Opening the black-box of Neighbor Embedding with Hotelling's T^2 statistic and Q-residuals

Roman Josef Rainer[a], Michael Mayr[a], Johannes Himmelbauer[a], Ramin Nikzad-Langerodi[a]

[a]*Software Competence Center Hagenberg, Softwarepark 32a, Hagenberg, 4232, , Austria*

Abstract

In contrast to classical techniques for exploratory analysis of high-dimensional data sets, such as principal component analysis (PCA), neighbor embedding (NE) techniques tend to better preserve the local structure/topology of high-dimensional data. However, the ability to preserve local structure comes at the expense of interpretability: Techniques such as t-Distributed Stochastic Neighbor Embedding (t-SNE) or Uniform Manifold Approximation and Projection (UMAP) do not give insights into which input variables underlie the topological (cluster) structure seen in the corresponding embedding. We here propose different "tricks" from the chemometrics field based on PCA, Q-residuals and Hotelling's T^2 contributions in combination with novel visualization approaches to derive local and global explanations of neighbor embedding. We show how our approach is capable of identifying discriminatory features between groups of data points that remain unnoticed when exploring NEs using standard univariate or multivariate approaches.

Keywords: neighbor embedding, t-SNE, UMAP, PCA, chemometrics

1. Introduction

Neighbor embedding techniques have gained increasing popularity in a wide variety of scientific disciplines such as machine learning [1], biology [2], physics [3] or engineering [4]. In contrast to classical techniques for exploratory analysis of high-dimensional data sets, such as principal component analysis (PCA), these methods tend to better preserve the "intrinsic structure" or (local) topology of high-dimensional data when mapped to (i.e. embedded in) low-dimensional spaces which often provides additional insights. However, the ability to preserve local structure comes at the expense

of interpretability: Techniques such as t-Distributed Stochastic Neighbor Embedding (t-SNE) [1] or Uniform Manifold Approximation and Projection (UMAP) [5] do not give insights into which input variables underlie the topological (cluster) structure seen in the corresponding embedding. Tools, such as Embedding Projector (EP) [6] or t-viSNE [7] can help to better understand the corresponding embedding. EP provides a web-based, interactive environment that allows different "views" on high-dimensional data sets and their embedding. However, while EP is useful to explore embedding in terms of pre-defined labels, there is no possibility to investigate the relationship between (semantically) meaningful directions in the embedded space and the input variables. In general, inspection of how single variables change across an embedding is straightforward and can be undertaken for low-dimensional data sets but becomes prohibitive for data sets with hundreds of thousands of variables. In order to address this issue, t-viSNE comes with a so-called "dimension correlation tool", which allows to explore the correlation between the input variables and a user-defined (e.g. local and/or non-linear) direction in the embedded space. Bibal et al. ([8]) employ a modified LIME (local interpretable model-agnostic explanations) approach to explain t-SNE embedding. Similar to t-viSNE, this approach derives "explanations" for local neighborhoods around particular data points in terms of the input variables. Both methods derive local explanations rather than explaining the (global) directions of an embedding and require considerable computational resources for data sampling and local modelling. In addition, both methods require considerable user-interactions, which might restrict their use to expert users familiar with the techniques. An alternative approach has been recently proposed in [9], where the authors use local neighborhood information derived from UMAP as an additional constraint in PCA. However, for some data sets, replication of the (non-linear) neighborhood structure with a linear model is difficult and eventually yields latent variables (LVs) that capture the noise rather than the systematic variation in the data.

In the current contribution, we propose a simple, yet effective, workflow to derive explanations of local and global directions of (non-linear) embedding that is computationally efficient and reduces the overhead of user-interactions. In brief, we first create the corresponding embedding and fit a PCA model to the input data. We then employ relative Hotelling's T^2-contributions to derive explanations for the difference between individual data points or entire clusters. In addition, we propose novel visualizations that include the PCs scores along with the corresponding Q-residuals in order

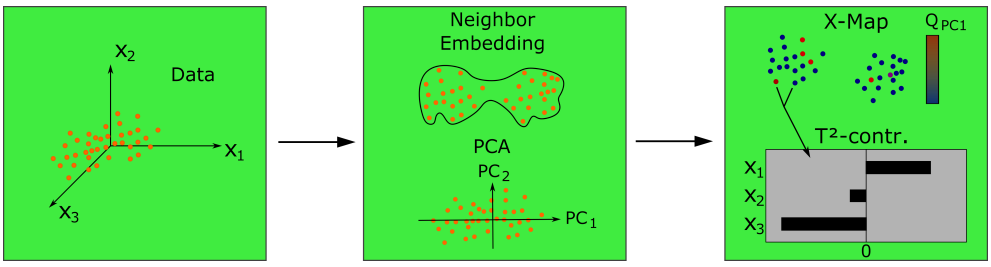


Figure 1: PCA-derived explanations of neighbor embedding by means of relative Hotelling's T^2 contributions and Q-residuals.

to understand which data points are well characterized by individual LVs underlying the data set. The main components of our approach are summarized in Figure 1.

