The dark side of Progressive Visual Analytics

Marco Angelini and Giuseppe Santucci
Sapienza University of Rome
Email: [angelini][santucci]@dis.uniroma1.it

***Abstract*—Big data, complex computations, and the need for fluent interaction, are the well known enemies of Visual Analytics. They can seriously impair the fluent interactive back and forth between computational analysis and human analyst that happens using the visualizations that a Visual Analytics solution provides for supporting this combined analysis. The emerging Progressive Visual Analytics (PVA) took the field and wage a war against such enemies, providing a way around this conundrum by iteratively computing partial results of increasing quality, that constitute a natural means for providing the analyst with early and continuous interaction with the initial results even while the whole process is still far from being completed. However, this promising solution has several implications that must be dealt with, requiring to address different issues. This paper has the goal of providing a travel guide about the practical usage of PVA, discussing the motivations that call for its usage, the possible strategies that can be used to solve the problem and the fee that requires to be payed in order to use this approach. In order to provide a concrete discussion, the paper is driven by a concrete and intriguing example, an intractable PVA solution developed for Telecom Italia Mobile (TIM).**

I. Introduction

Progressive Visual Analytics (PVA) [1], [2] is the recently agreed name for an underlying idea that was around with different wording and used for addressing a variety of different perspectives; a non exhaustive list includes *Fine-Grain Visualization* [3] – from the perspective of dataflow computing – *Online Visualization* [4] – from the perspective of database query processing –, *Progressive Information Presentation* [5] – from the perspective of image transmission – *Incremental Visualization* [6] – from the perspective of human-computer interaction – *Progressive Visualization* [7] – from the perspective of volume rendering – and *Progressive Analytics* [2] – from the perspective of computational analyses.

While each of these proposals deals with issues and details that are associated with the current problem they are working on, their common denominator is to "produce partial results during execution" [1] to solve a problem. So, as a first consideration we assume that PVA is needed when a *problem* exists and everything else (e.g., parallel computation, faster CPUs, etc.) but PVA fails, and that the solution relies on the production of *partial results*.

For the discussion in this paper, we assume that these partial results are visualizations $v_1, v_2, \ldots, v_n$ yielded at time points $t_1, t_2, \ldots, t_n$, respectively. At each instant t_i, some processed data is available, used for driving the visualization, namely $d_i = \{a_i, s_i\}$, where a_i represent a series of computed values (e.g., count, mean, standard deviation, clusters composition, values maximizing a function, etc.) and s_i constitutes the representation of the actual state of the system (e.g., percentage of consumed data or algorithm iteration progress). According to the particular nature of the application and of the aggregation function, an error $\varepsilon(a_i)$ can be introduced during the computation of d_i (see, e.g., [8]).

For the remainder of the discussion, the paper will adhere to the basic definition of *partial results* given above, making possible to characterize the concept in all its generality.

According to [9] PVA can generate partial results in two ways: by subdividing the data and processing it chunk by chunk, or by subdividing the computational process and running it step by step. In the first case, which we call *data chunking*, the partial results v_i relies on increasingly using more data over time, until at t_n *all* the data has been used to produce the final output v_n. Whereas in the second case, which we call *process chunking*, the partial results v_i are generated using all the data all the time, but the quality of the computed data d_i increases with each step – from a quick and dirty first computation d_1 to a full final computation d_n. In principle, it would be possible to combine both strategies [9]. However recent user studies show that users seem uneasy to make early decisions based on progressive and changing estimates [10]. This makes clear that, at least, two *implications* may arise by the PVA adoption:

