InVis: A Tool for Interactive Visual Data Analysis

Daniel Paurat and Thomas Gärtner

University of Bonn and Fraunhofer IAIS, Sankt Augustin, Germany
{daniel.paurat,thomas.gaertner}@uni-bonn.de
http://kdml-bonn.de

Abstract. We present **InVis**, a tool to visually analyse data by interactively shaping a two dimensional embedding of it. Traditionally, embedding techniques focus on finding one fixed embedding, which emphasizes a single aspects of the data. In contrast, our application enables the user to explore the structures of a dataset by observing and controlling a projection of it. Ultimately it provides a way to search and find an embedding, emphasizing aspects that the user desires to highlight.

1 Introduction

We present an application[1] that enables the user to layout a two dimensional embedding of a possibly higher dimensional dataset by selecting and rearranging some of the embedded data points as *control points*. Working with our application resembles observing the shadow of a higher dimensional object from different angles and actively reshaping it. As the constellation of control points and the projection angle are dependent, specifying where the shadow of the chosen control points falls to, enforces the rest of the embedding to follow, see Figure 1. Gradually rearranging the constellation also changes the shadow gradually. An example for this is depicted on a real world dataset in Figure 2.

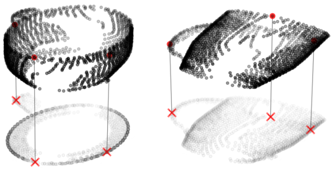


Fig. 1. The projection can be controlled by arranging the control points (dataset from [7]).

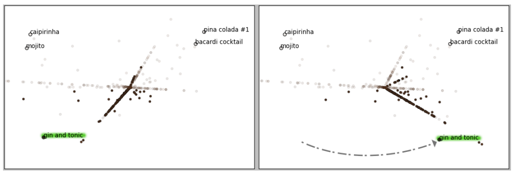


Fig. 2. Re-positioning the control point *gin and tonic* (green) influences the location of related, gin containing, cocktails (dark).

Embedding data into a lower dimensional space for visual analysis is a wide field that is approached by a lot of different techniques. Many of them are unsupervised, like the well known *principle component analysis* (PCA) [5], *Isomap*

[1] The tool can be downloaded under: http://www-kd.iai.uni-bonn.de/index.php?page=software_details&id=31

[11], *Locally linear embedding* [10], *non-negative matrix factorization* [6], *archetypal analysis* [3] and *CUR decomposition* [4].

Apart from these unsupervised embedding techniques, there are methods that take supervision into account, like *guided locally linear embedding* [1] and *supervised PCA* [2]. Many of the classic embedding methods also have a semi supervised extensions [12]. One particularly interesting setting is utilizing *must-link* and *cannot-link* constraints [13]. In this paper we employ the semi-supervised *least squares projections* (LSP) [8, 9] method, which computes an embedding based on a set of exemplary embedded data points.

In contrast to other authors applying semi-supervised embedding techniques, our aim is not a fixed one-time-embedding. Our application rather exploits the influence of the control points in order to enable the user to shape and steer a life-updating embedding. This active layout approach ultimately empowers the user to highlight aspects of the dataset that he considers interesting. This is illustrated in Figure 3 on a selection of four persons from the *CMU Face Images* dataset. While a regular PCA embedding does not directly convey insights, arranging a few control points in different constellations, can highlight different semantic aspects of the data.

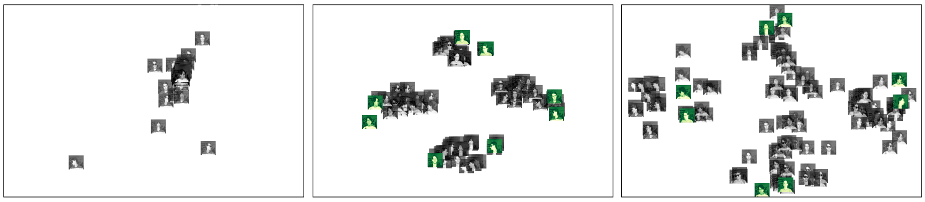


Fig. 3. A dataset of facial images embedded in different ways. The left figure shows a plain PCA embedding, while the other two figures use LSP to group the control points by person and by pose (*looking-straight*, *-up*, *-left* and *-right*), respectively.