We will focus on deriving PCA-based explanations of UMAP embedding, which build upon t-SNE and now belongs to the most widely used neighbor embedding methods in the data science community. In general, however, the same workflow can be applied to any (non-linear/parametric) dimensionality reduction method. We refer to the excellent review by Wang et al. [10] for an overview of the current state-of-the-art dimension reduction techniques.

2. Theory

2.1. Principal Component Analysis (PCA)

Principal component analysis (PCA) is among the most widely used techniques for exploratory analysis of multivariate datasets [11]. PCA decomposes an $I \times J$ (e.g. time points $\times$ process variables) matrix $\mathbf{X}$ into a set of *scores* $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_A]^{I \times A}$ and *loadings* $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_A]^{J \times A}$ vectors such that [12]

$$\begin{aligned} &\mathbf{X} = \sum_{i=1}^{A} \mathbf{t}_i \mathbf{p}_i^T + \mathbf{E}. \\ &\textbf{s.t.} \;\; \mathbf{t}_i^T \mathbf{t}_j = 0 \text{ and } \mathbf{p}_i^T \mathbf{p}_j = 0 \;\text{ for }\; i \neq j. \end{aligned} \tag{1}$$

Ideally, A in Eq.(1) is chosen such that the systematic variation in $\mathbf{X}$ is encoded in the mutually orthogonal principal components (PCs) $\mathbf{t}_1\mathbf{p}_1^T, \ldots, \mathbf{t}_A\mathbf{p}_A^T$ while the random noise is encoded in the $I \times J$ residual matrix $\mathbf{E}$. The strength of PCA is rooted in the fact that each PC represents a separate

source of variation underlying the observed data that can be analyzed at the samples (scores) and variables (loadings) level. While the former allows to detect (e.g. temporal) trends in (process) data, the latter allows to investigate the correlations between the original variables and those trends. In addition, computation of the variance explained by each PC is straightforward, allowing to quantify the contribution of a trend to the total variance in a data set. Finally, PCA fits a bilinear model to the data, which can be useful when aiming at making predictions on new data. This is exploited e.g. in multivariate statistical process control (MSPC) to detect departure from normal operating conditions ([13]) or in classification problems ([14]).

Despite its versatility, several characteristics of PCA are less favorable for data analysis. First, it is important to note that the objective of PCA is to find directions in the high-dimensional space of the input variables where the (explained) variance is largest. Consequently, PCA is good at capturing global trends while largely neglecting local structures that might be present and eventually encode important information about the process under investigation. On the other hand, the latent variables (LVs) that span the PCA subspace are linear combinations of the original variables. According to the central limit theorem, linear combinations of random variables converge to normal distributed LVs [15] as the number of variables increases and scores plots thus often tend to show spherically shaped clusters where the natural topology of the data is "distorted" which is particularly the case for very high-dimensional data

Over the past decades, several dimension reduction methods have been proposed that better preserve the (local) structure/topology of high dimensional data, two of which we find particularly useful. These shall thus be introduced in the next sections. These will therefore be briefly introduced in the following sections

2.2. Neighbor Embedding

The general idea of Neighbor Embedding (NE) is to preserve (local) distances between data points in the high-dimensional space when deriving the corresponding low-dimensional representation.

t-SNE. t-SNE models the distance between two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the high-dimensional space as the (conditional) probability

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \tag{2}$$

with

$$p_{j|i} = \frac{v_{j|i}}{\sum_{k \neq i} v_{j|i}}; \quad v_{j|i} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma_i^2}\right). \tag{3}$$

N denotes the number of data points, and σ_i (i.e. the Gaussian kernel bandwidth) controls how fast the probability decreases with increasing Euclidean distance from the i-th data point. Instead of manually choosing the (optimal) bandwidth for each data point, t-SNE "automatically" adjusts σ to be small in dense regions and large in sparse regions of the input space based on the so-called perplexity parameter

$$\text{Perplexity}(P_i) = 2^{-\sum p_{j|i} \log_2 p_{j|i}} \tag{4}$$

that needs to be specified by the user. The perplexity is related to the number of (close) neighbors around each data point that should be considered in the embedding. A large perplexity emphasizes global while a small one focuses on local structure. The distance in the (low-dimensional) embedded space is modeled using Student's t-distribution with the corresponding probability

$$q_{ij} = \frac{w_{ij}}{\sum_{k \neq l} w_{kl}}; \quad w_{ij} = \frac{1}{1 + \|\mathbf{y}_i - \mathbf{y}_j\|_2^2}, \tag{5}$$

where $\mathbf{y}_i$ denotes the (two-dimensional) coordinates of the i-th data point in the embedded space. Finally, the so-called *Kullback-Leibler* (KL) divergence between the high- and low dimensional distributions P and Q with respect to these coordinates is minimized, i.e.

$$\min_{\mathbf{y}_i, \cdots, \mathbf{y}_N} \text{KL}(P||Q) \tag{6}$$

with $\text{KL}(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$. Technically, this is achieved by means of gradient descent using some random initialization of the y-coordinates.

