¹⁸ with a sampling process, as Figure 3 shows. In the following discussion, we revisit questions related to sampling methods, sample visualization, and interactivity between the user and the sampling process.

Sampling and Perception

Sampling is a two-edged sword. On the one hand, sampling can reveal the structure of a dataset by removing visual clutter. On the other hand, sampling too few elements can hide structure because of data sparsity. Effective deployment of sampling for visual analysis requires a judicious choice of sampling and visualization methods, along with good values for the associated parameters.

For instance, consider a dataset containing 50,000 2D data points. As discussed earlier, if we visualize 100 percent of the data points (see Figure 4c), we can see groups but the visual clutter

masks the fine structure of the data. One could argue, along with Fisher,² that a scatterplot is inappropriate and that a contour plot or heat map of point densities should be used instead. However, those latter visualizations do not let users select and drill down on individual data points. If we wished to use a scatterplot, we could use sampling to reduce visual clutter. As Figure 4a shows, for our example dataset, a 1 percent sample yields points that are too sparse to clearly reveal the grouping structure. On the other hand, Figure 4b shows that a 10 percent sample communicates both cluster and density information nicely. Perhaps sampled points could be combined with a density plot similarly to Figure 2, in which case the optimal sampling rate might differ from that in Figure 4.

Besides the sampling rate, various visualization parameters can be manipulated to effectively convey structural information. Figure 5 shows the effects of manipulating the opacity and radius of the scatterplot points to clearly reveal clusters. Alternatively, it might be possible to develop a brushing mechanism that works well on a density plot so that individual points do not need to be rendered at all.

In general, a complex interplay exists between visualization methods, sampling rates, data topologies, and visual parameters. The design of effective visual methods for sampled data is a vast, uncharted research area.

Visualizing Sampling-Induced Uncertainty

To engender trust and avoid bias, it is essential to communicate to the user the uncertainty in a visualization that is induced by data sampling. Almost all work so far has used CIs to communicate uncertainty. Although such intervals may suffice for aggregation queries (given sufficient statistical knowledge), how do we communicate uncertainty for other types of possibly complex visualizations? For some visualizations, such as trend fits via regression, it is possible to compute confidence bands, a straightforward generalization of CIs, by modifying traditional confidence-band formulas with finite-population corrections (see Figure 6). In many other cases, however, it is not obvious how to visualize uncertainty.

The following example, albeit rather crude, illustrates how to apply bootstrap-resampling techniques to this end. Such techniques are widely used to quantify uncertainty in complex statistical settings. Suppose that the goal is to identify clusters in a 2D dataset. One approach is to use a kernel density (KD) estimator and define a cluster's boundary as a low-valued isodensity contour.

After taking a sample of n points, we generate 100 bootstrap samples, where each bootstrap sample consists of n points sampled uniformly with replacement from the original sample. We then compute a KD estimate for each bootstrap sample, which yields a set of cluster boundaries. In Figures 7a and 7b, the light gray curves correspond to a 10 and 80 percent sample, respectively. (We used low-opacity black curves to obtain some shading.) To compute a point estimate of the true cluster boundaries, we average 100 KD estimates and then compute contours; these contours are shown in black in Figure 7. The red curves correspond to the true cluster boundaries based on the entire dataset. As the visualizations show, the gray uncertainty region shrinks and the estimated contours approach the true contours as the sample size increases.

A more refined version of this procedure might consolidate the gray lines into a solid (perhaps shaded) uncertainty band around the estimated contours, similarly to Figure 6. (We could also take multiple samples instead of bootstrapping a single sample. Bootstrapping is typically much faster, however, especially if analytical bootstrapping techniques¹⁹ can be used.) The procedure can undoubtedly be improved in several ways, but this example illustrates that uncertainty representations can potentially be extended beyond CIs. In general, a user's confidence in observations inferred from a sampled visualization is not based merely on statistical measures per se, but is deeply rooted in the visual design and interaction provided by the system.