1) Approximation and errors introduced by either *data chunking* or *process chunking*. As discussed in [11], in the most general case PVA can introduce errors and, while it is obvious that at any time $t_i,\ i < n$ PVA is typically affected by an approximation (either the computation has not yet finished or the data has not been fully processed), the way in which PVA has been implemented may affect also the last partial result d_n, i.e., $\varepsilon(a_n) > 0$. As an example, assume that an application needs to visualize the average and the $top5\%$ of a large set of m numerical values and that constraints on the response time exist, making the $m \times log(m)$ sorting cost unacceptable. A viable solution is to chunk the data in $ch_1, \ldots, ch_n$ subsets of size m/n, such that $m/n \times log(m/n)$ matches the time response constraint, and producing at each t_i $mean_i$ and $top5\%_i$ that can be visualized in v_i. It is clear that at any $t_i,\ i < n$ both $mean_i$ and $top5\%_i$ are affected by an error; at t_n $mean_n$ will be the exact mean of the whole data, while $top5\%_n = \cup_{i=1,n} top5\%(ch_i)$ will be only an approximation of the real $top5\%(n)$ of the whole dataset. However, the main point is not whether an error is introduced or not; what is critical is to have a *control*

on it, making the user aware of the current situation;

2) Variability and usefulness of the result. Variable, non converging, and not enough informative partial result may confuse the user, making the PVA process much less effective. Considering the previous example, $mean_i$, after initial variations, will exhibit a somehow converging behaviour, while $top5\%_i$ will be much more variable, being affected by an increasing size and a variation of the ordered number list.

   Many papers add useful assumptions or requirements to their working definition of this fundamental concept. For example, Stolper et al. argue that the produced partial results v_i must be *semantically meaningful* [1] and Fekete and Primet state that the sequence of partial results must adhere to *user-specified bounds* [2], by which they mean given time constraints.

On the basis of the previous discussion, it is possible to acknowledge that PVA applications are motivated by and imply (at least) the following:

- A detrimental problem, i.e., the issue that affects the performance of the designed Visual Analytics (VA) application and calls for the adoption of PVA;
- The design of the PVA solution, dealing with data and process chunking, *characterizing errors and approximations*;
- Issues associated with the partial results v_i, issues that can be mitigated increasing the v_i usefulness, reducing their variability, and giving a clear feedback to the user on what is going on.

While many other aspects are still relevant to the PVA approach it is the authors' believe that the aforementioned aspects are the most relevant ones and they will be discussed along a step by step analysis of a concrete PVA solution.

Summarizing, the contribution of the paper is to explicitly point out problems, implications, and mitigations that stay behind any PVA application, representing its *dark side*.

The paper is structured as follows. Section 2 presents related proposals, Section 3 introduces the demonstration scenario, Section 4 uses the demonstration scenario to discuss issues, implications, and mitigations in PVA, and Section 5 concludes the paper, outlining future work.

II. Related Work

Several papers, in the literature, dealt with the concept of PVA, even using different terminology. As an example one active field is connected with *Data Streaming* [12], [13] mainly dealing with transient dynamic datasets that are continuously generated and possibly infinite. Such data is generated, for example, in the form of news feeds or sensor data. Typical constraints are that it is not feasible to store the whole stream and that, in some cases, it is even impossible to process all the data. The rationale is to obtain, from the first steps of the process, a visualization generated from the whole dataset that will take into account a variable degree of confidence, in contrast with a simple mapping of the subset of data received at each step. Apart from that, it is also employed for transmitting large but not infinite datasets across slow network connections. In both cases, the data comes as a sequence of smaller chunks and is processed as such. Hence, this scenario matches quite well with the PVA idea of data chunking. Another field calling for PVA ideas is the so called *Out-of-core visualization* [14], [15], used when the size of the dataset to be visualized exceeds the available memory space. In order to generate a visualization, the dataset is partitioned into smaller chunks, which are then sequentially passed through the visualization process to reduce its memory occupation. It thus works in the very same spirit as described by the idea of data chunking and partial results.

Recently, the wording Progressive Visual Analytics (PVA) has been consistently used by several visual analytics solutions that use a sequence of partial results, see , e.g., [1], [16]–[18].

What has been presented in [9] is closer to the paper proposal, because it tries to abstract from providing a specific solution and focuses on a graphical notation for *modeling* a process producing intermediate results, process that authors call *Incremental Visualization*.

Besides these fundamental concepts and their usefulness for different application scenarios, the body of related work discusses mainly their technical challenges. These concern mostly their applicability to certain data types and visualization operators that do not lend themselves to a high-level discussion of problems, partial results, implications, and mitigations that instead is the focus of this paper.