UMAP. Conceptually, t-SNE and UMAP share the same basic idea with subtle differences related to how similarity is modelled. Most importantly, UMAP does not normalize pair-wise distances, i.e. $\sum_{i,j} p_{ij} \neq 1$ (the same is true for q_{ij}), employs

$$p_{j|i} = \exp\left(-\frac{\text{d}(\mathbf{x}_i, \mathbf{x}_j) - \rho}{\sigma}\right) \tag{7}$$

as distance metric and minimizes cross-entropy

$$\mathrm{CE}(P||Q) = \sum_{i \neq j} \left(p_{ij} \cdot \log \left(\frac{p_{ij}}{q_{ij}} \right) + (1 - p_{ij}) \cdot \log \left(\frac{1 - p_{ij}}{1 - q_{ij}} \right) \right) \tag{8}$$

instead of KL-divergence. The parameter ρ in Eq.(7) represents the distance from the i-th data point to its nearest neighbor and implies a different distance metric for each data point (that in turn requires a symetrization that is slightly different from Eq. (2)). $\mathrm{d}(\mathbf{x}_i, \mathbf{x}_j)$ denotes some distance (e.g. Euclidean distance) between $\mathbf{x}_i$ and $\mathbf{x}_j$. In addition, UMAP uses the number of k nearest neighbors

$$k = 2^{-\sum p_{ij}} \tag{9}$$

when computing $p_{j|i}$ and the family of curves

$$q_{ij} = \left(1 + a \cdot (\|\mathbf{y}_i - \mathbf{y}_j\|_2^2)^b\right)^{-1} \tag{10}$$

instead of perplexity and Student t-distribution, respectively. For the latter, the parameters a and b are obtained by non-linear least-squares fitting.

t-SNE vs. UMAP. The fact, that UMAP does not normalize the "probabilities" not only makes it faster (due to omission of sum computations) and consume less memory (replacement of gradient descent by stochastic gradient descent) but also allows one to "map" new samples to the embedding without changing the position of the training samples, e.g. by computing

$$\hat{\mathbf{y}}_{N+1} = \min_{\mathbf{y}_{N+1}} \mathrm{CE}(P_{\mathbf{y}_1,\ldots,\mathbf{y}_{N+1}} || Q_{\mathbf{y}_1,\ldots,\mathbf{y}_{N+1}}). \tag{11}$$

With t-SNE this is not possible, since any new data point changes $p_{j|i}$ and $q_{j|i}$ (through normalization) and thus the (optimal) coordinates of the training samples. In addition, UMAP better balances global and local structure preservation. This is because data points that are far apart (small p_{ij}) contribute little to the KL-divergence but contribute to CE through the second term in Eq. (8).

2.3. Interpretability

Unlike in PCA, where each embedded sample can be expressed as linear transformation of the corresponding inputs, i.e.

$$\mathbf{y}_i = \mathbf{x}_i^{\mathrm{T}} \mathbf{P}, \tag{12}$$

the (non-linear) map $f : \mathcal{X} \to \mathcal{Y}$ remains a black-box in both t-SNE and UMAP. Since there is no model, interpretability is not given. To mitigate this issue we propose two complementary workflows to better understand the embedded space.

2.4. Our approach

The idea of using PCA as preprocessing step or to use PCA scores (i.e. individuals columns of the $\mathbf{T}$ matrix from Eq. (1)) as color code for NE has been proposed earlier (e.g in [7]) and is a useful tool to get a quick overview about which variables contribute to the topological structure of the embedding. We extend this idea by introducing two concepts widely used in chemometrics: Q-residuals and relative T^2-contributions, and introduce Voronoi plots for visualization of the corresponding NE.