Interacting with the Sampling Process

How can a visual analytics system endow trust in a sample? In the previous section, we discussed methods for building trust by communicating uncertainty in the context of a specific visualization. For example, we can use a resampling method to assess the sensitivity of a visualization-based inference to perturbations of the sample. Typically, however, a sample will be used in multiple visualizations, so it may be important for a user to assess a sample on its own merits and modify it interactively.

Whereas samples traditionally have been chosen to be "representative" of the population in some global sense, systems such as online aggregation let users interactively drive the sampling process according to their evolving interests. Online aggregation offers rudimentary interactive control of sampling in that users can adjust the sampling rate, or stop sampling altogether, for individual groups. It seems natural to generalize this idea

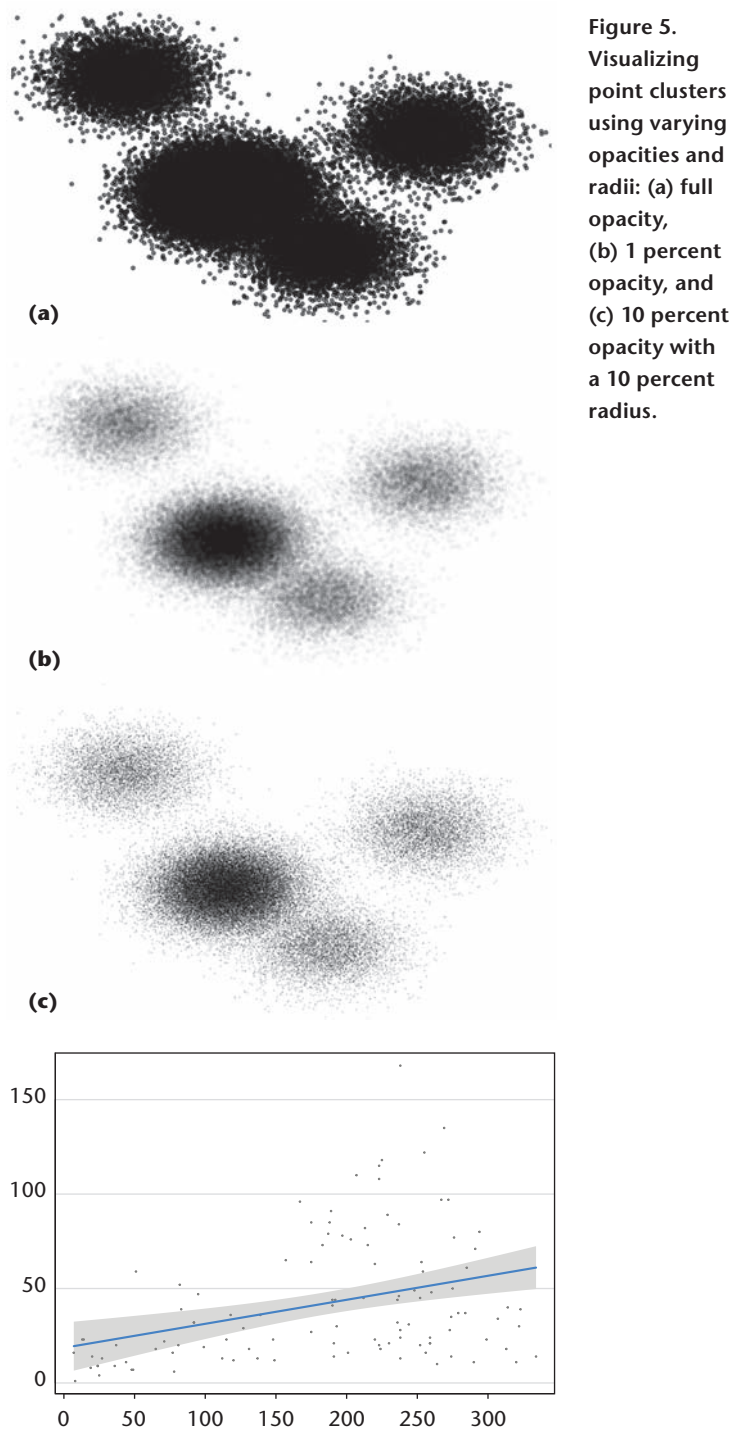


Figure 5. Visualizing point clusters using varying opacities and radii: (a) full opacity, (b) 1 percent opacity, and (c) 10 percent opacity with a 10 percent radius.

