Exploring Big Data Landscapes with Elastic Displays

Dietrich Kammer, Mandy Keck, Mathias Müller, Thomas Gründer, Rainer Groh

Professur für Mediengestaltung, Technische Universität Dresden

Abstract

In this paper, we propose a concept to help data analysts to quickly assess parameters and results of cluster algorithms. The presentation and interaction on a flexible display makes it possible to grasp the functioning of algorithms and focus on the data itself. Two interaction concepts are presented, which demonstrate the strength of elastic displays: a layer concept that allows the recognition of differences between various parameter settings of cluster algorithms, and a Zoomable User Interface, which encourages the in-depth analysis of clusters.

1 Introduction

There is a vast amount of data available on the internet and in industries. The term Big Data is used to refer to this wealth of massive and heterogeneous data sources. Data scientists work on algorithms and technologies that allow to analyze and classify this data. For instance, text-mining is used to find similar data items based on textual information. The resulting multi-dimensional feature spaces are almost impossible to grasp for human users. To this end, multi-dimensional scaling (MDS) can be applied, in order to obtain positional data that can be shown in a two-dimensional scatterplot or a three-dimensional interaction space. One goal is to identify clusters of similar data items to be able to classify existing and new items that are added to the data set. In this case, clustering algorithms and machine-learning are applied, which usually require training data produced by human users.

On the one hand, interfaces are needed that support users in these semi-supervised machine-learning approaches. On the other hand, data analysts are challenged to modify key parameters of the underlying classification algorithms. In each case, numerous variations need to be assessed quickly. We propose to use elastic displays to answer this novel challenge of handling these intractable *Big Data landscapes*. Elastic displays such as FlexiWall (Müller, Knöfel, et al., 2014; Müller, Gründer, et al., 2015) or DepthTouch (Peschke et al., 2012) possess promising features that are suitable for the required applications:

Veröffentlicht durch die Gesellschaft für Informatik e. V. 2017 in
M. Burghardt, R. Wimmer, C. Wolff, C. Womser-Hacker (Hrsg.):
Mensch und Computer 2017 – Workshopband, 10.–13. September 2017, Regensburg.
 https://doi.org/10.18420/muc2017-ws08-0342

- Intuitive Stacking of Views (Müller, Knöfel, et al., 2014): Differences between layouts can be quickly recognized by exploring them in a layered data space. To this end, different MDS algorithms can be assessed (see Layer Concept, section 3.1 and 3.2.1)
- Natural Zoomable User Interfaces: different levels of detail can be easily explored and thus comparison of clusters is possible (see Comparison Concept, section 3.2.2)
- Gestural Interaction with Force Feedback: cluster correction is afforded by using hand gestures with immediate and graded force feedback supplied by the elastic surface

2 Related Work

This section gives a brief overview of the current state of the research about elastic displays and the means of clustering used in our concepts.

2.1 Elastic Displays

The term Elastic Display refers to displays that extend the direct manipulation paradigm of touch interaction by using the deformation of the surface for interaction. Typically, elastic displays are realized using depth cameras such as Microsoft Kinect or Intel RealSense that detect the deformation of the surface onto which images are rear-projected. The expressiveness of the interaction is increased, allowing more control over the manipulation when working with complex data sets. In contrast to spatial gestures, the deformable surface offers haptic feedback and establishes natural boundaries of the interaction space. Furthermore, the additional dimension enables complex gestures (Troiano et al., 2014) and extended concepts for the use of tangibles, e.g. by using physical shapes to control the deformation or span a physical grid on top of the elastic fabric (Müller, Knöfel, et al., 2014). Accordingly, the concept of gravibles – tangibles that use gravity to persist the deformation – has been proposed for elastic tabletop surfaces (Gründer et al., 2013). The use of physics metaphors (Agarawala and Balakrishnan, 2006) or concepts like Reality-Based Interaction (Jacob et al., 2008) allow to playfully interact and explore complex data sets. Furthermore, metaphors derived from natural concepts such as Inertia, Force (e.g. Mass and Gravity), Collision or Friction support the intuitive understanding of relations. Due to more complex tracking, elastic displays are less accurate than touch displays and introduce latency from smoothing algorithms and depth image analysis. Other issues include the inconsistent depth deviation over the surface.

There are several design guidelines that can be derived from observations of former prototypes of elastic displays:

- Natural undo: the surface always returns to its initial state. This supports the use of trial and error to explore the system and its contents, and invites users to experiment.
- Quick assessment: direct interaction and immediate visual and haptic feedback for comparing layers, finding relations by gradually changing the amount of pressure applied and observing the changes in the visualization.

