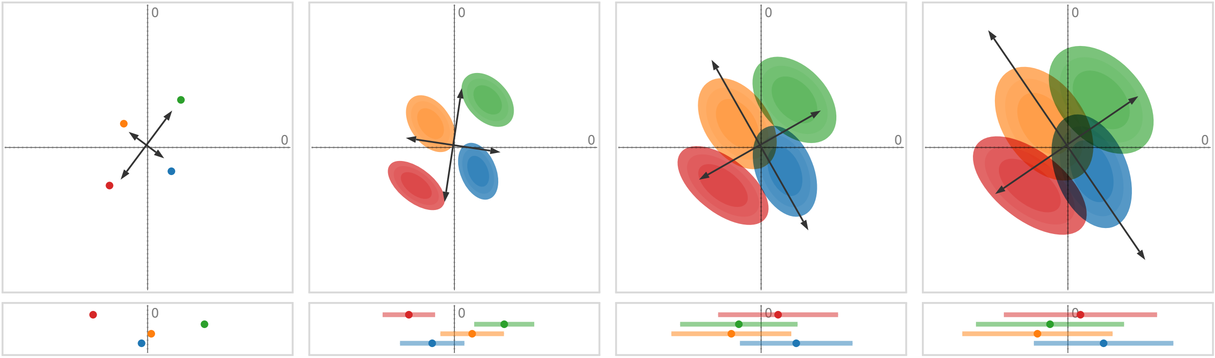
Uncertainty-Aware Principal Component Analysis

Jochen Görtler, Thilo Spinner, Dirk Streeb, Daniel Weiskopf, and Oliver Deussen



(a) Regular PCA (b) s = 0.5 (c) s = 0.8 (d) s = 1.0

Fig. 1. Data uncertainty can have a significant influence on the outcome of dimensionality reduction techniques. We propose a generalization of principal component analysis (PCA) that takes into account the uncertainty in the input. The top row shows the dataset with varying degrees of uncertainty and the corresponding principal components, whereas the bottom row shows the projection of the dataset, using our method, onto the first principal component. In Figures (a) and (b), with relatively low uncertainty, the blue and the orange distributions are comprised by the red and the green distributions. In Figures (c) and (d), with a larger amount of uncertainty, the projection changes drastically: now the orange and blue distributions encompass the red and the green distributions.

Abstract—We present a technique to perform dimensionality reduction on data that is subject to uncertainty. Our method is a generalization of traditional principal component analysis (PCA) to multivariate probability distributions. In comparison to non-linear methods, linear dimensionality reduction techniques have the advantage that the characteristics of such probability distributions remain intact after projection. We derive a representation of the PCA sample covariance matrix that respects potential uncertainty in each of the inputs, building the mathematical foundation of our new method: *uncertainty-aware PCA*. In addition to the accuracy and performance gained by our approach over sampling-based strategies, our formulation allows us to perform sensitivity analysis with regard to the uncertainty in the data. For this, we propose *factor traces* as a novel visualization that enables to better understand the influence of uncertainty on the chosen principal components. We provide multiple examples of our technique using real-world datasets. As a special case, we show how to propagate multivariate normal distributions through PCA in closed form. Furthermore, we discuss extensions and limitations of our approach.

Index Terms—Uncertainty, dimensionality reduction, principal component analysis, linear projection, machine learning

1 Introduction

Dimensionality reduction techniques can be applied to visualize data with more than two or three dimensions, projecting the data to a lower-dimensional subspace. These projections should be meaningful so that the important properties of the data in the high-dimensional space can still be reconstructed in the low-dimensional representation. In general, dimensionality reduction techniques can either be linear or non-linear [21]. Linear dimensionality reduction has the advantage that properties and invariants of the input data are still reflected in the resulting projections. Another advantage of linear over non-linear methods is that they are easier to reason about because the subspace for the projection is always a linear combination of the original axes. Also, linear methods are usually efficient to implement [4].