Q-Residuals. are also known as reconstruction loss in the machine learning community. For an input sample $\mathbf{x}_i$

$$Q_i = \mathbf{x}_i(\mathbf{I} - \mathbf{P}_A^{\mathrm{T}}\mathbf{P}_A)\mathbf{x}_i^{\mathrm{T}}, \tag{13}$$

with $\mathbf{P}_A$ denoting the loadings matrix of a PCA model with A components. Samples with low Q-residuals are well represented by the model, i.e. they have small (orthogonal) distances to their low-dimensional projections (Figure 2). When using the PC scores $\mathbf{t}_k$ for $k \in 1, \ldots, A$ as color code for the embedded data, we here propose to always also display the corresponding Q-residuals in order to see for which samples the corresponding "explanations" have strong (small Q) and weak (large Q) support. For the latter, a different PC might better explain their location in the embedding. PC specific Q-residuals are calculated by replacing $\mathbf{P}_A$ in Eq.(13) by the k-th loadings vector $\mathbf{p}_k$.

Relative T^2-contributions. The Hotelling's T^2-statistic is a measure of outlingness w.r.t. the centroid of the (training) data within the PC space (Figure 2). The corresponding T^2-contributions are a measure of influence of each input variable to the outlingness. For the i-th sample

$$\mathbf{t}_{\mathrm{cont},i} = \mathbf{t}_i\mathbf{\Lambda}^{-1/2}\mathbf{P}_A, \tag{14}$$

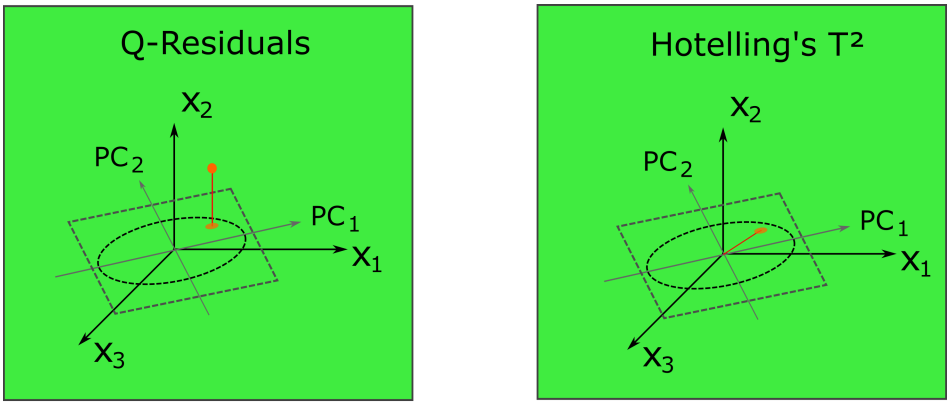


Figure 2: Q-Residuals and Hotelling's T^2-statistic.

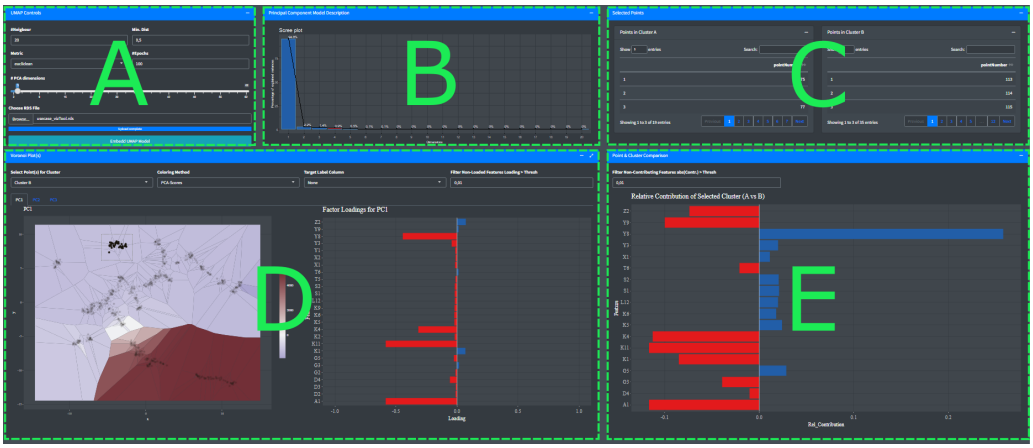


Figure 3: The X-MAP tool at a glance. A) Data import, UMAP settings and number of PC's. B) Explained variance per PC. C) Selected data points from the UMAP embedding. D) Voronoi plot of the embedded data (color-coded either by PCA score, Q-residuals or a single input variable) and loadings of the corresponding PC. E) Relative T^2 contributions explaining the difference between two data points or clusters selected in D.

with $\mathbf{\Lambda}$ being a diagonal matrix holding the A leading Eigenvalues of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$[1]. The corresponding scaling takes care that the same weight is assigned to the contribution from each subspace dimension to the explanations. Relative T^2-contributions on the other hand, can be used to investigate which

[1] `https://wiki.eigenvector.com/index.php?title=T-Squared_Q_residuals_and_Contributions`