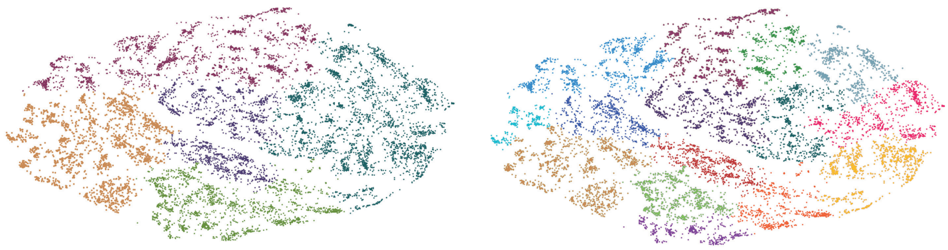

Figure 1: BIRCH algorithm with clusters represented by colors: 5 clusters (left) and 15 clusters (right)

- Force simulation: Selecting, grouping, and filtering operations are easier to understand when using physics metaphors.
- Orientation: tabletop use is preferred for collaborative use and planning scenarios, communication and interaction around the table is more direct compared to the use on an elastic wall which is more frequently used to present content or explore details, e.g. by pushing to zoom into details.

2.2 Clustering

There is a wide variety of approaches to group data. Each one focuses on different properties of a given dataset. Some partition the dataset by neighborhood relations like Agglomerative Clustering, e.g. (Sibson, 1973), and K-Means (Arthur and Vassilvitskii, 2007), whereas others use density like DBSCAN (Ester et al., 1996) or representatives and centroids like Affinity Propagation (Frey and Dueck, 2007). Furthermore, these approaches can be combined as in the BIRCH Algorithm (Zhang et al., 1996) or spectral clustering algorithms (Ng et al., 2002). These algorithms are mathematically complex and have parameters to optimize their execution and results. A data analyst needs to be able to understand the algorithm and the impact the parameters have on the results. The parameters can be the number of clusters (K-Means, Spectral Clustering, BIRCH, and Agglomerative Clustering) or algorithm specific like damping (Affinity Propagation), branching factor (BIRCH) or neighborhood size (DBSCAN). To understand these specifics, it would be helpful for the data analyst to investigate the different outcomes and results by varying the parameters and algorithms.

3 Concept

This section explains our concept, encompassing the analysis of different algorithms and their parameters (see section 3.1) and the comparison of two selected algorithms on a split surface (see section 3.2).

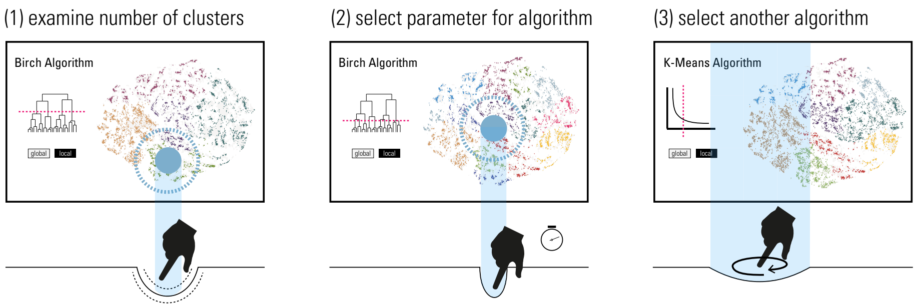

Figure 2: Layer concept with different interactions: exploring the number of clusters by pushing into the flexible display (1), number of clusters can be selected using hold gesture (2), another cluster algorithm chosen with a swirl gesture (3)

3.1 Layer Concept

Our concept focuses on the examination of different cluster algorithms and its parameters. Figure 1 shows the BIRCH (*balanced iterative reducing and clustering using hierarchies*) algorithm with varying number of clusters that are represented by colors. The goal is to determine a number of clusters that groups the current data set in the most meaningful manner. Differences between parameters and the affected items in these clusters are difficult to identify by representing all different versions side by side. In animated sequences, the change can be perceived better, but the impact on single data items is difficult to comprehend, because these animations always affect the whole data set. We introduce a layer concept on a flexible display that allows to examine different parts of the data set by stacking all versions of one algorithm (cp. Figure 1). These different versions can be explored by a depth interaction on the flexible display, which allows the comparison of the impact of different parameters for this algorithm on a group of data items (*local*) or on the whole data set (*global*).