Arguably the most frequently used linear method is principal component analysis (PCA). It is most effective if the dimensions of the input data are correlated, which is common. PCA uses this property and finds the directions of the data that contain the largest variance. This is achieved by performing eigenvalue decomposition of the sample covariance matrix that is estimated from the input. The input to conventional PCA is a set of points. However, we often encounter data afflicted with uncertainty or variances. According to Skeels et. [31], there are several sources for this uncertainty. Measurement errors might arise from imperfect observations. In many cases, we rely on data that is the output of predictive models or simulations that provide probabilistic estimations. And lastly, uncertainty is inevitable when aggregating data, as some of the original information has to be discarded. The natural way to model these instances of uncertain data is by using probability distributions over possible realizations of the data.

In this paper, we derive a generalization of PCA that directly works on probability distributions. Like regular PCA, our new method of *uncertainty-aware PCA* solely requires that the expected value and the covariance between dimensions of these distributions can be determined—no higher-order statistics are taken into account. This uncertainty of the input data can have a strong impact on the resulting projection, because it directly influences the magnitude of the eigenvalues of the sample covariance matrix.

In addition to extending PCA, we introduce *factor traces* as a vi-

- *J. Görtler, T. Spinner, D. Streeb, and O. Deussen are with the University of Konstanz, Germany. E-mail: {firstname.lastname}@uni-konstanz.de*
- *D. Weiskopf is with the University of Stuttgart, Germany. E-mail: weiskopf@visus.uni-stuttgart.de*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

sualization that shows how the projections of the original axes onto the subspace change with a varying degree of uncertainty. This enables to perform a sensitivity analysis of the dimensionality reduction with respect to uncertainty and gives an interpretable representation of the linear projection that is performed. Our paper has four main contributions:

- a closed-form generalization of PCA for uncertain data,
- sensitivity analysis of PCA with regards to uncertainty in the data,
- factor traces as a new visualization technique for the sensitivity of linear projections, and
- establishing a distance metric between principal components.

In Figure 1, we compare our method to regular PCA and illustrate why it is important to consider the uncertainty in the data when determining the projection: it shows a projection of four bivariate probability distributions, each with varying levels of uncertainty, that are projected onto a single dimension. For input with low uncertainty, the red and green data points define the extent of the projected data. With increasing uncertainty and due to the shape of the underlying distributions, the projection looks quite different: now the orange and blue data points mark the extent of the projected data. This change in the projection shows that it is important to incorporate the uncertainty information adequately into our dimensionality reduction algorithms. Although all distributions in this example are Gaussian, our method works on any probability distribution for which the expected value and the covariance can be determined.

2 Related Work

The survey by Nonato and Aupetit [21] offers a broad overview of dimensionality reduction from a visualization perspective. Principal component analysis [23] is one of the oldest and most popular techniques. It is often applied to reduce data complexity, which is a common task in visualization. By construction, PCA yields the linear projection that retains the most variance of the input data in the lower-dimensional subspace. Probabilistic PCA [36] extends traditional PCA by adding a probabilistic distribution model. In contrast to our method, an unknown isometric measurement error is assumed. Likewise, many extensions have been introduced to PCA [3, 16]. For example, Kernel PCA [28] enables non-linear projections by first transforming objects into a higher-dimensional space in which a good linear projection can be found. Techniques such as Bayesian PCA [2,20,22], and the method introduced by Sanguinetti et al. [26] focus on estimating the dimensionality of the lower-dimensional space. Robust PCA methods [1,39,40] target datasets with outliers. Different extensions to PCA have also been developed in the context of fuzzy systems. The technique described by Denoeux and Masson [5] applies PCA to fuzzy numbers by training an artificial neural network that incorporates the different possible realizations for each fuzzy number. Giordani and Kiers [10] provide an overview of methods that can be used to apply PCA to interval data. In contrast, we extend traditional PCA to an uncertainty-aware linear technique for exploratory visualization that works on general probability distributions.