Figure 6. Sample points and 95 percent confidence bands for a linear trend fit. Modifying traditional confidence-band formulas with finite-population corrections helps visualize uncertainty.

to allow users to dynamically point the sampling process toward specific, interesting regions of the data domain. Users may want to steer the sampling process by explicitly specifying sample characteristics or criteria. Or with a mouse, a user could indicate regions of interest on a visual representation of the data space. Thus, at any

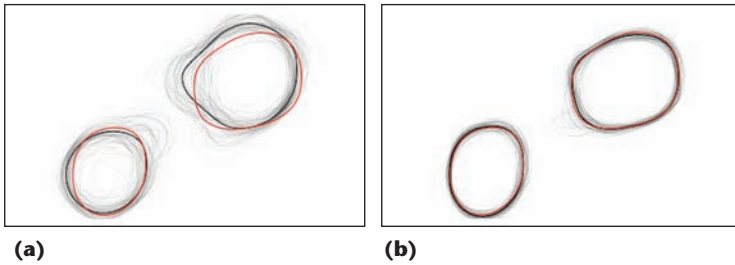


Figure 7. Visualization of 2D clusters using density estimation and resampling: (a) 10 percent and (b) 80 percent samples. The light gray curves correspond to bootstrap samples, black curves indicate point estimates of the true cluster boundaries, and red curves show the true cluster boundaries based on the entire dataset.

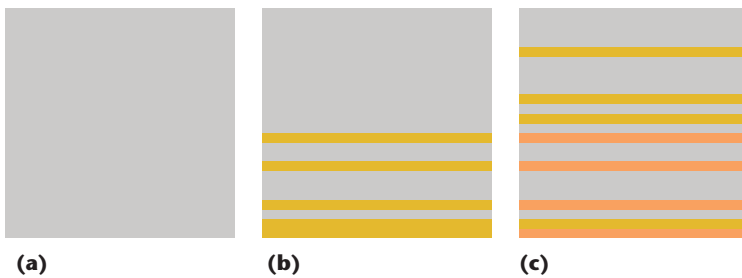


Figure 8. A data barrel view. (a) The grey area represents all rows, (b) the colored rows represent a sample's data coverage, and (c) the orange indicates rows shared by multiple subsamples.

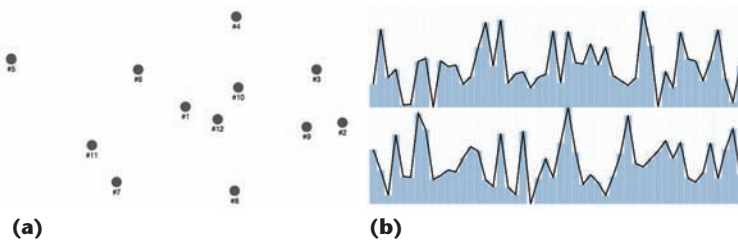


Figure 9. Visualizing subsamples. (a) In this scatterplot, each subsample is represented by a dot, and the distance between a pair of dots represents the similarity of the corresponding subsamples. (b) The histogram of an attribute value in two different subsamples allows a side-by-side comparison.

point in time, the current sample may be viewed as a union of dynamically produced subsamples from different regions. Each subsample aims to be representative of its locale.