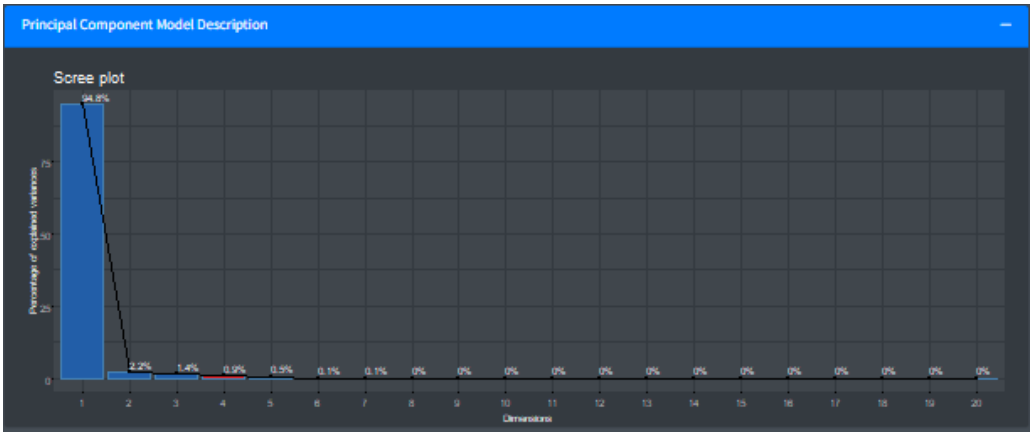


Figure 4: Fraction of explained variance for each PC. The number of PCs included in the model is indicated by the red bar (#PCs = 4).

variables contribute most to the difference between $\mathbf{x}_i$ and $\mathbf{x}_j$ w.r.t. the (A-dimensional) PC space:

$$\mathbf{t}_{\text{cont},ij} = \mathbf{t}_{\text{cont},i} - \mathbf{t}_{\text{cont},j} \tag{15}$$

The same approach can be used to derive explanations for why entire clusters within an embedding differ from each other. To do so, the contributions are computed with respect to the corresponding cluster centroids.

Voronoi plots. In order to better explore the non-linear structure of the embedded space, we propose Voronoi diagrams. A Voronoi diagram is a partitioning of a plane into regions with each point in that region being closest to a single data point in terms of Euclidean distance. This allows to see the value of a statistic to be displayed (e.g. Q-Residuals) for individual data points over the whole embedded space.

Implementation. A prototypical implementation of the entire data exploration workflow including graphical user interface (GUI) was undertaken using the shiny web application framework for R [16]. We provide open source code that will be hosted on a public Git repository upon acceptance of the manuscript.

3. X-Map GUI

Figure 3 outlines the functionality of the proposed data exploration tool that we coin explainable map (X-MAP). First, a .rds file holding the data

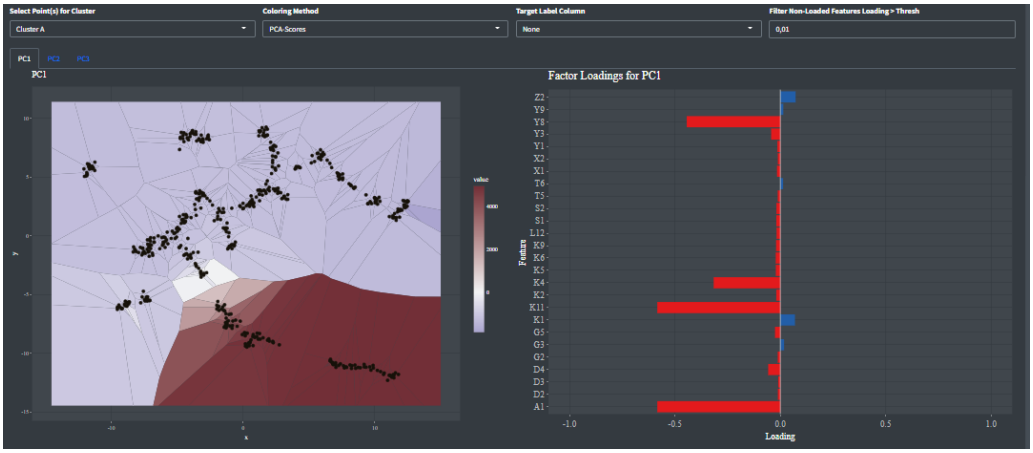


Figure 5: Voronoi diagram color-coded according to PC1 scores (left) and corresponding loadings (right).

to be explored must be provided in long format followed by specifying the UMAP parameters, i.e. number of nearest neighbors, minimum distance (see Eq. (7), similarity metric, and number of training epochs (Figure 3A). Subsequently, the maximum number of PC's is specified followed by fitting the PCA model and UMAP embedding. Once model fitting has completed, the percentage of variance explained by each PC appears in a separate subplot (Figure 4).