III. Demonstration Scenario

The demonstration scenario is PVA solution developed for supporting a Telecom Italia Mobile (TIM) decision making process [19] for analyzing users' distribution across Italy. Such data has been integrated with information coming from open data and mapped on the Italian hierarchy of 20 regions and 110 provinces. The system allows for interactively selecting an optimal set of provinces (top10) according to an objective function that is computationally intractable. In order to have an explorative system, the analytical solution allows for visually selecting a subset of provinces, allowing for a real time interaction with partial results, and leaving the full optimization phase to a post processing step. The typical decision making process, evolves as follows.

1) The analyst inspects a scatterplot (see Figure 1) and select a subset of provinces based on three main parameters: average income, percentage of TIM customers with respect to the total population, number of potential customers in that area;
2) The candidate provinces are subsequently evaluated against an optimization function that relies on traffic information between provinces pairs, computing the top 10 provinces maximizing the function;
3) A sequence of intermediate results, approximating the top 10, is then visualized on a Sankey diagram (see Figure 2), where the position and direction of each line immediately give the user a clear idea of the marketing

and geographical context of each of them; if the user does not like the current top 10 can switch back to the scatterplot and trigger the optimization phase on a different selection; if, conversely, he is fine with the current top 10 he can trigger a background computation that further refine the result. In order to make the user aware of the approximation quality some statistical indicators are presented on the top of the visualization (see figure 2).

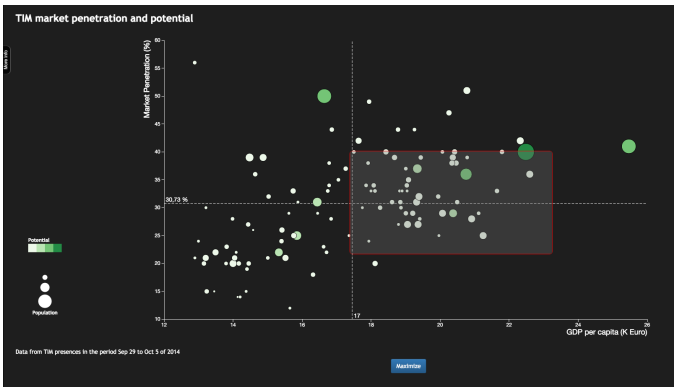

Fig. 1. The scatterplot allows for selecting group of provinces that satisfy some high level campaign scenarios. In the example the analyst has selected provinces characterized by high income and a penetration close to the median; the objective is to promote some additional non basic features (e.g., fast network services) that will likely be accepted by either potential users and TIM customers, in a scenario in which TIM has a good chance of increasing its presence and people GDP is above the median.

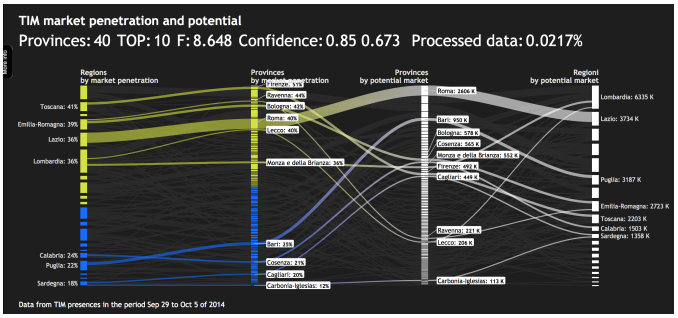

Fig. 2. The intermediate results of the optimizing process are presented on the Sankey plot allowing the analyst to focus on the most convenient 10 provinces. Numerical quality indicators help the user in understanding the PVA progress.

A. The TIM PVA application dark side

The explorative task of the TIM application poses clear constraints on the response time of the <select provinces-visualize top 10> activity that is the main elementary task the user performs on the data; according to [20] this *unit task* dialogue must be completed in an interval between 10 to 30 seconds.

The problem. Respecting this time constraint is not an easy task: in this intriguing PVA example while the data size is neglectable (110 provinces with four attributes and traffic information among province pairs: $110 \times 109/2$, about 40 Kbyte) the optimization phase is computationally not tractable.