Next to PCA, Factor Analysis (FA) [32] is a well known linear method. Its goal is to identify (not necessarily orthogonal) latent variables underlying a higher-dimensional space of measurements. Factor Analysis models measurement errors, yet constraining the errors to be uncorrelated is common. One reason for this is that modeling correlated errors can be problematic if the actual errors are unknown [11]. In our description, we assume that all errors are known, or can at least be estimated. Many other linear techniques such as Classical Multi-Dimensional Scaling [38] and Independent Component Analysis [13] are covered by Cunningham and Ghahramani [4]. To the best of our knowledge, none of them can deal with data that has explicitly encoded (measurement) errors.

Liu et al. [19] provide an overview of the visualization and exploration of high-dimensional data. The Star Coordinates [15] visualization technique, for example, provides interactive linear projections of high-dimensional data. Recently measure-driven approaches for exploration have gained interest, e.g., by Liu et al. [18] as well as by Lehmann and Theisel [17]. Visualizing the projection matrix of linear dimensionality reduction techniques (instead of projections of the data) can be done with factor maps or Hinton diagrams [2, 12].

Advances in visualizing uncertainty and errors often originate from the need to represent prediction results [33]. More generally, visualizing Gaussian distributions by a set of isolines is a common practice. In this paper, we aim at bringing uncertainty-aware dimensionality reduction and visualization together. For example, our technique can be used to extend Wang et al.'s [42] approach to visualizing large datasets by allowing a fast approximate visualization of clusters. Furthermore, correlated probability distributions are often the result of Bayesian inference, which is widely used in prediction tasks, where the result is always a probability distribution. In this domain, Gaussian processes [24] are a prime example of correlated uncertainty.

Lately, there has been a push in the visualization community to gain a better understanding of the intrinsic properties of projection methods. However, the focus mainly has been on exploring non-linear approaches. For instance, Schulz et al. [27] propose a projection for uncertainty networks based on sampling different realizations of the data and investigate potential effects of uncertainty. With *DimReader*, Faust et al. [8] address the problem of explaining non-linear projections. Their technique uses automatic differentiation to derive a scalar field that encodes the sensitivity of the projected points against perturbations. Wattenberg et al. [43] examine how the choice of parameters affects the projection results of t-SNE. Similarly, Streeb et al. [35] compare a sample of (non-)linear techniques and influences of their parameters on projections.

3 Statistical Background

The typical way to model uncertainty is by using probability distributions over the data domain. This approach is well established in other fields, such as measurement theory and Bayesian statistics. Before getting to the gist of our method, we want to give a quick overview of the statistical background we need for our technique. More details can be found in the textbook by Wickens [44].

3.1 Random Variables and Random Vectors

A *random variable* is used to describe the values of possible outcomes x of a random phenomenon. It is usually defined as a real-valued scalar $x \in \mathbb{R}$. Probability distributions are used to assign a probability (density) to each outcome of the random variable—both concepts are closely tied together. To extend this one-dimensional case to multi-dimensional phenomena, we can group several random variables into a multivariate random variable, which is also called a *random vector*. Throughout this article, we denote random vectors by $\mathbf{x} = (x_1, \ldots, x_d)^\mathsf{T}$, with $\mathbf{x} \in \mathbb{R}^d$. Analogously, the corresponding multivariate probability distributions span the same d-dimensional domain. An interesting property arises from the fact that $\mathbf{x}$ can be viewed as a position vector: it can be manipulated using affine transformations. These transformations can, for example, be used to scale, translate, or rotate $\mathbf{x}$. Generally, an affine transformation has the form $\mathbf{y} = A\mathbf{x} + \vec{b}$. It consists of a linear transformation A and a translation vector $\vec{b}$ that together transform an input $\mathbf{x}$ to obtain a new random vector $\mathbf{y}$, which can be described using a modified distribution.