In the example shown, more than 94% of the variance is explained by the first PC. Note that an optimal number of PCs to be retained is proposed and automatically selected during model fitting but can be manually adjusted after completion of the training epochs.

In addition to the explained variance plot, a Voronoi diagram of the UMAP embedding appears with the polygones being by default colored according to the score of PC 1 side by side with the corresponding loadings (Figure 5).

In the example shown, data points located in the lower right corner of the embedding exhibit a higher score on PC 1 compared to the points located in the blue areas. Separation between the two areas is mostly due to lower values of variables Y8, K4, K11 and A1 for the former as can be seen by inspecting the corresponding loadings. Figure 6 shows the same embedding with the data points color-coded according to the value of variable Y8, which is in-line with this interpretation. PC 2, on the other hand, encodes infor-

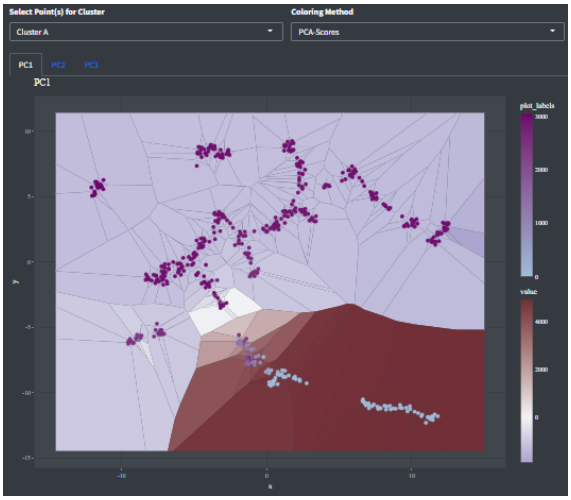


Figure 6: Data points colored according to Y8 variable.

mation about differences between clusters in the lower half of the UMAP (Figure 7). An important criterion for PCA-based interpretation of a non-

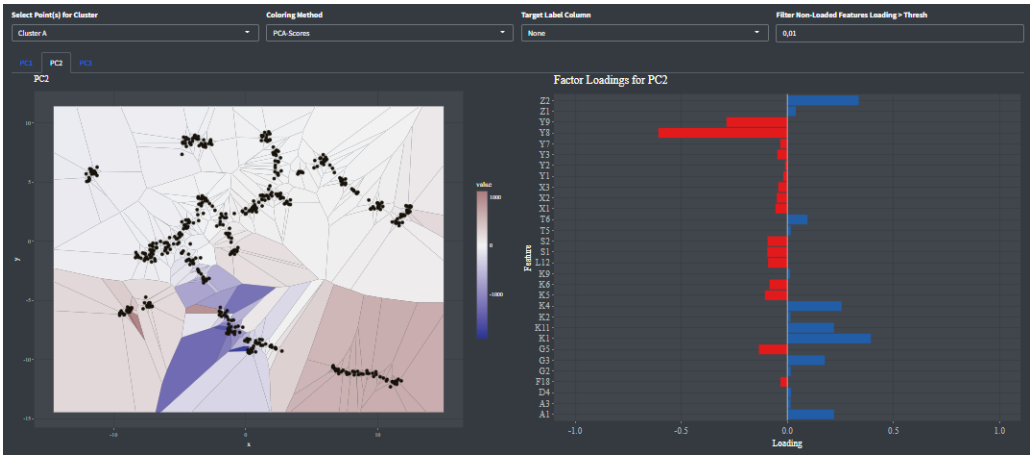
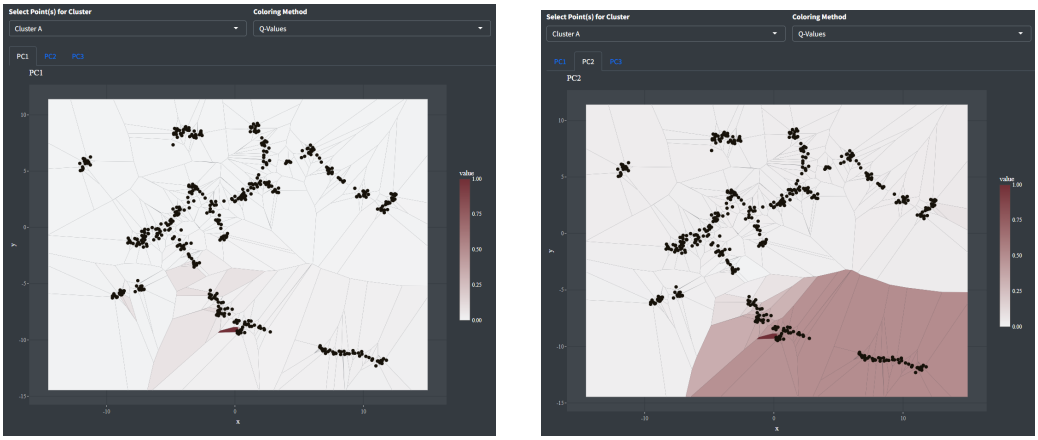


