Interactive Visual Data Exploration with Subjective Feedback: An Information-Theoretic Approach

Kai Puolamäki [#,+], Emilia Oikarinen [#,+], Bo Kang [*], Jefrey Lijffijt [*], Tijl De Bie [*]

[#] *Department of Computer Science, Aalto University, Espoo, Finland*
[+] *Finnish Institute of Occupational Health, Helsinki, Finland*
[*] *Department of Electronics and Information Systems, IDLab, Ghent University, Ghent, Belgium*

***Abstract*—The exploration of high-dimensional real-valued data is one of the fundamental exploratory data analysis (EDA) tasks. Existing methods use predefined criteria for the representation of data. There is a lack of methods eliciting the user's knowledge from the data and showing patterns the user does not know yet. We provide a theoretical model where the user can input the patterns she has learned as knowledge. The background knowledge is used to find a MaxEnt distribution of the data, and the user is shown maximally informative projections in which the MaxEnt distribution and the data differ the most. We provide an interactive open source EDA system, study its performance, and present use cases on real data.**

I. Introduction

Ever since Tukey's pioneering work on *exploratory data analysis* (EDA) [1], effective exploration of data has remained an art as much as a science. Human analysts are remarkably skilled in spotting patterns and relations in adequately visualized data, but coming up with insightful visualizations is a task hard to formalize, let alone to automate.

Modern computational methods for dimensionality reduction, such as Projection Pursuit and manifold learning, allow one to spot complex relations from the data automatically and to present them visually. Their drawback is however that the criteria by which the views are found are defined by static objective functions. The resulting visualizations may or may not be informative for the user and task at hand. Often such visualizations show the most prominent features of the data, while the user might be interested in other subtler structures. It would therefore be of a great help if the user could efficiently tell the system what she already knows and the system could utilize this when deciding what to show the user next. Achieving this is the main objective of this paper.

To illustrate our idea, we use here a synthetic 3-dimensional dataset with 150 points with two clusters of 50 points and two of 25 points. The smaller clusters are partially overlapping in the third dimension. The computer maintains a distribution, called the *background distribution* modelling the belief state of the user. The visualizations we use are scatter plots of the data points after projection onto a 2-D subspace, and the system shows the user projections in which the data and the background distribution differ the most. Looking at the first two principal components, we can only observe three clusters with 50 points each (the black points in Fig. 1 left). In our interactive approach, the data analyst will learn not only that there are actually four clusters, but also that two of the clusters correspond to a single cluster in the first view of the data. In addition to showing the data in the scatterplot, we display a sample from the background distribution as gray points (and lines that connect the respective points, to give an indication of the displacement in the background distribution).

The user's interaction consists of informing the system about sets of data points perceived to form clusters in this scatter plot. The system takes this information into account by updating the background distribution accordingly. When the user has ascertained herself that the background distribution matches the data in the projection as she thinks it should, the system can be instructed to find another 2-D subspace to project the data onto. The new projection displayed is the one maximally insightful *considering the updated background distribution*. In Fig. 1 (right), the new projection reveals that one cluster from the previous view, in fact, splits into two.

To summarize, our contributions are as follows:

- A background distribution accounting for a user's knowledge fomalized as a constrained MaxEnt distribution.
- A principled way to obtain projections showing the maximal difference between the data and the background distribution for the PCA and ICA objectives.
- An interaction model by which the user can input what she has learned from the data, in terms of constraints.
- An experimental evaluation of the computational performance of the method and use cases on real data.
- A free open source application demonstrating the method.

This paper is a summary of a tech report [2]. Related work and technical details are discussed in detail in [2].