The concept is shown in Figure 2. The user can choose between different cluster algorithms by using a swirl gesture, which can be performed by a circular movement on the elastic display (see Figure 2 – (3)). Once an algorithm is chosen, the user can examine the impact of different parameters such as the number of clusters (e.g. from 5 to 20 clusters, represented by different colors) or the neighborhood size (in case of the DBSCAN algorithm) by pushing into the flexible display. The presented layer depends on the depth of the interaction (see Figure 2 – (1) and Figure 4, left). Next to the cluster, a second visualization provides insights about the result of the algorithm manipulation. Both visualizations are connected by linking and brushing. The selection of the presented visualization depends on the chosen algorithm (e.g. for Agglomerative Clustering, the dendrogram visualizes the tree structure of the hierarchical accumulation; for K-Means, the elbow diagram shows the wcss (*within-cluster sums of squares*) score in connection to the number of clusters and indicates the status through a highlighted marker in which level or position the current depth interaction is leading (Figure 2, pink line on the left side). Furthermore, two buttons on the left side of the interface allow to change the influence of the push gesture. It can influence the whole data set (*global*) or just the current group of data items that is selected by the push gesture (*local*). As soon as the user decides for a specific value

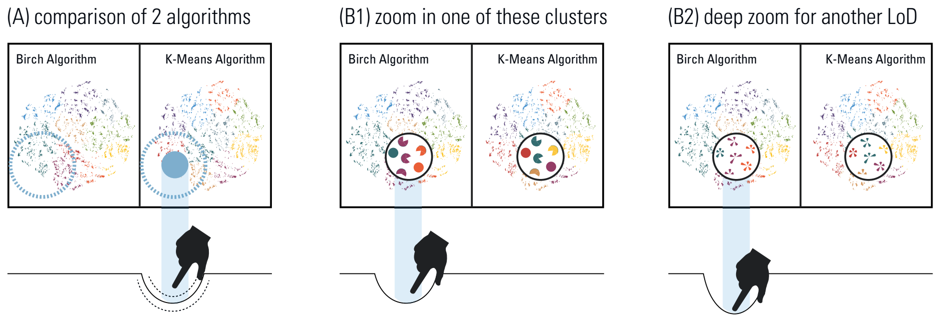

Figure 3: Interface is split into two regions to compare two algorithms with each other: A – number of clusters can be explored with two algorithms (left), B – push gesture creates a magic lens, which shows the underlying data in a different level of detail (LoD): verification of empty values in the second level (middle) and verification of different features in the third level (right)

of the parameter, e.g. the number of clusters, he or she can select these parameters by a hold gesture in this depth position and save them for further examination (see section 3.2). Again, another algorithm can be selected by using the swirl gesture (see Figure 2 – (3)), shown by a wcss diagram from the elbow method for the analyzation of the number of clusters in the K-Means algorithm.

3.2 Comparison Concept

The comparison of two different algorithms is afforded by using a split display. We explain this comparison concept in combination with the depth interaction, which presents different layers to the user as well as different levels of detail.

3.2.1 Comparison of Cluster Algorithms with Layer Concept

In order to find the most suitable clustering of the data, it is useful to compare different algorithms and to fine-tune the result. This is the main purpose of the comparison concept. Once the desired number of clusters has been defined using the layer concept (cf. 3.1), the clustering result can be compared with the outcome of another algorithm in a side-by-side view. Each side shows the result of the clustering with n-2 clusters (with n being the result from the previous step). By pushing into the surface on one side, the number of clusters can be increased (see Figure 3 – A), similar to the interaction in the first concept, but with more control over the specification of clusters, as the number of clusters only increases four steps over the whole interaction depth at most. Therefore, the user can compare the outcome of the algorithm in the range of [n-2, n+2] clusters on each side. To compare the clustering of algorithms, the user can apply the same pressure on both sides, or apply different pressure on each side to make a detailed comparison of clustering results with slightly different parameterizations. When pulling the surface, the user can navigate back to Step 1 to adjust the cluster parameterization on a large scale, e.g. if the comparison of another algorithm shows that the selected parameter was unsuitable.

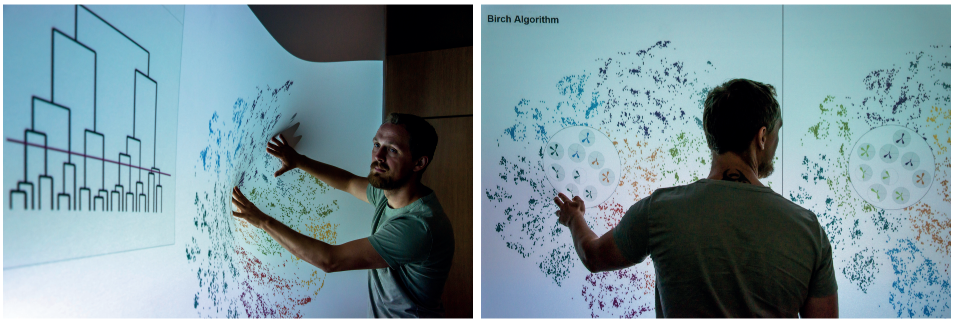

Figure 4: Concept tested with elastic display: Layer concept (left), Comparison concept with semantic zoom (right)