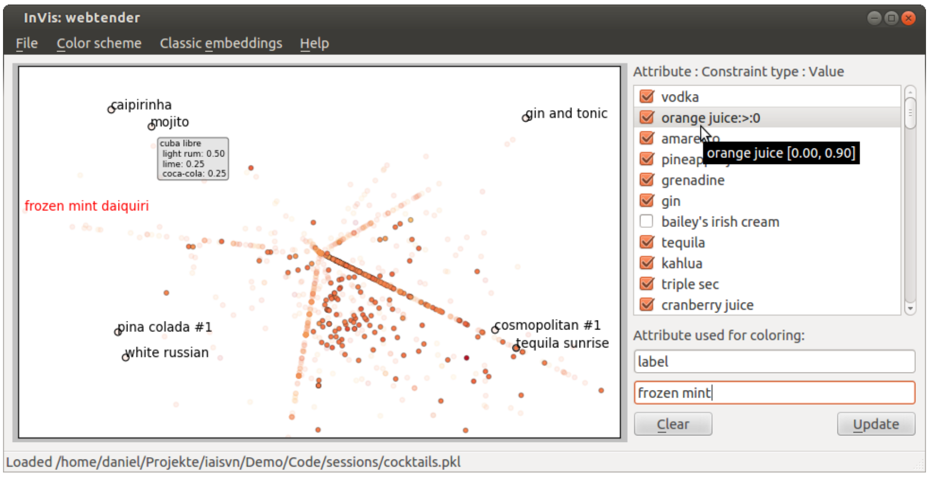
2 Method

Consider a dataset X with n data records $x_1, ..., x_n$ from an instance space $\mathcal{X} \subseteq \mathbb{R}^d$ and the general task to map $\{x_1, ..., x_n\}$ into an embedding space $\mathcal{Y} \subseteq \mathbb{R}^2$, yielding $\{y_1, ...y_n\}$. To determine this mapping, the user chooses a set of k data records from X, denoted by $\hat{X}$, and fixes their coordinates in the embedding space, providing $\hat{Y}$. For the purpose of our application, we consider the desired projection $P : \mathcal{X} \to \mathcal{Y}$ to be the linear projection matrix with the least squared error in mapping $\hat{X}$ to $\hat{Y}$. Regarding $\hat{X}$ and $\hat{Y}$ as data matrices of shape $d \times k$ and $2 \times k$ we can formulate the system of linear equations $P\hat{X} \approx \hat{Y}$, which can be solved for P with least squared error efficiently, especially since the calculation only depends on k and not all n data points. The least squares projection matrix P is then used to determine the final embedding Y of all n data points X by matrix multiplication $PX = Y$. Note, that every time $\hat{Y}$ changes P has to be recalculated. P can be derived by right multiplying the pseudo inverse of $\hat{X}$ (given by $\hat{X}^\dagger = \hat{X}^T(\hat{X}\hat{X}^T)^\dagger$) to $\hat{Y}$. As long as the user only relocates the k control points, $\hat{X}^\dagger$ does not change and P can be determined by matrix multiplication with a time complexity of $\mathcal{O}(d^2 \cdot k)$. However, if the user alters the selection of the control points, the pseudo inverse $\hat{X}^\dagger$ has to be recalculated, which leads to an additional calculation with a time complexity of $\mathcal{O}(d^3 + d^2 \cdot k)$.