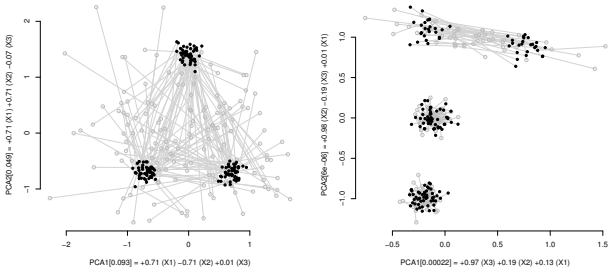

Fig. 1. Synthetic 3-D data. Left: Projection of the data onto the first two principal components together with a sample of background distribution; Right: The next most informative projection shown to the user.

II. METHODS

Preliminaries. A dataset consists of n d-dimensional real vectors $\hat{\mathbf{x}}_i \in \mathbb{R}^d$, where $i \in [n] = \{1, \ldots, n\}$. A matrix $\hat{\mathbf{X}} = (\hat{\mathbf{x}}_1 \hat{\mathbf{x}}_2 \ldots \hat{\mathbf{x}}_n)^T \in \mathbb{R}^{n \times d}$ represents the whole dataset. We use hatted variables (e.g., $\hat{\mathbf{X}}$) to denote the observed data and non-hatted variables the respective random variables (e.g., $\mathbf{X}$). We assume that the initial background distribution equals a spherical Gaussian distribution with zero mean and unit variance, given by $q(\mathbf{X}) \propto \exp\left(-\sum_{i=1}^n \mathbf{x}_i^T \mathbf{x}_i / 2\right)$.

Constraints. We can define constraints on subsets of points in $\mathbb{R}^{n \times d}$ for a given projection by introducing *linear and quadratic constraint functions* [3]. A constraint is parametrized by the subset of rows $I \subseteq [n]$ involved and a projection vector $\mathbf{w} \in \mathbb{R}^d$. The *linear constraint function* is defined by

$$f_{\text{lin}}(\mathbf{X}, I, \mathbf{w}) = \textstyle\sum_{i \in I} \mathbf{w}^T \mathbf{x}_i, \tag{1}$$

and the *quadratic constraint function* by

$$f_{\text{quad}}(\mathbf{X}, I, \mathbf{w}) = \textstyle\sum_{i \in I} \left(\mathbf{w}^T (\mathbf{x}_i - \hat{\mathbf{m}}_I)\right)^2, \tag{2}$$

where $\hat{\mathbf{m}}_I = \sum_{i \in I} \hat{\mathbf{x}}_i / |I|$. Notice that $\hat{\mathbf{m}}_I$ is not a random variable but a constant depending on the observed data.

The linear and quadratic constraint functions can be used to define several types of knowledge about the data. (i) *A margin constraint* consists of a linear and a quadratic constraint for each column in $[d]$. (ii) *A cluster constraint* encodes the mean and (co)variance statistics of a point cluster as follows: make a singular value decomposition (SVD) of the points in I, and define a linear and a quadratic constraint for each of the eigenvectors. (iii) *A 1-cluster constraint* is a special case of a cluster constraint with $I = [n]$. (iv) *A 2-D constraint* consists of a linear and a quadratic constraint for the two vectors spanning the current 2-D projection.

Background distribution. A triplet $C = (c, I, \mathbf{w})$, where $c \in \{\text{lin}, \text{quad}\}$ is a constraint, and the constraint function is then given by $f_c(\mathbf{X}, I, \mathbf{w})$. Our main problem is stated as follows.

Problem 1. *Given a dataset $\hat{\mathbf{X}}$ and k constraints $\mathcal{C} = \{C^1, \ldots, C^k\}$, find a probability density p over datasets $\mathbf{X} \in \mathbb{R}^{n \times d}$ such that the entropy defined by*

$$S = -E_{p(\mathbf{X})}\left[\log\left(p(\mathbf{X})/q(\mathbf{X})\right)\right] \tag{3}$$

is maximized, while satisfying

$$E_{p(\mathbf{X})}\left[f_{c^t}(\mathbf{X}, I^t, \mathbf{w}^t)\right] = \hat{v}^t, \tag{4}$$

for all $t \in [k]$, where $\hat{v}^t = f_{c^t}(\hat{\mathbf{X}}, I^t, \mathbf{w}^t)$.