Discussion with TIM analysts produced the following top 10 objective function:

$$\sum_{i,j=1}^{10} traffic(p_i, p_j) * \sum_{i=1}^{10} Value(p_i)$$

where

$$Value(p) = 0.3 * market(p) + 0.5 * gdp(p) + 0.2 * (1 - |penetration(p) - median(penetration)|)$$

The rationale is that the 10 selected provinces should present a strong intra-province traffic together with a balanced combination of potential market, high income, and high likelihood of increasing the penetration (the analysts assumption is that the closer to the median the penetration is, the greater the likelihood of increasing it).

All objective function values come from tabular relations and a closed formula for it does not exist, making usual optimization techniques unfeasible: a computation of all the possible provinces combinations is needed, making this otherwise trivial task unfeasible. E.g., if the analyst selects 40 provinces on the scatterplot the number of combinations is $\binom{40}{10}$ that is about $8.5 * 10^8$. Assuming a value of about 3000 calculations per second (actual value on an Intel Core i7, 3.1 GHz), getting the exact maximum value requires about 75 hours. While this is not a hard constraint for a one shot analysis task, it makes an interactive explorative analysis impossible. Moreover, if the size of the user selection increases, getting the exact maximum is not feasible at all: e.g, computing the top 10 on all provinces: $\binom{110}{10}$ is about $4.7 * 10^{13}$, requiring $1.5 * 10^{10}$ seconds, about 500 years.

The solution. To solve this problem we use an adaptive partitioning strategy based on the selection size, computing the function maximum on each partition and merging them afterwards (see 3). More precisely, till a selection of 19 provinces we compute the exact solution (max response time=29 seconds), between 20 and 50 we partition the selected provinces in two subsets (max response time=34 seconds), and above 50 in three subsets (max response time= 19 seconds). This allows for presenting the user with early partial results but introduces errors. In order to estimate the error we experimented the adaptive strategy with different selection size, process granularity, and data chunking, collecting and averaging two measures:

$$FunctionRatio(FR) = \frac{Estimated\ function}{optimum\ value}$$

$$Top10Proportion(TP) = \frac{Estimated\ top10 \cap real\ top10}{10}$$

Figure 4 presents the results of such analysis and give the basis for defining the adaptive chunking thresholds needed to satisfy the required response time and providing indication of the errors produced during the PVA process. Such measures are presented to the user to help his comprehension and making more confident decisions. It is worth to recall that the focus in this section is not on the solution itself, but its

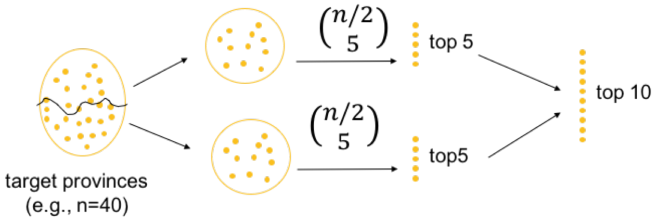

Fig. 3. The PVA solution, producing a first useful result on 40 provinces, in about 10 seconds w.r.t. the 75 hours needed to compute the exact maximum inspecting all the combinations.

use as an example clarifying how to cope with an intractable problem and how to *dominate* the introduced errors. Different PVA applications will likely use different ways of solving the problem and estimating the error (e.g., computing the error using statistics) but the methodology and the final picture will be conceptually very close to what discussed on Figure 4.