Under this complex sampling regime, a user must be able to assess a sample's quality. Visualization-based techniques seem particularly appropriate here because a mathematical statistical analysis is likely to be complex. A typical criterion for a set of subsamples is that they exhibit good coverage over the regions of interest. Moreover, it is usually desirable for the samples to be disjoint: resampling a previously sampled point adds no new information in a statistical sense, and multiply-resampled points might unduly influence a given visualization. When

sampling a table's rows, we can use a barrel plot (as in Figure 8) to indicate subsample coverage and overlapping subsamples as well as the overall percentage of sampled rows. The latter has been shown to affect user confidence.¹¹ One can easily envision variants of the plot, such as in a heat map format, for more general data domains.

We expect that users will typically define subsamples in terms of a finite subset of data attributes. They will then be interested in comparing properties of the subsamples with respect to both these attributes and others. Users will also want to evaluate subsample properties with respect to the analysis task at hand. One simple approach is to provide summary statistics on each subsample such as mean and standard deviation¹¹ as well as min, max, quantiles, higher-order moments, and so on.

More generally, we can view a subsample as a point in a vector space or a manifold. Using dimensionality reduction techniques, we can project the samples onto two dimensions and then display them as a scatterplot. Each subsample is represented by a dot, and the distance between a pair of dots represents the similarity of the corresponding subsamples (see Figure 9a). Furthermore, we can show the distribution of attribute values over the subsamples; Figure 9b illustrates this using histograms.

What is the best way to present users with dynamically generated subsamples? The design space is vast. As more subsamples are materialized, the potential for visual clutter increases, so one possibility is to assign these subsamples to different layers and display multiple subsamples by overlaying them (see Figure 10). We can select collections of subsamples for display based on both data characteristics and the stages at which the subsamples were collected.

The techniques illustrated in Figures 9 and 10 can also be applied to bootstrapped subsamples of a given sample, in which case there will be large overlaps between subsamples. It is then desirable to compare the subsamples directly to see if sample features of interest persist from one subsample to another. Besides using layers, one can visualize bootstrap subsamples via a small multiples approach (see Figure 11). One disadvantage of this approach is that it can be difficult to visually compare subtle differences between subsamples by viewing juxtaposed thumbnails. We can make such subtle differences more apparent by animating the transition between subsamples. We can also augment the visual comparison by quantifying dissimilarities between two given samples with various statistical distance measures such as the Kullback-Leibler divergence and the Hellinger distance.

The design issues we have discussed in this section are closely related to those arising in *progressive visualization*.²⁰ PV techniques apply to analysis algorithms, such as k-means clustering, that let us display intermediate results to the user. PV does not address sampling or scalability questions directly, but it does share key design principles, including the ability to focus attention on data regions of interest. An important point raised in earlier work²⁰ is the need to update visualizations in a timely, but visually nondistracting manner. This might require user control over display updating rates and the careful design of visual cues.

Future Directions

Significant progress around design and user experience is needed if sampling-based techniques are to be widely adopted for visual analysis. The goal should be to understand how sampling can enhance user experience with visual analytics. With this in mind, our discussion indicates three important directions for future research in the visualization community:

- *Understanding the interplay of sampling and perception.* Little is known about how sampling affects user comprehension. Which visualizations are amenable to sampling? How can we combine sampling and other visualization methods to best allow a user to perceive the key patterns in a dataset?
- *Communicating sampling-based uncertainty.* A key aspect of user experience with sampling is dealing with sampling-induced uncertainty. Even basic CI methodology needs improvement to enhance usability.¹¹ More broadly, how do we visualize uncertainty in analytic settings beyond simple aggregation queries? How can we accommodate users who are not experts in statistics? Should we instead try to make sampling uncertainty imperceptible to the user?
- *Enhancing user interactivity.* Users must feel comfortable with the sampling process independent of any particular analysis visualization. How can we give users more dynamic control over the sampling process? Research is needed to develop mechanisms for steering the sampling process beyond the simple group-oriented controls of an online aggregation system. Such steering mechanisms in turn require system feedback about a sample's quality—or its constituent subsamples—in terms of representativeness, structural fidelity, and coverage. We need to develop, evaluate, and visualize such quality measures.