The *background distribution* is the distribution p that is a solution to the Prob. 1. Intuitively, the background distribution is the maximally random distribution that preserves the constraints in expectation. The form of the solution to Prob. 1 is given by the following lemma.

Lemma 1. *A solution to Prob. 1 is of the form*

$$p(\mathbf{X}) \propto q(\mathbf{X}) \times \exp\left(\textstyle\sum_{t=1}^k \lambda^t f_{c^t}(\mathbf{X}, I^t, \mathbf{w}^t)\right), \tag{5}$$

where $\lambda^t \in \mathbb{R}$ are real-valued parameters.

See, e.g., Ch. 6 of [4] for a proof. For details of solving Prob. 1 numerically, we refer the reader to the extended version [2].

Whitening out the background distribution. Once we have found the distribution that solves Prob. 1, the next task is to find and visualize the maximal differences between the data and the background distribution defined by Eq. (3). To this end we sample a dataset from the background distribution, and produce a *whitened* version of the data. The direction-preserving whitening transformation of the data results in a unit Gaussian spherical distribution, if the data follows the current background distribution. Thus, any deviation from the unit sphere distribution is a signal of difference between the data and the current background distribution.

More specifically, we define new data vectors $\mathbf{y}_i = U_i D_i^{1/2} U_i^T (\mathbf{x}_i - \mathbf{m}_i)$, where the SVD decomposition of Σ_i^{-1} is given by $\Sigma_i^{-1} = U_i D_i U_i^T$, where U_i is an orthogonal matrix and D_i is a diagonal matrix. If we used one transformation matrix for the whole data, this would correspond to the normal whitening transformation. However, here we may have a different transformation for each of the rows. Furthermore, normally the transformation matrix would be computed from the data, but here we compute it from the constrained model.

PCA and ICA. To find directions where the whitened data looks different from the unit Gaussian distribution with zero mean, an obvious choice is to use Principal Component Analysis (PCA) and look for directions in which the variance differs most from unity. However, it may happen that the variance is already taken into account in the variance constraints, in which case PCA is not informative because all directions in whitened data have equal mean and variance. Instead, we can, e.g., use Independent Component Analysis (ICA) and the FastICA algorithm [5] with log-cosh G function as a default method to find non-Gaussian directions.

III. IMPLEMENTATION AND EXPERIMENTS

We have implemented an interactive demo system SIDER using R 3.4.0 with SHINY and FASTICA. SIDER runs in the web browser using R as a back-end, and is published as a free open source system under the MIT license at

TABLE I
MEDIAN WALL CLOCK RUNNING TIMES, BASED ON 10 RUNS FOR EACH SET OF PARAMETERS FOR FINDING THE CORRECT PARAMETERS (OPTIM) AND RUNNING THE ICA ALGORITHM (ICA) WITHOUT TIME CUTOFF.

		seconds, $k \in \{1, 2, 4, 8\}$	
n	d	OPTIM	ICA
2048	16	$\{0.0, 0.2, 0.3, 0.5\}$	$\{0.6, 0.6, 0.6, 0.6\}$
2048	32	$\{0.0, 0.6, 1.0, 2.1\}$	$\{1.5, 1.5, 1.6, 1.6\}$
2048	64	$\{0.1, 2.7, 5.2, 11.0\}$	$\{5.1, 5.2, 4.9, 4.9\}$
2048	128	$\{1.2, 21.4, 48.1, 124.6\}$	$\{17.8, 17.6, 17.4, 17.0\}$
4096	16	$\{0.0, 0.2, 0.3, 0.5\}$	$\{1.1, 1.1, 1.1, 1.1\}$
4096	32	$\{0.0, 0.6, 1.0, 2.0\}$	$\{3.1, 3.4, 3.0, 3.1\}$
4096	64	$\{0.2, 2.5, 6.0, 11.6\}$	$\{9.8, 9.3, 9.5, 9.6\}$
4096	128	$\{1.2, 23.4, 56.4, 121.3\}$	$\{34.2, 34.7, 34.4, 34.4\}$
8192	16	$\{0.0, 0.2, 0.3, 0.6\}$	$\{2.6, 2.2, 2.5, 2.1\}$
8192	32	$\{0.0, 0.6, 1.0, 2.0\}$	$\{6.5, 6.0, 5.9, 5.9\}$
8192	64	$\{0.2, 2.7, 6.0, 12.2\}$	$\{20.7, 20.4, 19.8, 20.1\}$
8192	128	$\{1.2, 21.9, 44.1, 110.3\}$	$\{67.9, 67.5, 67.1, 67.6\}$

