Eurographics Conference on Visualization (EuroVis), Posters Track (2017)
A. Puig Puig and T. Isenberg (Editors)

Projection Navigation In Extremely Large Datasets (PNIELD)

J. F. Kruiger[1,2], A. C. Telea[1], C. Hurter[2]

[1]University of Groningen, the Netherlands
[2]École Nationale de l'Aviation Civile, France

1. Introduction

Multidimensional projections (MPs) visualize high-dimensional data by mapping a set $X = \{\mathbf{x}_i\} \subset \mathbb{R}^n$ of such observations to a lower-dimensional space. Formally put, a projection P is a function

$$P : \mathbb{R}^n \to \mathbb{R}^m, \quad m \ll n.$$

If $m = 2$, we can represent the projected data by a traditional scatterplot. Many MP methods exist, offering various trade-offs between ease of use (automation), accuracy of representing n-dimensional distances [PVG*17] or neighbourhoods [vdMH08], computational scalability [JCC*11], and robustness with respect to small changes in the data [RFT16]. For a very large number of observations $N = |X|$ and a large number of dimensions n, computing a *single* high-accuracy projection $P(X)$ of the entire dataset X becomes either too expensive or creates too large inaccuracies. In the limit, very large N values make even the rendering of $P(X)$ hard to follow, due to clutter. Such problems are partially solved by so-called *landmark* methods, such as LAMP [JCC*11], LSP [PNML08], or LandmarkMDS [DST03]. These methods select a small subset $X_l \subset X$ of so-called landmarks, representatives, control points, or anchors. Next, X_l is projected to $Y_l \subset \mathbb{R}^m$ using a—typically high-accuracy—method P or manual placement [JCC*11], and the projections of remaining observations $X \setminus X_l$ are arranged around points in Y_l based on a local low-cost stress minimization principle. Landmark MPs can thus be described by

$$\hat{P} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m,$$
$$\hat{P}(X, X_l, P(X_l)) = Y.$$

While faster than classical methods, landmark MPs cannot directly handle very large datasets X: A *single* subsampling X_l may not be enough, as this yields either too many landmarks for the expensive landmark-projection P to work quickly, or too few landmarks in which case P has a large error. Also, it is not evident how to control the level-of-detail in Y so as to emphasize specific data patterns with controlled error.

We propose a framework for the exploration of large high-dimensional datasets via MPs that addresses the above challenges, with the following key contributions C_i:

Scalability (C_1): We handle large datasets X in time linear to $|X|$.
Level-of-detail (C_2): We propose a multiscale view on P which ranges between overviews of the full X (with higher errors) and detailed views on subsets of X (with lower errors).
Continuity (C_3): Navigation between our multiscale levels is continuous in the projection space $\mathbb{R}^2$. This helps users maintaining their mental map.
Control (C_4): For navigation, we extend classical 2D zoom-and-pan, familiar to most users, to handle $\mathbb{R}^n$ space. Intuitively put, we allow exploring a high-dimensional space via a 'Google Earth' metaphor of navigating point clouds, where more details—*i.e.*, more points—are automatically added, on-demand.

2. Method

Our method can be compactly described in terms of three operations—subsampling, projection, and exploration—as follows.

Subsampling: We handle very large input datasets X by subsampling these by an operator $S^M : \mathbb{R}^n \to \mathbb{R}^n$, $S^M(X) \subset X$. Subsampling allows us to construct a smaller dataset $\left|S^M(X)\right| = M \ll |X|$ which we can next project by landmark MPs (Sec. 1). Simple subsampling methods that are linear in $|X|$ include random sampling [Vit85, Knu81], which we denote as S^M_{RND}.

Projection: With $X_v = S^{M_v}_{RND}(X)$ computed as above, we project X_v by LAMP [JCC*11], with metric MDS [PVG*17] used for accurate projection of $X_l \subset X_v$, where landmarks X_l are selected by *further* subsampling X_v. In detail, we define

$$\begin{aligned} X_v &= S^{M_v}_{RND}(X), \\ X_l &= S^{M_l}_{RND}(X_v), \\ Y_l &= P_{MDS}(X_l), \\ Y_v &= \hat{P}_{LAMP}(X_v, X_l, Y_l), \end{aligned} \tag{1}$$

that is, we subsample X to $M_v = 1000$ observations, of which we next select $M_l = 50$ landmarks to project via MDS, and using this, construct the projection Y_v of X_v using LAMP.

