Supplementary Material

INTRODUCTION

Here we supplement our paper [1] with results of a pilot user study, an additional performance analysis, and a further discussion of the framework interactions. We first present an evaluation of our framework's visual interactions through a user study with twelve data scientists. Next we provide additional performance analysis results for forward and backward projections in principal component analysis (PCA). We then discuss how our framework can extend beyond model-based analysis and provide examples of forward and backward projection interactions in everyday visualizations, examining them under the visual embedding model [2].

USER STUDY

We evaluate the user experience with our techniques through a user study with twelve data scientists. We have two goals. The first is to assess the effectiveness of our projection and visualization techniques in what-if analysis of dimensionally reduced data, the second is to understand how the use of the techniques differs for changing task type and complexity.

Participants: We recruited twelve participants with at least two years' experience in data science. Their areas of expertise included healthcare analytics, genomics and machine learning. Participants ranged from 25 to 55 years old, and all had at least a master's degree in science or engineering. Ten participants regularly applied dimensionality reductions in their data analysis, using Matlab, R and Python. All participants were familiar with using PCA, while six had used multidimensional scaling (MDS) before. Four participants cited additional projection methods that they had previously used, including t-SNE [5] and autoencoder [7, 3].

Tasks and Data: Participants were asked to perform the following six high-level tasks using our tool Praxis. Our tabular dataset [6, 8] containing eight socioeconomic development indices for thirty-four countries belonging to the Organization for Economic Co-operation and Development (OECD). The dataset was a CSV file with 34 rows and nine columns, where one column contained country names. Participants were free to use any combination of interactions and visualizations to complete a given task.

T1: What four development indices contribute the most to determining the position of points in the projection plane? Can you rank them based on their relevance?

T2: Can you explain why Portugal is in its specific position of the projection plane, distant from the other European countries?

T3: Suppose Chile has a near-term plan to attain a development level similar to Greece but can increase spending in only one of the development index areas. On which area would you advise the Chilean government to focus its resources?

T4: Consider the cluster formed by Turkey and Mexico. Compare it to the cluster formed by Asian countries.

T5: Suppose Israel cannot increase its education spending for the foreseeable future due to budgetary constraints. Is that country likely to attain a development level similar to that of Canada?

T6: Given that the Italian government would not allow WORKINGLONGHOURS to increase beyond the distribution mean, say which countries Italy could be considered similar to if Italy was able to improve its STUDENTSKILLS index value to 500.

Procedure: The study took place in the experimenter's office; one user at a time used Praxis running on the experimenter's laptop. The study had three stages. In the first, participants were briefed about the experiment and filled out a pre-experiment questionnaire eliciting information about their experience in data science and use of dimensionality-reduction techniques. In the second stage, participants were introduced to the Praxis interface and to our techniques via a training dataset. Five minutes were dedicated to showing each user how to perform forward projection and backward projection along with interactively adding constraints. Participants were then introduced to a new dataset and asked to complete the six tasks above. Task duration was manually timed and subject responses were collected through think-aloud protocol. Participants had at most two trials to perform each task; in the event of a failure, they moved on to the next task. For each task we also recorded whether the task was completed with or without the experimenter's help. In the last stage, participants were asked to complete a post-experiment questionnaire to gather subjective feedback.

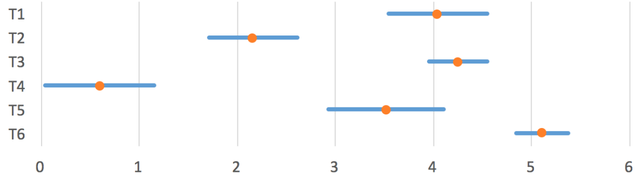


Figure 1: Task completion time. Average log time for participants to complete the six assigned tasks.