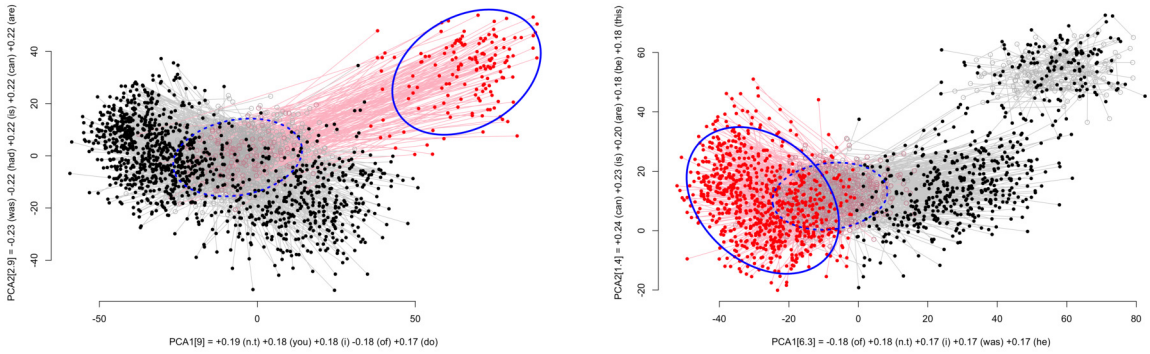

Fig. 2. A use case with the BNC data. Left: Selection of points for a cluster constraint in the first PCA projection; Right: Selection of points for the second cluster constraint. The view is the next most informative PCA projection obtained after adding a cluster constraint for the previous selection and updating of the background distribution.

http://www.iki.fi/kaip/sider.html. The user can add data points to a selection by directly marking them, by using pre-defined classes in the dataset, or previously saved groupings. The time-consuming operations are executed only by a direct command of the user, which makes the system responsive and predictable. For further details of SIDER, see [2].

Our focus in the experimental part is to show how SIDER is able to provide the user with insightful projections of data and reveal the differences between the background distribution and the data. Additionally, the user interface makes it easy to explore various statistics and properties of selections of data. We test the system with data set sizes typical for interactive systems (on the order of thousands of data points); if there are more it often makes sense to downsample the data first.

Runtime experiment. We generated synthetic datasets parametrized by the number of data points (n), dimensionality (d), and the number of clusters (k) by randomly sampling k cluster centroids and allocating data points around each of the centroids. We added column constraints ($2d$ constraints) for each dataset, and for the sets with $k > 1$, a cluster constraint for each cluster in the data ($2dk$ constraints). In Table I the median wall clock running times are provided without any cutoff, based on 10 runs for each set of parameters, ran on a Apple MacBook Air (2.2 GHz Intel Core i7) and a single-threaded R 3.4.0 implementation of the algorithm.

The algorithm has the following steps: (1) Initialization, (2) optimization for the correct parameters, (3) preprocessing for sampling and whitening, producing (4) a whitened dataset and (5) a random sample of the MaxEnt distribution, and (6) running the PCA and ICA algorithms. Step (2) takes most time. As expected (see Table I) the time consumed does not depend on the number of rows n. Each optimization step takes $O(d^2)$ time per constraint and there are $O(kd)$ constraints. In SIDER the default is to stop the optimization after a time cut-off of 10 seconds, even when convergence has not been achieved. For larger matrices the time consumed by ICA becomes significant, scaling roughly as $O(nd^2)$. All the other steps always take less than 2 seconds each and are not reported.