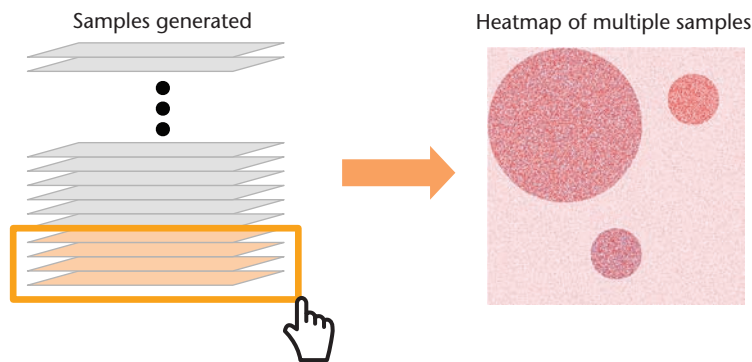


Figure 10. Interactive exploration of subsamples. Assigning subsamples to different layers allows users to select collections of subsamples and explore them based on both data characteristics and the stages at which the subsamples were collected.

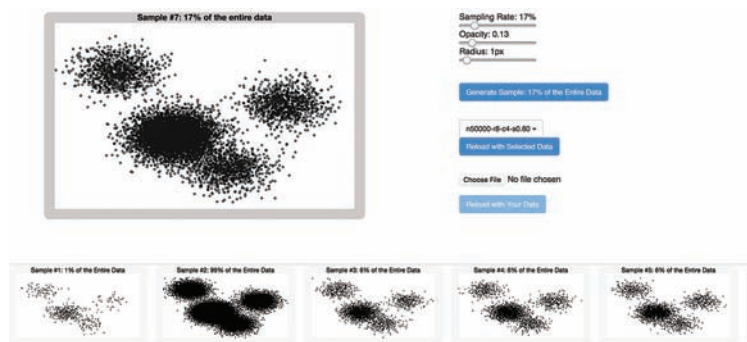



Figure 11. Small multiples for bootstrap subsamples. Users can select a thumbnail from the options on the bottom to view in detail.

All three of the foregoing research directions require the development of well-grounded models that capture user perception and behavior (trust) with respect to sampling. To this end, researchers must conduct careful studies of how users interact with sampling in visual analytics tools.¹¹

As Fisher articulated,² research on sampling for visual analysis must be pursued in close collaboration with the database community. Database researchers need to provide mechanisms that enable users to efficiently sample not just high probability regions but also rare data points, in an ad hoc, flexible manner. Fast computation is needed not only for processing and rendering the data, but also for executing the statistical analysis that underlies uncertainty visualization and sample-quality assessment. This suggests the need for a hybrid approach that exploits both precomputation and sampling. Moreover, visualization methods should take advantage of modern parallel and distributed systems and specialized hardware such as GPUs.²¹

Enabling the visual analysis of large datasets while sustaining an interactive user experience is

an important challenge with many facets. Sampling will be an effective tool in addressing this challenge. Indeed, the enormous recent progress in underlying database technology makes this an especially propitious time to incorporate sampling into visual analytics. 

Acknowledgments

The authors thank Theresa-Marie Rhyne and the anonymous reviewers for their many suggestions, which greatly improved this article.