Exploration: Our method's main strength becomes apparent when we consider interactive exploration. Applying Eq. (1) to our whole input data X yields an *overview* scatterplot Y_v which shows the general structure of X. However, we do not have *details*, since X_v is a coarse subsampling of X. We next enable interactive level-of-detail exploration of the data by *multiscale projections* (see also Fig. 2): The user selects a focus point $\mathbf{y} \in \mathbb{R}^2$, *e.g.*, at the mouse location. We next select all observations $X_k \subset X_v$ whose projections in Y_v are the k-nearest neighbours of $\mathbf{y}$ *in the 2D space*, where k defines the zoom level - *e.g.*, setting k to 90% of M_v yields a zoom of roughly 10%. Points outside X_k are discarded. There is now room for $M_v - k$

DOI: 10.2312/eurp.20171181

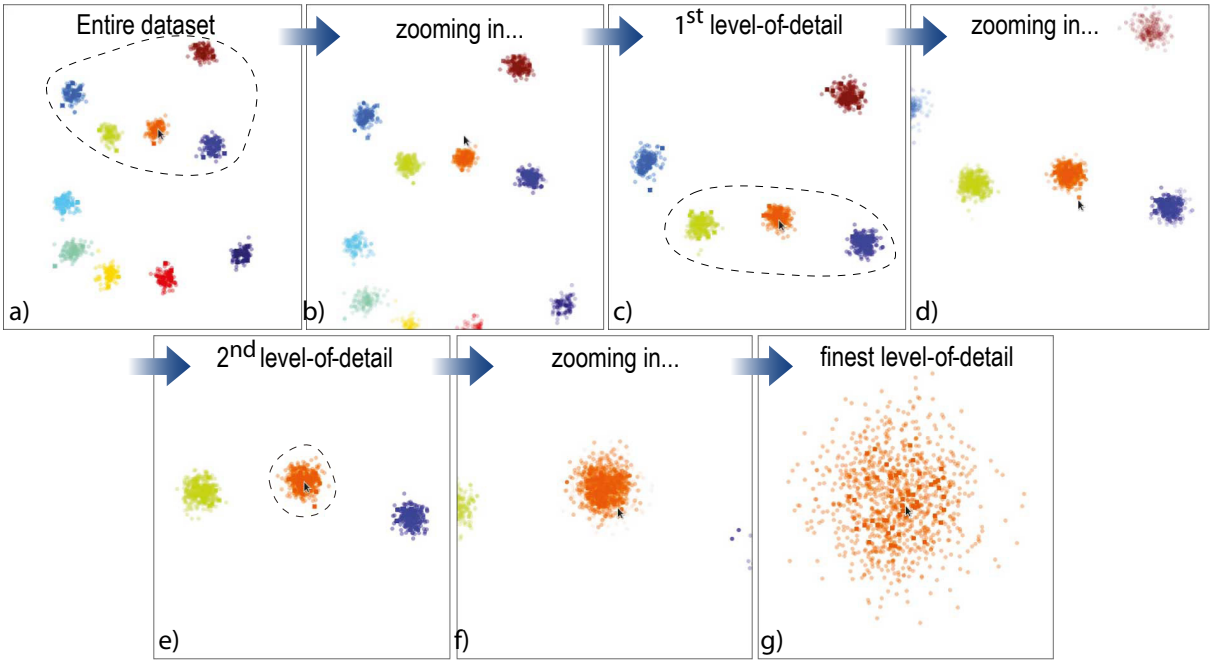

Figure 1: *Multiscale projection exploration. From the overview of X (a), we zoom three times to get details in the orange cluster, yielding views (c), (e), and (g). As we zoom, points are added on-demand—(g) has about $M_v = 1000$ orange points as compared to only 100 in (a). Images (b), (d), and (f) show intermediate interpolation stages during the zooming. Dashed lines show the regions of interest (ROIs) Y_k.*

more points, so we compute the set X_c of $M_v - k$ observations from $X \setminus X_k$ that are closest to X_k. Next, we define the new set of observations $X'_v = X_k \cup X_c$, and project it using as landmarks X'_l a set of M_l randomly chosen points from X_k, *i.e.*, $X'_l = S^{M_l}_{RND}(X_k)$. The projection Y'_l of the landmarks is not re-computed, to preserve visual continuity, but is set to the points from Y_k that map the observations in X'_l. The new set of landmarks yields a new projection $Y'_v = \hat{P}_{LAMP}(X'_v, X'_l, Y'_l)$, analogous to Eq. (1). Finally, we interpolate between the current scatterplot Y_v and the new one Y'_v by linearly interpolating the positions of the points common to the two plots and also fading out points that exist in Y_v (but not in Y'_v) and fading in points that exist in Y'_v (but not in Y_v). This ensures a smooth transition during zooming (see also the additional material).