Results and Discussion: We adopt a task performance criterion similar to that in [8]. For each task, we count the number of participants who completed the task (C), completed the task with help (H), or were unable to complete the task (I). Similarly, we also report the frequency of the interaction (forward projection and backward projection) and visualization (prolines and feasibility map) techniques used by users to complete each task. We list these results in Table 1. Figure 1 shows the average log time spent on each task, inclusive of the

	performance			techniques used			
Task	C	H	I	*forward*	prolines	*backward*	*feasibility*
T1	10	2	0	2	12	0	0
T2	12	0	0	0	12	0	0
T3	11	1	0	8	9	4	0
T4	12	0	0	1	12	2	0
T5	11	1	0	2	12	11	0
T6	9	2	1	2	2	2	10

Table 1: Results of user study. For all twelve participants, the table indicates for each task how many of them completed the task with no help (C), completed it with help (H), or did not complete the task (I). We also show the number of participants who used each of our proposed techniques to perform single tasks. Note that both prolines and feasibility map can give users visual information without requiring them to perform forward projection or backward projection (whereas, for instance, forward projection is intrinsically bound to prolines).

cases in which the participant didn't complete the task. All the completed tasks were performed in less than 30 seconds. Task completion times and their standard errors reflect the intrinsic complexities of the tasks.

Overall, prolines proved to be a simple yet powerful visualization technique for exploring dimensionally reduced data as well as reasoning about the dimensionality reduction. Prolines were used 59 times by participants over the course of the six tasks performed. One participant mentioned he particularly liked "how prolines generate meaningful axes in a scatter plot where a clear mapping to data dimensions is unclear," and another described prolines as a "great way to understand dimensionality reductions, especially for people who used to treat them as a black box." The second most frequently used technique was backward projection, (used 19 times), followed by forward projection (15 times). Note that the use of forward projection always involves displaying prolines, which incorporates paths of forward projection locations for hypothetical feature values. Participants used forward projection when they wanted to interactively change the feature values and see precisely the projection change of the corresponding data point. In particular, one participant declared, "I feel backward projection is more natural to use and useful to see which features correlate to each other, but I would prefer forward projection for more precise control over feature values." Also, despite its lower incidence of use, feasibility map was employed by the participants when the task was judged sufficiently complex (T6).

MODEL ANALYSIS

Since the proposed techniques are intended for use in interactive applications, their computational complexity needs to adhere to certain responsiveness requirements. At the same time, forward and backward projection methods need to be accurate enough at estimating changes in the dimensionally reduced space as well as in the underlying multidimensional data so as not to lead the user to false assumptions. We evaluate our proposed techniques for PCA in terms of time and accuracy over varying number of samples and dimensions of the input dataset and also over the amount of change introduced by the user (i.e. how much a feature value is modified in the case of forward projection, how much a data point is moved in the case of backward projection).

In our evaluation we iteratively perform forward projection and unconstrained backward projection on automatically generated Gaussian random multivariate distributions, changing either the number of data samples or the number of data dimensions and leaving the other one fixed. We apply our techniques for each data point of the original distribution. The forward projection algorithm is then applied to each dimension with an amount of change in $\{\sigma_i/8, \sigma_i/4, \sigma_i/2, \sigma_i\}$, where σ_i is the standard deviation for the current feature. Backward projection is performed in eight possible directions of movement (horizontal and vertical axes plus diagonals), with an amount of change in $\{m/80, m/40, m/20, m/10\}$, where m is the width of the projection plane. Accuracy and time performances are determined for each execution of the two techniques and then averaged over all dimensions (directions), data samples and test iterations. All experimental results presented were generated on a MacBook Pro, 2015 edition.

Time Performance: Figure 2 shows that the execution time for both forward projection (e,f) and backward projection (g,h) is on the order of microseconds and is influenced neither by the number of samples nor by the amount of change; charts i and h show a linear dependence on the number of dimensions that does not, however, significantly affect the time performance. Even when dealing with larger datasets (e.g., more than 500 samples, greater than 100 dimensions), both techniques are suitable for interactive data analysis tools. Figure 2 also shows the time performance of prolines (i,j) and feasibility map (k,l), respectively assuming the computation of each proline with an average resolution of 5 forward projection samples and the generation of the feasibility map with a resolution of 100 backward projection samples. In particular, we note that the time to compute prolines depends linearly on the number of dimensions.