Figure 7: Voronoi diagram color-coded according to PC2 scores (left) and corresponding loadings (right).

linear embedding like UMAP is, how well each data point is represented by the current PC. In the worst case, a PC that explains a high proportion of variance in the data (i.e. PC 1 in our example) only explains a small portion of the information contained in some data points. Q-Residuals are a means of



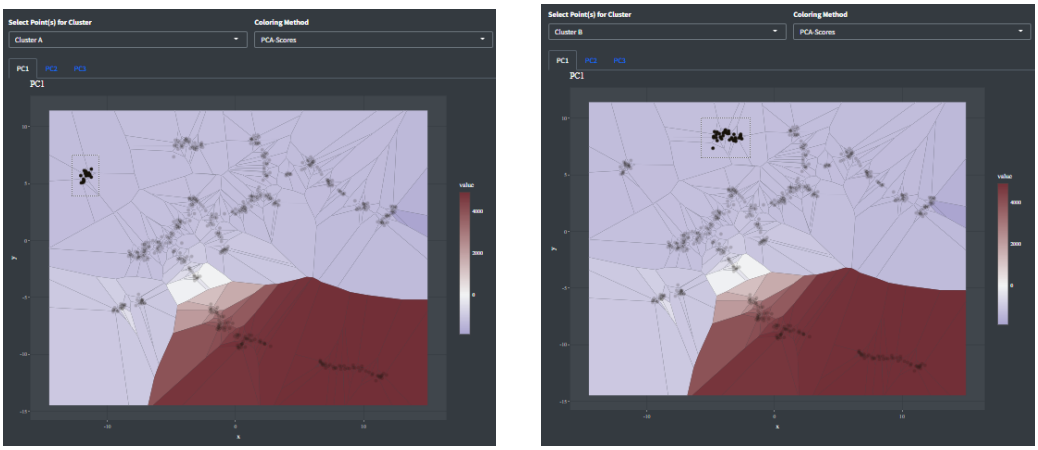
(a) Principal component 1

(b) Principal component 2

Figure 8: PC-specific Q-Residuals. Note that Q-values are normalized between 0 and 1.

diagnosing if some data points, i.e. outliers, are poorly represented. Figure 8a and 8b show the same Voronoi diagramm of the embedding with the polygones color-coded according to the Q-residuals with respect to the first and second PC. Note that the values are min-max normalized and lie between 0 and 1 for samples with the lowest and highest Q-residuum in the data set, respectively. Notably, a single data point appears as potential outlier (with respect to the first and second PC) and thus warrants further investigation. For PC 2, Q-residuals indicate that samples located in the lower right corner are poorly represented. Thus, the corresponding loadings might provide limited information about these samples.

In some cases, exploring a non-linear embedding using a single PC at the time is too complex, which is especially true for data sets with a flat Eigenvalue distribution. We here propose relative Hotelling's T^2 contributions (see Eq. (15)) in order to include information from all relevant PCs when deriving explanations for why pairs of data points or clusters differ from each other. Figure 10 shows the relative T^2-contributions corresponding to the difference between the clusters A and B shown in Figure 9. The results indicate that the two clusters differ mostly by the value of variable Y8, which does not become evident from the color-coded UMAP shown in Figure 6. This is because the absolute difference between the two distributions is small compared to the total variance of the variable across the entire data set (Figure 10b). Altogether, this example underpins the strength of analyzing a UMAP-type



(a) Cluster A (b) Cluster B

Figure 9: Cluster selection for computation of relative T^2-contributions.

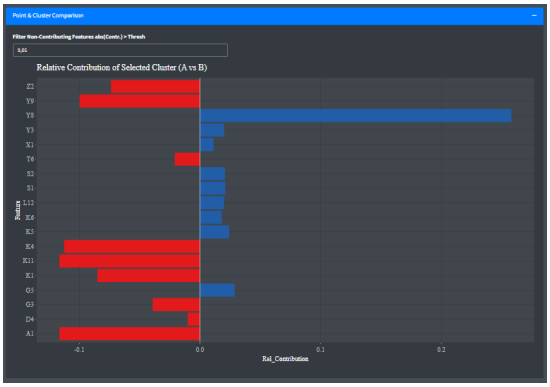
embedding by means of relative T^2-contributions.