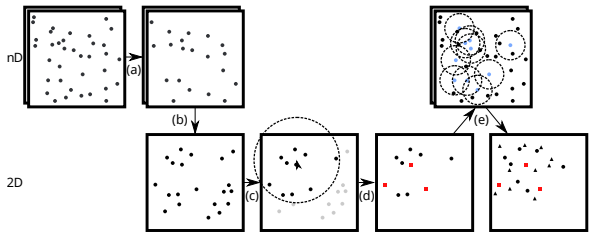

Figure 2: *Multiscale projection exploration. a) Subsampling the dataset $X \subset \mathbb{R}^n$. b) Projecting $S(X)$ to 2D. c) User selects ROI in 2D. d) Landmarks are sampled from ROI points. e) $\mathbb{R}^n$ observations are selected as nearest-neighbors of observations mapped to ROI points. Newly selected points are projected with the other remaining points using landmarks from (d).*

Results: Our method has several key advantages *vs.* state-of-the-art MP methods. Following Sec. 1, these are as follows. (C_1): We can rapidly project datasets of any size by controlling the parameters M_v and M_l (Eq. (1)); this gives a trade-off between speed and level-of-detail. Since we use subsampling, and LAMP scales well in M_l and M_v, our method is real-time for datasets with millions of observations. (C_2): We can smoothly navigate between coarse views of large datasets X and detailed views of subsets X_k of such datasets. (C_3): We ensure continuity during navigation, by the *consistent* use of landmarks X_l during zooming (Sec. 2), and by the linear interpolation of the scatterplot positions. (C_4): Navigating $\mathbb{R}^n$ data spaces is simple—just use classical point-and-zoom 2D tools. This is the first time, to our knowledge, that this mechanism has been used for the navigation of $\mathbb{R}^n$ spaces. Simply put: our proposal lets users zoom in/out in $\mathbb{R}^n$ datasets as easily, and intuitively, as when doing it in 2D space. We coded the proposed framework in *Python 3* using *SciPy* [JOP*17]. Our implementation can easily handle datasets of over a million observations with real-time zoom exploration.

3. Acknowledgements

This work was partly supported by the project MOTO (H2020-SESAR-2015-1), grant 699379, offered by the European Commission.

References

[DST03] De Silva V., Tenenbaum J. B.: Global *vs* local methods in nonlinear dimensionality reduction. *Adv Neur Inf Process Syst* (2003), 721–728. 1

[JCC*11] Joia P., Coimbra D., Cuminato J. A., Paulovich F. V., Nonato L. G.: Local affine multidimensional projection. *IEEE TVCG 17*, 12 (2011), 2563–2571. 1

[JOP*17] Jones E., Oliphant T., Peterson P., et al.: SciPy: Open source scientific tools for Python. `scipy.org`, 2017. 2

[Knu81] Knuth D. E.: *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, 2 ed. 1981. 1

[PNML08] PAULOVICH F. V., NONATO L. G., MINGHIM R., LEVKOWITZ H.: Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE TVCG 14*, 3 (2008), 564–575. 1

[PVG*17] PEDREGOSA F., VAROQUAUX G., GRAMFORT A., ET AL.: Scikit-learn: Machine learning in Python. scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS, 2017. 1

[RFT16] RAUBER P., FALCÃO A., TELEA A.: Visualizing time-dependent data using dynamic t-SNE. *Proc. EuroVis – Short papers* (2016). 1

[vdMH08] VAN DER MAATEN L. J. P., HINTON G. E.: Visualizing high-dimensional data using t-SNE. *JMLR 9* (2008), 2579–2605. 1

[Vit85] VITTER J. S.: Random sampling with a reservoir. *ACM TOMS 11*, 1 (1985), 37–57. 1