References

1. "What Was the Largest Dataset You Analyzed/ Data Mined?" poll, KDnuggets, Aug. 2015; www.kdnuggets.com/polls/2015/largest-dataset-analyzed-data-mined.html.
2. D. Fisher, "Big Data Exploration Requires Collaboration Between Visualization and Data Infrastructures," *Proc. Workshop on Human-in-the-Loop Data Analytics (HILDA)*, 2016, article no. 16.
3. J. M. Hellerstein et al., "Interactive Data Analysis: The Control Project," *Computer*, vol. 32, no. 8, 1999, pp. 51–59.
4. J. M. Hellerstein, "Interactive Analytics," *Readings in Database Systems*, 5th ed., P. Bailis, J.M. Hellerstein, and M. Stonebraker, eds., MIT Press, 2015.
5. L. Battle, R. Chang, and M. Stonebraker, "Dynamic Prefetching of Data Tiles for Interactive Visualization," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 2016, pp. 1363–1375.
6. D. Sacha et al., "The Role of Uncertainty, Awareness, and Trust in Visual Analytics," *IEEE Trans. Visualization and Computer Graphics*, vol. 22, no. 1, 2016, pp. 240–249.
7. J.M. Hellerstein, P.J. Haas, and H.J. Wang, "Online Aggregation," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 1997, pp. 171–182.
8. F. Li et al., "Wander Join: Online Aggregation via Random Walks," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 2016, pp. 615–629.
9. N. Pansare et al., "Online Aggregation for Large MapReduce Jobs," *Proc. VLDB Endowment*, vol. 4, no. 11, 2011, pp. 1135–1145.
10. C. Jermaine et al., "Scalable Approximate Query Processing with the DBO Engine," *ACM Trans. Database Systems*, vol. 33, no. 4, 2008, article no. 23.
11. D. Fisher et al., "Trust Me, I'm Partially Right: Incremental Visualization Lets Analysts Explore Large Datasets Faster," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems*, 2012, pp. 1673–1682.
12. P.B. Gibbons and Y. Matias, "New Sampling-Based Summary Statistics for Improving Approximate Query Answers," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 1998, pp. 331–342.
13. S. Agarwal et al., "BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data," *Proc. 8th ACM European Conf. Computer Systems*, 2013, pp. 29–42.
14. M. El-Hindi et al., "VisTrees: Fast Indexes for Interactive Data Exploration," *Proc. Workshop on Human-in-the-Loop Data Analytics (HILDA)*, 2016, article no. 5.
15. U. Jügel et al., "M4: A Visualization-Oriented Time Series Data Aggregation," *Proc. VLDB Endowment*, vol. 7 no. 10, 2014, pp. 797–808.
16. D. Alabi and E. Wu, "PFunk-H: Approximate Query Processing using Perceptual Models," *Proc. Workshop on Human-in-the-Loop Data Analytics (HILDA)*, 2016, article no. 10.
17. A. Kim et al., "Rapid Sampling for Visualizations with Ordering Guarantees," *Proc. VLDB Endowment*, vol. 8, no. 5, 2015, pp. 521–532.
18. D. Sacha et al., "Knowledge Generation Model for Visual Analytics," *IEEE Trans. Visualization and Computer Graphics*, vol. 20, no. 12, 2014, pp. 1604–1613.
19. K. Zeng et al., "The Analytical Bootstrap: A New Method for Fast Error Estimation in Approximate Query Processing," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 2014, pp. 277–288.
20. C.D. Stolper, A. Perer, and D. Gotz, "Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics," *IEEE Trans. Visualization and Computer Graphics*, vol. 20, no. 12, 2014, pp. 1653–1662.
21. Z. Liu, B. Jiang, and J. Heer, "imMens: Real-time Visual Querying of Big Data," *Computer Graphics Forum*, vol. 32, no. 3, part 4, 2013, pp. 421–430.

Bum Chul Kwon is a data analytics and visualization researcher at the IBM T.J. Watson Research Center. Contact him at bumchul.kwon@us.ibm.com.

Janu Verma is a research engineer at the IBM T.J. Watson Research Center. Contact him at jverma@us.ibm.com.

Peter J. Haas is a principal research staff member at the IBM Almaden Research Center. Contact him at phaas@us.ibm.com.

Çağatay Demiralp is a researcher at the IBM T.J. Watson Research Center. Contact him at cagatay.demiralp@us.ibm.com.

Contact department editor Theresa-Marie Rhyne at theresamarierhyne@gmail.com.