BNC data. The British National Corpus (BNC) [6] is one of the largest annotated text corpora freely available in full-text format. As a high dimensional use case we explore the high-level structure of the corpus. For preprocessing, we compute the vector-space model (word counts) using the first 2000 words from each text belonging to one the four main genres ('prose fiction', 'transcribed conversations', 'broadsheet newspaper', 'academic prose') as in [7]. After preprocessing we have 1335 texts and use the 100 words with highest counts as the dimensions and the main genres as the class information.

The most informative PCA projection of the BNC data is shown in Fig. 2 (left). In the upper right corner there is a group of points (red selection) that appear to form a group. These points are mainly from 'transcribed conversations' (Jaccard-index to class 0.928). After we added a cluster constraint for this selection, updated the background distribution and computed a new PCA projection, we obtained the projection in Fig. 2 (right). The next selection shows another set of points (mainly from classes 'academic prose' and 'broadsheet newspaper'; Jaccard-indices 0.63 and 0.35) differing from the background distribution. After adding a cluster constraint for this selection, we updated the background distribution and computed another PCA projection. There was no apparent difference to the background distribution (as reflected in low PCA scores), and we conclude that the identified clusters explain the data well wrt. variation in counts of the most frequent words. Notice class labels were only used retrospectively.

UCI Image Segmentation data. As a second use case, we have the Image Segmentation dataset from the UCI machine learning repository [8] with 2310 samples. Initially, the background distribution has a much larger variance than the data in the first PCA projection. Thus, we first added a 1-cluster constraint for the data (overall covariance) and updated the background distribution. After this, in Fig. 3 (left), we can observe $\geq$ 3 sets of points quite clearly separated in the projection. The selection of points in red contains solely points from the class 'sky', while the points clustered in the lower left corner are mainly from the class 'grass' (with Jaccard-

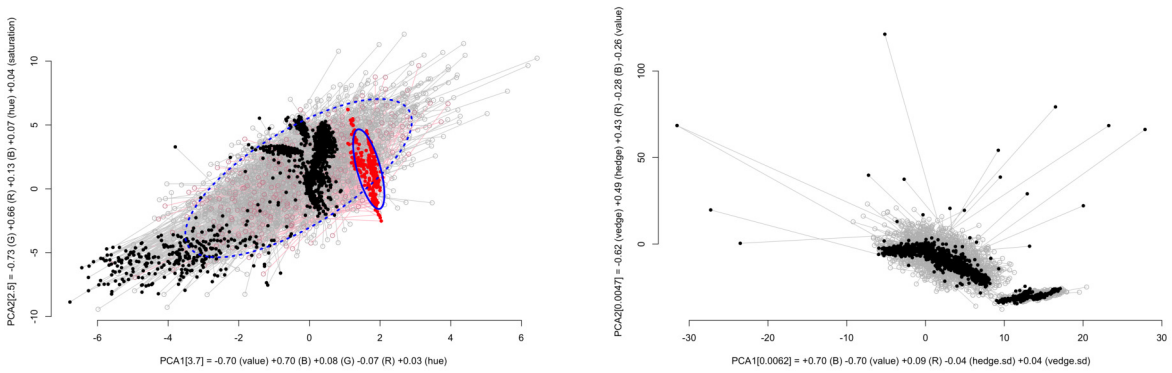

Fig. 3. A use case with the UCI Image Segmentation data. Left : Initially the scale of background distribution significantly differs from that of the data, after a 1-cluster constraint is added and the background distribution updated, there is visible structure present. Cluster constraints are added for the three sets of points clustered in this view, and the background distribution is updated accordingly; Right: The next PCA projection shows mainly outliers.

index 0.964). The set of points clustered in the middle are mainly from classes 'brickface', 'cement', 'foliage', 'path', and 'window' (with Jaccard-index approx. 0.2 each). We add a cluster constraint for each of these sets of points, and update the background distribution, after which the the background distribution matches the data rather well with the exception of some outliers. The next PCA projection (Fig. 3 (right)) reveals that indeed there are outliers. For brevity, we did not continue the analysis, but the data obviously contains a lot more structure that we could explore in subsequent iterations.