Accuracy: To assess the accuracy of our techniques, we introduce a new similarity criterion for data-point neighborhoods in dimensionally reduced spaces. For each execution of the algorithms on a data point, we compute two sets of neighbors: 1) the k closest neighbors in the projection plane after performing forward projection or after moving the data point with backward projection, and 2) the k closest neighbors in the projection plane after performing the dimensionality reduction on the multidimensional data, after it has been modified through forward projection or by backward projection. Optimally, these two neighborhoods should contain the same elements, which should have the same relative distance from the data point on which the technique is performed. We define a neighborhood correlation index $c_k = c_e \times c_o$, where c_e is the percentage of elements that appear in both neighborhoods, whereas c_o is the percentage of elements whose distance from the data point considered remains in the same order. The index varies between 0 and 1, with 1 corresponding to very similar neighborhoods. Figure 2 shows that the accuracy of

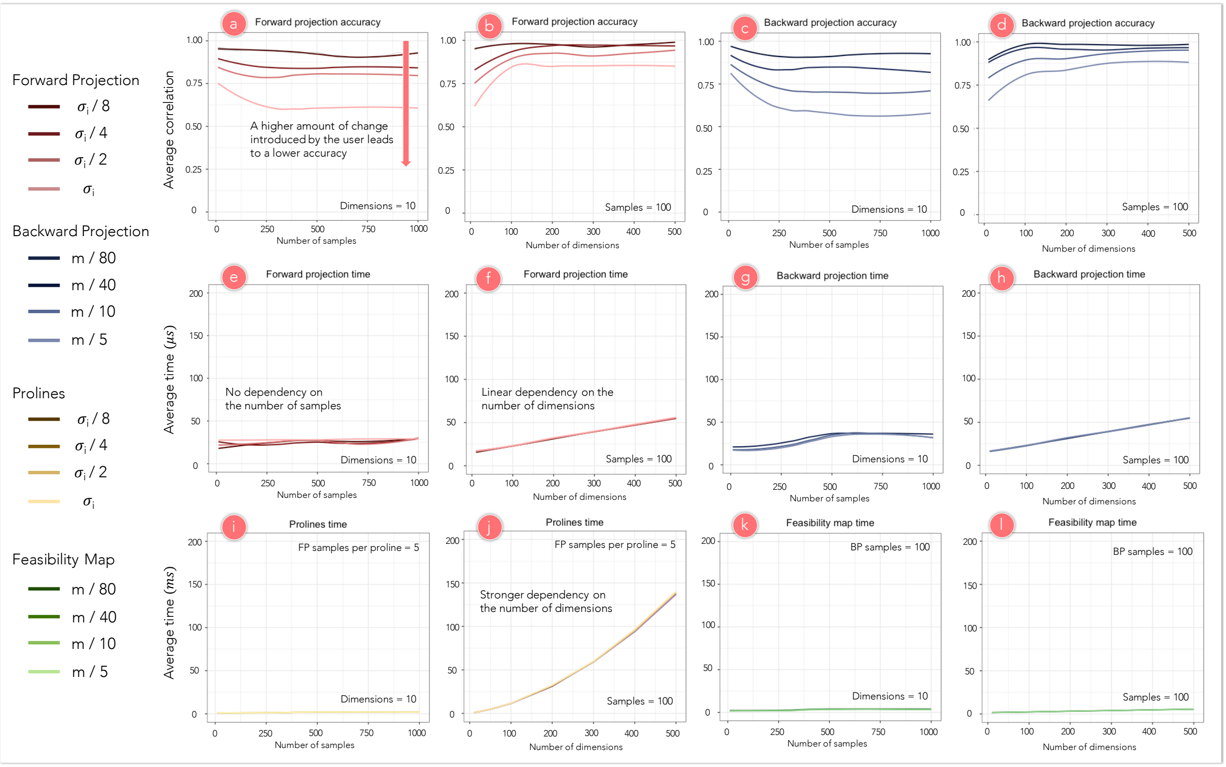


Figure 2: Accuracy and time performance results for PCA. We note that the accuracy and time performance of forward projection and backward projection are mostly insensitive to the number of samples and dimensions. Accuracy is instead tied to the amount of change introduced by the user. The computational time for generating prolines shows a linear dependence on the number of dimensions. Note that time is displayed in microseconds in charts (e-h) and in milliseconds in charts (i-l).