3.2 Summary Statistics

For many applications, it is helpful to summarize the probability distributions into simpler, yet characteristic quantities. Ideally, these simple terms still allow us to make statements about the shape and properties of the original distribution. Such descriptions are called summary statistics. The most well-known statistics are the first and second moments, which, in the real-valued case, are also called *mean* and *variance*. They are used to describe the center of gravity and the spread of a distribution. For multi-dimensional data, the mean is a d-dimensional vector, and the variance is replaced by the *covariance* that also reflects correlations between each of the d components. Because the covariance describes these relationships, it has the form of a symmetric $d \times d$ matrix. Every covariance matrix is always positive semi-definite—we provide a detailed discussion in the appendix.

For some distributions, these two summary statistics are explicitly defined. The multivariate normal (MVN) distribution, which is widely used in many domains, has an interesting property—it is completely determined by its first and second moments. Therefore, if $\mathbf{x}$ follows an MVN distribution, with mean μ and covariance matrix Ψ, we write:

$$\mathbf{x} \sim N(\mu, \Psi).$$

Sometimes our random vector $\mathbf{x}$ is given by a set of samples $\{\vec{x}_n\}, n \in \{1, \ldots, N\}$ from an arbitrary distribution. Given this set, we can estimate the first and second moments of $\mathbf{x}$ using the sample mean $\mu_{\mathbf{x}}$, which is defined in terms of the expected value $\mathbb{E}[\cdot]$:

$$\mu_{\mathbf{x}} = \mathbb{E}[\mathbf{x}] = \frac{1}{N} \sum_{n=1}^{N} \vec{x}_n$$

and the sample covariance matrix $\mathrm{Cov}(\mathbf{x}, \mathbf{x})$:

$$\begin{aligned} \mathrm{Cov}(\mathbf{x}, \mathbf{x}) &= \mathbb{E}\left[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^{\mathsf{T}}\right] \\ &= \mathbb{E}\left[\mathbf{x}\mathbf{x}^{\mathsf{T}}\right] - \mu_{\mathbf{x}}\mu_{\mathbf{x}}^{\mathsf{T}} \end{aligned} \tag{1}$$

The term $\mathbb{E}\left[\mathbf{x}\mathbf{x}^{\mathsf{T}}\right]$ is the expected outer product $\mathbf{x}\mathbf{x}^{\mathsf{T}}$ and can be approximated as follows:

$$\mathbb{E}\left[\mathbf{x}\mathbf{x}^{\mathsf{T}}\right] = \frac{1}{N} \sum_{n=1}^{N} \vec{x}_n \vec{x}_n^{\mathsf{T}}$$

In the previous section, we explained how to transform a random vector $\mathbf{x}$ using affine transformations. Transforming $\mathbf{x}$ in this way also influences the summary statistics. For the mean, it holds that:

$$\mathbb{E}\left[A\mathbf{x} + \vec{b}\right] = A\mathbb{E}[\mathbf{x}] + \vec{b}$$

In a similar fashion, we can transform the covariance matrix:

$$\mathrm{Cov}(A\mathbf{x} + \vec{b}, A\mathbf{x} + \vec{b}) = A\,\mathrm{Cov}(\mathbf{x}, \mathbf{x})A^{\mathsf{T}} \tag{2}$$

Both equations follow from the linearity of the expected value operator $\mathbb{E}[\cdot]$. Intuitively, only the mean of $\mathbf{x}$ is influenced by the translation $\vec{b}$. The covariance matrix, in contrast, is invariant to translation. The reason for this is that the covariance only captures the relative variance of each component because it is always centered around the sampling mean by the term $\mu_{\mathbf{x}}\mu_{\mathbf{x}}^{\mathsf{T}}$. In the following section, we will use these above definitions to formulate our method.

4 Method

We have motivated the different causes of uncertainty in the input data in the introduction. In this part, we describe the necessary adaptions to the framework of PCA that are required to handle uncertainty, as modeled in the previous section. We will first show how to adapt the computation of the covariance matrix to work on probability distributions, which is a fundamental part of our technique. Then, we will describe how this fits into the context of regular PCA. Afterward, we will demonstrate how our method allows us to perform PCA analytically on uncertain data, using multivariate normal distributions as an example. Finally, we will show that our approach is a generalization of regular PCA. This allows us to combine certain and uncertain data within the same mathematical framework and provides us with the foundation for sensitivity analysis, as described in Section 5.