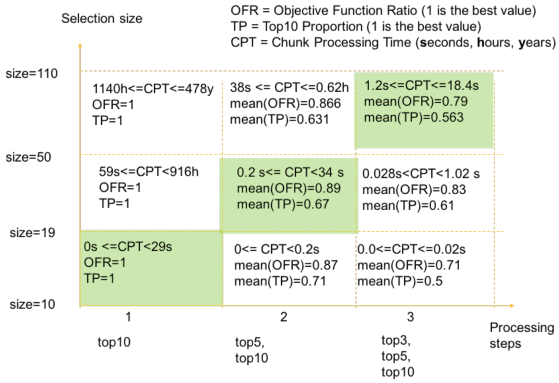

Fig. 4. The PVA solution, chunking data and process. The y axis represents the provinces selection size, ranging from 10 to 110 provinces; the x axis represents the process, ranging from (1) a monolithic process computing $\binom{n}{10}$ on the whole selection, to (2) computing $\binom{n}{5}$ splitting the selection in 2 chunks to produce an early partial result, and then compute $\binom{n}{10}$ on the whole chunk in a longer time, till (3) a process composed by the sequence of computations $\binom{n}{3}$, $\binom{n}{5}$, $\binom{n}{10}$ splitting the selections in 3 chunks and producing an early partial result, refining it using two chunks, and then computing the optimal solution on the whole selection. Within the 9 tiles, the figures associated with *the first useful partial result* report the minimum and maximum time needed to produce it, and the errors, in term of means of the numerical ratio between the estimated function and the optimal one, and the proportion of the provinces belonging to both the estimation and the optimum. The green tiles represent the selected strategies in the current implementation: they respect the time constraints and minimize errors.

Issues associated with the partial results. Variability and approximations in partial results can be dealt assuring the usefulness of what is presented to the user, minimizing the variations, and, most importantly, providing him with clear indication on the status of the process and its quality (requiring error *control*). As an example, in the TIM application we altered the processing of the first visualization v_1, using more time than the one foreseen for the others in order to compute a complete top10 and iterating the process as much as the time constraints allow for to produce a better top10 (i.e., closer to the optimal one) and when a new partial result comes on altering it, we smoothed the transition using animation. Moreover, on the top of the visualization (see Figure 1) the user can observe the following metrics:

- *Provinces*, recalling the size of the selected provinces set;
- *TOP*, showing the target number of provinces to consider in the optimization phase (top10, in the current figure);
- *F*, the current value of the objective function;
- *Confidence*, reporting the Function Ratio(FR) and Top10 Proportion(TP) metrics, providing the user with an indication on how much he can trust the current approximation. The current FR value (0.85) provides for an indication that the current objective function has a value that is about the 85% of the maximum value; the current TR value (0.673) gives him the confidence that about 7 out of the current top10 provinces will be in the optimal top10;
- *Processed data*, a natural indicator that the process is running, providing an understanding on how far the optimum result is; in the current situation, the discouraging value of 0.0217% makes clear that the optimum is really far away, pushing the user to either look for a new search or trigger the background computation if the total number of selected provinces is less than or equal to 40 (the current implementation prevents the user to start background computations longer than three days...).

Once digested, these figures contribute to mitigate the user uncertainty in making decisions; obviously it would be possible to convey this information visually (e.g., using the hue of the top10 province color), but discussing this issue is out of the scope of the paper.

IV. Conclusions & Future Work

This paper attempted to provide a better comprehension of what is behind the scene of a Progressive Visual Analytics (PVA) application, showing the implications that must be dealt with and how to address them. To this aim a formal characterization of the partial results and the associated errors has been provided, using it within a concrete and intriguing example, an intractable combinatorial visual analytics solution developed for supporting one of the decision making processes of TIM, giving an example of motivating problem and a concrete solution, with the goal of abstracting them, showing how to deal with the approximations introduced by the approach and providing examples on how to use these pieces of information to mitigate the visualization issues. Concerning future activities, the authors are currently working on developing a richer classification of problems, solutions, implications, and mitigations, in order to get a better understanding of the problem.

References

[1] C. Stolper, A. Perer, and D. Gotz, "Progressive visual analytics: User-driven visual exploration of in-progress analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1653–1662, 2014.

[2] J.-D. Fekete and R. Primet, "Progressive analytics: A computation paradigm for exploratory data analysis," *arXiv.org e-print*, vol. 1607.05162, 2016.

[3] D. Song and E. Golin, "Fine-grain visualization algorithms in dataflow environments," in *Vis'93:Proceedings of the IEEE Conference on Visualization*, G. M. Nielson and D. Bergeron, Eds. IEEE Computer Society, 1993, pp. 126–133.

[4] J. M. Hellerstein, R. Avnur, A. Chou, C. Hidber, C. Olston, V. Raman, T. Roth, and P. J. Haas, "Interactive data analysis: The control project," *IEEE Computer*, vol. 32, no. 8, pp. 51–59, 1999.