both forward projection (a,b) and backward projection (c,d) is mostly insensitive to the number of samples or dimensions. Instead, we notice a strong dependence on the amount of change introduced by the user. This shows that our proposed techniques are well suited for local data changes, and that greater user modifications could possibly alter the properties of the dimensionality reduction.

A MODEL FOR DYNAMIC VISUALIZATION INTERACTIONS

We look at our framework from the perspective of interaction design. The interaction techniques we introduce belong to a class of interactions that tightly couples data with its visual representation: when users interactively change one, they can observe a corresponding change in the other. For example, through forward projection, users observe how the visual representation (2D position) changes as they change the value of a dataset's attributes. Conversely, users can see how the data changes through backward projection as they change the visual representation. This class of interactions is essential for realizing dynamic visualizations (e.g. [9, 4]); we call them *dynamic visualization interactions* or *dynamic interactions* for short.

We now discuss dynamic interactions using the visual embedding model [2]. The visual embedding model provides a functional view of data visualization and posits that a good visualization is a structure- or relation-preserving mapping from the data domain to the range (co-domain) of visual encoding variables (Figure 3). Visual embedding gives us criteria to evaluate the effectiveness of dynamic interactions (Figure 4): 1) a change in data (e.g., induced by the user through direct manipulation) should cause a proportional change in its visual representation and 2) a perceptual change in a visual encoding value (e.g., by dragging nodes in a scatter plot or changing the height of a bar in a bar chart) should be reflected by a change in data value that is proportional to the perceived change. However, in order to enable a dynamic interaction on a visualization, we need access to both the visualization function f and its inverse f^{-1}. The visual embedding model

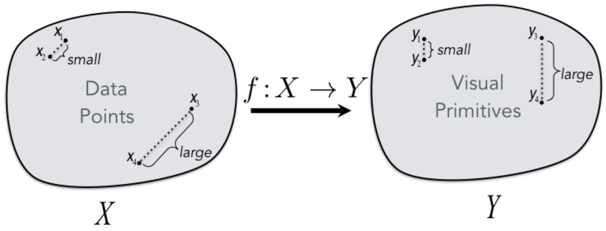


Figure 3: Visual embedding is a function that preserves structures in the data domain X within the embedded perceptual space Y (adapted from [2]).

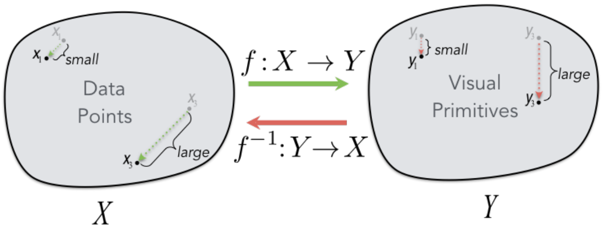
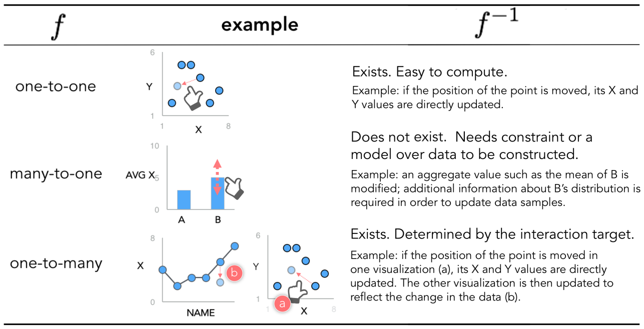


Figure 4: Bidirectionally coupling data and its visual representations. Visual embedding suggests that a change in the data should be reflected by a proportional change in its visual representation using f, the visual embedding function. Conversely, a change in a visual representation should be reflected by a proportional change in the corresponding data using f^{-1}..

also suggests why implementing back mapping to the data space can be challenging.