IV. Conclusions

There have been many efforts in analysis of multivariate data in different contexts, e.g., using Projection Pursuit and manifold learning methods for compressing the data into a lower dimensional—typically 2-D—presentation while preserving features of interest. The drawback is that the criteria for dimensionality reduction are defined typically in advance and it may or may not fit the user's need. It may be that a visualization shows only the most prominent features of the data already known for the user, or features that are irrelevant for the task at hand. A natural alternative to static visualizations using pre-defined criteria is the addition of interaction. The drawback of such interactions is, however, that they lack the sheer computational power utilized by the dimensionality reduction methods.

Our method fills the gap between automated dimensionality reduction methods and interactive systems. We propose to model the knowledge of a domain expert by a probability distribution computed by using the Maximum Entropy criteria. Furthermore, we propose powerful and yet intuitive interactions for the user to update the background distribution. Our approach uses Projection Pursuit methods and shows the directions in which the data and the background distribution differ the most. In this way, we utilize the power of Projection Pursuit at the same the allowing the user to adjust the criteria by which the computers chooses the directions to show her.

The current work presents a framework and a system for real-valued data and the background distribution modeled by multivariate Gaussian distributions. The ideas could be generalized to other data types (e.g., categorical or ordinal data), or to higher-order statistics, likely in a straightforward manner, as the mathematics of exponential family distribution would lead to similar derivations. For concrete applications for our approach and the SIDER tool there is potential in, e.g., computational flow cytometry. Initial experiments with samples up to tens of thousands rows from flow-cytometry data [9] has shown the computations in SIDER to scale up well and the projections to reveal structure in the data potentially interesting to the application specialist.

Acknowledgements. This work has been supported by the ERC under the EU's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 615517, the FWO (project no. G091017N, G0F9816N), the EU's Horizon 2020 research and innovation programme and the FWO under the MSC Grant Agreement no. 665501, the Academy of Finland (288814, 313513), and Tekes (Revolution of Knowledge Work project).

References

[1] J. Tukey, *Exploratory data analysis*. Addison-Wesley, 1977.
[2] K. Puolamäki, E. Oikarinen, B. Kang, J. Lijffijt, and T. De Bie, "Interactive Visual Data Exploration with Subjective Feedback: An Information-Theoretic Approach," arXiv:1710.08167 [stat.ML], 2017.
[3] J. Lijffijt, B. Kang, W. Duivesteijn, K. Puolamäki, E. Oikarinen, and T. De Bie, "Subjectively interesting subgroup discovery on real-valued targets," arXiv:1710.04521 [stat.ML], 2017.
[4] T. Cover and J. Thomas, *Elements of Information Theory*, 2nd ed. Wiley, 2005.
[5] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE T. Neural Networ.*, vol. 10, no. 3, pp. 626–634, 1999.
[6] "The British National Corpus, v. 3 (BNC XML Edition)," Distributed by Oxford University Computing Services on behalf of the BNC Consortium, 2007. [Online]. Available: http://www.natcorp.ox.ac.uk/
[7] J. Lijffijt and T. Nevalainen, "A simple model for recognizing core genres in the BNC," *Studies in Variation, Contacts and Change in English*, vol. 19, 2017.
[8] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml
[9] Y. Saeys, S. Van Gassen, and B. Lambrecht, "Computational flow cytometry: helping to make sense of high-dimensional immunology data," *Nat. Rev. Immunol.*, vol. 16, no. 7, pp. 449–462, 2016.