[5] R. Rosenbaum and H. Schumann, "Progressive refinement – more than a means to overcome limited bandwidth," in *VDA'09: Proceedings of the Conference on Visualization and Data Analysis*, 2009, p. 72430I.

[6] D. Fisher, I. Popov, S. M. Drucker, and mc schraefel, "Trust me, i'm partially right: Incremental visualization lets analysts explore large datasets faster," in *Proceedings of the International Conference on Human Factors in Computing Systems*, J. A. Konstan, E. H. Chi, and K. Höök, Eds. ACM Press, 2012, pp. 1673–1682.

[7] S. Frey, F. Sadlo, K.-L. Ma, and T. Ertl, "Interactive progressive visualization with space-time error control," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2397–2406, 2014.

[8] A. Zhou, Z. Cai, L. Wei, and W. Qian, "M-kernel merging: towards density estimation over data streams," in *Database Systems for Advanced Applications, 2003. (DASFAA 2003). Proceedings. Eighth International Conference on*, March, pp. 285–292.

[9] H.-J. Schulz, M. Angelini, G. Santucci, and H. Schumann, "An enhanced visualization process model for incremental visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 7, pp. 1830–1842, 2016.

[10] S. K. Badam, N. Elmqvist, and J.-D. Fekete, "Steering the craft: UI elements and visualizations for supporting progressive visual analytics," *Computer Graphics Forum*, vol. 36, no. 3, 2017, to appear.

[11] M. Angelini and G. Santucci, "Modeling incremental visualizations," in *Proceedings of the EuroVis Workshop on Visual Analytics*, M. Pohl and H. Schumann, Eds. Eurographics Association, 2013, pp. 13–17.

[12] P. C. Wong, H. Foote, D. Adams, W. Cowley, L. R. Leung, and J. Thomas, "Visualizing data streams," in *Visual and Spatial Analysis: Advances in Data Mining, Reasoning, and Problem Solving*, B. Kovalerchuk and J. Schwing, Eds. Springer, 2004, pp. 265–291.

[13] P. C. Wong, H. Foote, D. Adams, W. Cowley, and J. Thomas, "Dynamic visualization of transient data streams," in *Proceedings of the IEEE Symposium on Information Visualization*, T. Munzner and S. North, Eds. IEEE Computer Society, 2003, pp. 97–104.

[14] J. A. Cottam, A. Lumsdaine, and P. Wang, "Abstract rendering: Out-of-core rendering for information visualization," in *Proceedings of the Conference on Visualization and Data Analysis*, P. C. Wong, D. L. Kao, M. C. Hao, and C. Chen, Eds. SPIE, 2014, p. 90170K.

[15] K. I. Joy, "Massive data visualization: A survey," in *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, T. Möller, B. Hamann, and R. D. Russel, Eds. Springer, 2009, pp. 285–302.

[16] N. Pezzotti, B. Lelieveldt, L. van der Maaten, T. Hollt, E. Eisemann, and A. Vilanova, "Approximated and user steerable tSNE for progressive visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, 2016, to appear.

[17] E. Zgraggen, A. Galakatos, A. Crotty, J.-D. Fekete, and T. Kraska, "How progressive visualizations affect exploratory analysis," *IEEE Transactions on Visualization and Computer Graphics*, 2016, to appear.

[18] C. Turkay, E. Kaya, S. Balcisoy, and H. Hauser, "Designing progressive and interactive analytics processes for high-dimensional data analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 131–140, 2017.

[19] M. Angelini, R. Corriero, F. Franceschi, M. Geymonat, M. Mirabelli, C. Remondino, G. Santucci, and B. Stabellini, "A visual analytics system for mobile telecommunication marketing analysis," in *EuroVA'16: Proceedings of the EuroVis Workshop on Visual Analytics*, N. Andrienko and M. Sedlmair, Eds. The Eurographics Association, 2016, pp. 7–11.

[20] S. Card, G. Robertson, and J. Mackinlay, "The information visualizer, an information workspace," in *CHI'91: Proceedings of the Conference on Human Factors in Computing Systems*, 1991, pp. 181–188.