We consider three basic forms of the visualization function f in Figure 5 through examples using a toy dataset in Figure 6. When the visualization function f is one-to-one, a dynamic interaction over f is straightforward, as f^{-1} exists. When f is one-to-many (still invertible but not necessarily a proper function), f^{-1} exists and is determined by the target of interaction. Consider the example in Figure 5. We visualize how X values change for each NAME with a line chart. We also visualize the correlation of X and Y with a scatter plot. If a user moves a point up or down in the line chart, the corresponding change in X can be easily computed and the Scatter plot can be updated in a brush-and-link fashion. Essentially, the one-to-many case can be seen as a collection of multiple one-to-one visualizations.

The most interesting case is when f is many-to-one and hence not invertible. A frequent source of such visualization functions is summary data aggregations, which are lossy. Note that dimensionality reduction is a form of lossy data aggregation, where structures or relations in high dimension are approximately aggregated into fewer dimensions. A simple example of a many-to-one visualization is the bar chart in Figure 5 that shows the mean X for the data points grouped by TYPE, A and B. Now, in a dynamic interaction scenario, how should we update the data values if a user changes the heights of the bars? Our backward projection solves a similar problem under a more complex visualization function, dimensionality reduction. In general, constructing a dynamic interaction over many-to-one visualization functions would require imposing a set of assumptions over data in the form of, e.g., constraints or models. This presents a challenging yet important future research direction.



Exists. Easy to compute.

Example: if the position of the point is moved, its X and Y values are directly updated.

Does not exist. Needs constraint or a model over data to be constructed.

Example: an aggregate value such as the mean of B is modified; additional information about B's distribution is required in order to update data samples.

Exists. Determined by the interaction target.

Example: if the position of the point is moved in one visualization (a), its X and Y values are directly updated. The other visualization is then updated to reflect the change in the data (b).

Figure 5: Visualization classes. Three basic forms of the visualization function f. Implementing a dynamic interaction is clearly challenging when f^{-1} does not exist.

NAME	TYPE	X	Y
id0	A	4	5
id1	A	2	2
id2	A	7	3
id3	B	3	5
id4	B	5	4
id5	B	6	2

Figure 6: Toy tabular dataset used for the three examples in Figure 5.

REFERENCES

1. Marco Cavallo and Çağatay Demiralp. 2018. A Visual Interaction Framework for Dimensionality Reduction Based Data Exploration. In *ACM Human Factors in Computing Systems (CHI)*.
2. Çağatay Demiralp, Carlos Scheidegger, Gordon Kindlmann, David Laidlaw, and Jeffrey Heer. 2014. Visual Embedding: A Model for Visualization. *Computer Graphics and Applications* (2014).
3. Geoffrey E. Hinton and Ruslan R. Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313, 5786 (2006), 504–507.
4. Brittany Kondo and Christopher Collins. 2014. Dimpvis: Exploring Time-varying Information Visualizations by Direct Manipulation. *IEEE TVCG* 20, 12 (2014), 2003–2012.
5. Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data Using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605.
6. OECD Better Life Index 2016. `http://www.oecdbetterlifeindex.org/`. (2016). Accessed: Dec 24th, 2017.
7. David E. Rumelhart, Geoffrey. E. Hinton, and Ronald. J. Williams. 1986. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, David E. Rumelhart and James L. McClelland (Eds.). Chapter Learning Internal Representations by Error Propagation, 318–362.
8. Julian Stahnke, Marian Dörk, Boris Müller, and Andreas Thom. 2016. Probing Projections: Interaction Techniques for Interpreting Arrangements and Errors of Dimensionality Reductions. *IEEE TVCG* 22, 1 (2016), 629–638.
9. Bret Victor. 2013. Media for Thinking the Unthinkable. `https://vimeo.com/67076984`. (2013). Accessed: Dec 24th, 2017.