4. Discussion

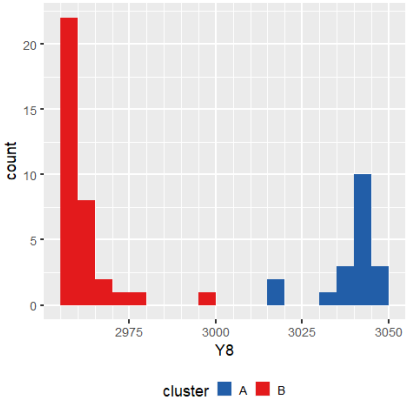
We proposed a novel workflow that employs the well-known Hotelling's T^2 statistic and Q-residuals from PCA to derive explanations for non-linear neighbor embedding such as UMAP. We integrated these diagnostic tools in a software prototype using the shiny web application framework for R and demonstrated how they can be used for exploratory analysis of a multivariate dataset with UMAP. In particular, we showed how our approach is capable of identifying discriminatory features between groups of data points that remain unnoticed when exploring NE using standard univariate (variable-by-variable) or multivariate (using PC-wise color-coding) approaches which underpins the strength of the approach.

5. Conclusion

Non-linear dimension reduction techniques such as UMAP often provide valuable insights into high-dimensional data sets beyond those obtained by PCA. However, interpretation of the former is not as straight forward as with the latter and thus requires a downstream (confirmatory) data analysis pipeline. We found previously proposed methodology to "open the black box



(a) Relative T^2-contributions.



(b) Histogram of variable Y8

Figure 10: Comparison clusters A and B by means of relative Hotelling's T^2-contributions

of neighbor embedding" limited in terms of both, user-friendliness and computational efficiency that would allow non-expert users to "make sense" out of NE in a consistent way. Our approach addresses both these shortcomings.

Acknowledgements

Funding was provided by the Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK), the Federal Ministry for Digital and Economic Affairs (BMDW), and the Province of Upper Austria in the frame of the COMET - Competence Centers for Excellent Technologies programme managed by the Austrian Research Promotion Agency FFG and the COMET Center CHASE.

References

[1] L. Van der Maaten, G. Hinton, Visualizing data using t-sne., Journal of machine learning research 9 (11) (2008).

[2] W. Li, J. E. Cerise, Y. Yang, H. Han, Application of t-sne to human genetic data, Journal of bioinformatics and computational biology 15 (04) (2017) 1750017.

[3] H. Chang, D.-Y. Yeung, Y. Xiong, Super-resolution through neighbor embedding, in: Proceedings of the 2004 IEEE Computer Society Confer-

ence on Computer Vision and Pattern Recognition, 2004. CVPR 2004., Vol. 1, IEEE, 2004, pp. I–I.

[4] P. Hajibabaee, F. Pourkamali-Anaraki, M. A. Hariri-Ardebili, An empirical evaluation of the t-sne algorithm for data visualization in structural engineering, arXiv preprint arXiv:2109.08795 (2021).

[5] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, arXiv preprint arXiv:1802.03426 (2018).

[6] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. B. Viégas, M. Wattenberg, Embedding projector: Interactive visualization and interpretation of embeddings, arXiv preprint arXiv:1611.05469 (2016).

[7] A. Chatzimparmpas, R. M. Martins, A. Kerren, t-visne: Interactive assessment and interpretation of t-sne projections, IEEE transactions on visualization and computer graphics 26 (8) (2020) 2696–2714.

[8] A. Bibal, V. M. Vu, G. Nanfack, B. Frénay, Explaining t-sne embeddings locally by adapting lime., in: ESANN, 2020, pp. 393–398.

[9] E. Andries, R. Nikzad-Langerodi, Dual-constrained and primal-constrained principal component analysis, Journal of Chemometrics (2022) e3403.

[10] Y. Wang, H. Huang, C. Rudin, Y. Shaposhnik, Understanding how dimension reduction tools work: an empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization, arXiv preprint arXiv:2012.04456 (2020).

[11] I. Jolliffe, Principal component analysis, Encyclopedia of statistics in behavioral science (2005).

[12] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometrics and intelligent laboratory systems 2 (1-3) (1987) 37–52.

[13] A. Ferrer, Multivariate statistical process control based on principal component analysis (mspc-pca): Some reflections and a case study in an autobody assembly process, Quality Engineering 19 (4) (2007) 311–325.

[14] S. Wold, M. Sjöström, Simca: a method for analyzing chemical data in terms of similarity and analogy, ACS Publications, 1977.

[15] R. Nikzad-Langerodi, W. Zellinger, S. Saminger-Platz, B. A. Moser, Domain adaptation for regression under beer–lambert's law, Knowledge-Based Systems 210 (2020) 106447.

[16] W. Chang, J. Cheng, J. Allaire, Y. Xie, J. McPherson, et al., Shiny: web application framework for r, R package version 1 (5) (2017) 2017.