3.2.2 Comparison of Clusters with Semantic Zoom

The next step in the concept allows the exploration of both selected cluster algorithms by providing a zoomable user interface. By pushing into the surface, a magic lens appears, which presents the selected items in another level of detail (LoD). This is done in order to support the user analyzing the individual features of the dataset that are responsible for their position. The magic lens is located next to the pressure point to avoid occlusion by the user's hand and also appears at the same location in the second cluster visualization. This allows a comparison of the containing glyphs and their colors (see Figure 3, B1 and B2 and Figure 4, right). Different levels of detail are provided, depending on the interaction depth: (1) the first level represents each data item as a colored pixel; (2) the second level appears in low interaction depth and present these pixels as pie glyphs, indicating empty values in the feature vector of the corresponding data item (see Figure 3, B1); (3) the third level represents all data items as flower glyphs (Keck et al., 2017), where each petal represents one feature of the underlying data item and its length the corresponding value (see Figure 3, B2).

4 Conclusions and Future Work

In this contribution, we discussed concepts that leverage the power of elastic displays in the domain of Big Data clustering. We showed how different algorithm parameters can be explored and modified. Many visualization options can be further investigated such as using inertia to return to the initial state after a zooming interaction. Moreover, further interaction possibilities should be made available, for instance to correct cluster algorithms by editing ground truth data. Multi-user scenarios and user identification should be addressed as well. After the exploration of clusters and algorithms, an interface needs to be specified in order to feed the results back to the system.

Acknowledgements. On behalf of Dietrich Kammer, Mandy Keck and Thomas Gründer, this research (project no. 100238473) has been funded by the European Regional Development Fund and the Free State of Saxony. On behalf of Mathias Müller, this research (project no. 100271686) has been funded by the European Social Fund and the Free State of Saxony.

References

Agarawala, A. & Balakrishnan, R. (2006). Keepin' it real: Pushing the desktop metaphor with physics, piles and the pen. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 1283–1292). CHI '06. Montreal, Quebec, Canada: ACM. doi:10.1145/1124772.1124965

Arthur, D. & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual acm-siam symposium on discrete algorithms* (pp. 1027–1035). SODA '07. New Orleans: Society for Industrial and Applied Mathematics.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining* (pp. 226–231). KDD'96. Portland, Oregon: AAAI Press.

Frey, B. J. & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, *315*(5814), 972–976. doi:10.1126/science.1136800

Gründer, T., Kammer, D., Brade, M., & Groh, R. (2013). Towards a design space for elastic displays. In *Acm sigchi conference on human factors in computing systems - workshop: Displays take new shape: An agenda for future interactive surfaces*. Paris - France.

Jacob, R. J., Girouard, A., Hirshfield, L. M., Horn, M. S., Shaer, O., Solovey, E. T., & Zigelbaum, J. (2008). Reality-based interaction: A framework for post-wimp interfaces. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 201–210). CHI '08. Florence, Italy: ACM. doi:10.1145/1357054.1357089

Keck, M., Kammer, D., Gründer, T., Thom, T., Kleinsteuber, M., Maasch, A., & Groh, R. (2017). Towards glyph-based visualizations for big data clustering. In *The symposium on visual information communication and interaction - vinci 2017*. Bangkok, Thailand (in press).

Müller, M., Gründer, T., & Groh, R. (2015). Data exploration on elastic displays using physical metaphors. In *Proceedings xcoax 2015, 25./26. juni, glasgow, schottland*.

Müller, M., Knöfel, A., Gründer, T., Franke, I. S., & Groh, R. (2014). Flexiwall: Exploring layered data with elastic displays. In *Proceedings its 2014, november 16.-19., germany*.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems 14* (pp. 849–856). MIT Press.

Peschke, J., Göbel, F., Gründer, T., Keck, M., Kammer, D., & Groh, R. (2012). Depthtouch: An elastic surface for tangible computing. In *Proceedings of the international working conference on advanced visual interfaces* (pp. 770–771). AVI '12. Capri Island, Italy: ACM. doi:10.1145/2254556.2254706

Sibson, R. (1973). Slink: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, *16*(1), 30. doi:10.1093/comjnl/16.1.30

Troiano, G. M., Pedersen, E. W., & Hornbæk, K. (2014). User-defined gestures for elastic, deformable displays. In *Proceedings of the 2014 international working conference on advanced visual interfaces* (pp. 1–8). AVI '14. Como, Italy: ACM. doi:10.1145/2598153.2598184

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). Birch: An efficient data clustering method for very large databases. *SIGMOD Rec. 25*(2), 103–114. doi:10.1145/235968.233324