3 User Interface

Figure 4 shows the user interface of our application running an exemplary analysis of a cocktail ingredient dataset. The core of our tool is the interactive canvas on the left side, displaying the embedding. The initial control points are provided by five randomly chosen points, placed according to their coordinates of a PCA-embedding on the whole dataset. From here the user can interact with the canvas in the usual way by clicking and dragging. The user can select, or de-select a control point by middle-clicking it and he can reposition the point simply by left-clicking and dragging it to the new location. While relocating a control point, the embedding is constantly updated to provide the user with a "hands on" sensation.

To support practical usability of the application, we also provide some extra features that can help a user in the exploration process. The user can shift the center of the displayed data by *Ctrl*-dragging on the canvas and zoom in and out of different regions by using the mouse wheel. He can also search for a data record by its name and highlight it in the embedding, or request additional information on any data point by right-clicking the according point in the embedding. In case of the cocktail dataset, ingredients and amounts of the particular cocktail are displayed. In case of an image database a thumbnail picture is rendered into the embedding. For deeper inspection of an attribute, we offer the option to colorize the data points according to the attribute-value and to fade out data points that



cuba libre
light rum: 0.50
lime: 0.25
coca-cola: 0.25

☑ vodka
☑ orange juice:>:0
☑ amaretto
☑ pineap
☑ grenadine
☑ gin
☐ bailey's irish cream
☑ tequila
☑ kahlua
☑ triple sec
☑ cranberry juice

orange juice [0.00, 0.90]

label

frozen mint

Loaded /home/daniel/Projekte/iaisvn/Demo/Code/sessions/cocktails.pkl

Fig. 4. A screenshot of InVis. The embedding shows some control points (black), a search result (red) and ingredient information (gray box). Interaction with the embedding is done by directly clicking and dragging. Setting the constraint "*orange juice:>:0*" fades the points not satisfying the restriction out.

do not satisfy a $\{>, <, =\}$-constraint. In addition, unwanted attributes can be excluded from any calculation and running sessions can be saved and restored.

4 Conclusion

We present a tool that encourages the user to explore a dataset in a "hands on" manner, by directly interacting with an embedding of it. In contrast to traditional one-time-embeddings our approach enables the user to develop a feeling for the underlying structure of the dataset by browsing it from different angles and layout the embedding in such a way that user desired aspects are emphasized.

Acknowledgments: Part of this work was supported by the German Science Foundation (DFG) under the reference number 'GA 1615/1-1'.

References

1. B. Alipanahi and A. Ghodsi. Guided locally linear embedding. *Pattern Recognition Letters*, 32(7):1029 – 1035, 2011.
2. E. Barshan, A. Ghodsi, Z. Azimifar, and M. Zolghadri Jahromi. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, 44(7):1357–1371, 2011.
3. A. Cutler and L. Breiman. Archetypal analysis. *Technometrics*, 36(4):338–347, 1994.
4. P. Drineas, R. Kannan, and M.W Mahoney. Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006.
5. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics. Springer New York Inc., 2001.
6. D.D. Lee, H.S. Seung, et al. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.
7. M. Neumann, R. Garnett, P. Moreno, N. Patricia, and K. Kersting. Propagation kernels for partially labeled graphs. In *ICML–2012 Workshop on Mining and Learning with Graphs (MLG–2012)*, Edinburgh, UK, 2012.
8. J.G.S. Paiva, W.R. Schwartz, H. Pedrini, and R. Minghim. Semi-supervised dimensionality reduction based on partial least squares for visual analysis of high dimensional data. In *Computer Graphics Forum*, volume 31, pages 1345–1354. Wiley Online Library, 2012.
9. F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *Visualization and Computer Graphics, IEEE Transactions on*, 14(3):564–575, 2008.
10. S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
11. J.B. Tenenbaum, V. De Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
12. X. Yang, H. Fu, H. Zha, and J. Barlow. Semi-supervised nonlinear dimensionality reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 1065–1072. ACM, 2006.
13. D. Zhang, Z.H. Zhou, and S. Chen. Semi-supervised dimensionality reduction. In *Proceedings of the 7th Siam International Conference on Data Mining*, pages 629–634, 2007.