4.1 Model

PCA is used to find the directions of the data with the largest variance by looking at the covariance of the input. We adopt this concept to arbitrary distributions to handle uncertain data. For our method, we only require that the expected value and the covariance can be determined for each of the distributions. It is important to note that this does not imply that the input distributions necessarily have to follow a Gaussian distribution. We want to illustrate this for a small example: let us consider an input distribution made up of two clusters spread about its mean. Then, the covariance of the distribution still captures the spread of the data, namely along the direction of the location of the two clusters. So even though the distribution might not be sufficiently described only by mean and covariance, its overall extent is still represented adequately using these first- and second-order statistics. In Section 6.3, we will show an example of a dataset that exhibits this property. And in Section 8.3, we will discuss its implications on the resulting projection.

It is important that there is an established relationship between the units of the original axes for PCA to yield a meaningful result. The usual approach to achieve this is to normalize the input data accordingly. The same preprocessing step needs to be performed for our method. For probability distributions, this can be performed using affine transformations, as outlined above.

4.2 Uncertain Covariance Matrices

As we have mentioned before, the goal of our method is to perform PCA on a set of N probability distributions that are used to model the uncertainty, as described in Section 3. Formally, we represent this collection of distributions as random vectors $\mathbf{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$. For each of these random vectors, we require that we can determine its expected value $\mathbb{E}[\mathbf{t}_i]$ and its pairwise covariance $\mathrm{Cov}(\mathbf{t}_i, \mathbf{t}_i)$. It is important to note that $\mathbf{T}$ can conceptually be interpreted as a random vector of second order, as its components $\mathbf{t}_n$ are random vectors themselves.

Our approach adapts the computation of the covariance matrix to account for uncertainty in the data. Regular PCA works on a set of points. Therefore, the covariance matrix can be understood as the computation of the expected products of deviations of these points from the sample mean. In contrast, our approach works on a set of random vectors, which changes the problem in the following way: Because of the uncertainty in the data, we do not know the actual deviation of each random vector from the overall sample mean. But we can determine the deviation that is to be expected for each of the distributions. We do this conceptually by integrating over the deviation of all possible realizations of each probability distribution. In the framework of PCA, where only the first- and second-order moments are taken into account, it turns out we do not even have to evaluate this integral: we can derive the covariance matrix directly from the summary statistics.

From Equation 1, we can derive a property of the covariance matrix that we will need later on: it gives us a way to compute the expected outer product $\mathbb{E}\left[\mathbf{x}\mathbf{x}^{\mathsf{T}}\right]$ of a particular random vector with itself. We achieve this by solving Equation 1 for $\mathbb{E}\left[\mathbf{x}\mathbf{x}^{\mathsf{T}}\right]$:

$$\mathbb{E}\left[\mathbf{x}\mathbf{x}^{\mathsf{T}}\right] = \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]^{\mathsf{T}} + \mathrm{Cov}(\mathbf{x}, \mathbf{x}) \tag{3}$$

For distributions, we use the following equation, which is akin to computing the expected products of *expected* deviations. To avoid confusion with the expected value of each random vector $\mathbb{E}[\cdot]$, we denote the expectation operator that stems from the covariance method with $\hat{\mathbb{E}}[\cdot]$:

$$\mathrm{Cov}(\mathbf{T}, \mathbf{T}) = \hat{\mathbb{E}}\left[\mathbb{E}\left[\mathbf{T}\mathbf{T}^{\mathsf{T}}\right] - \mu_{\mathbf{T}}\mu_{\mathbf{T}}^{\mathsf{T}}\right]$$

We can expand this further by making use of Equation 3:

$$\mathrm{Cov}(\mathbf{T}, \mathbf{T}) = \hat{\mathbb{E}}\left[\mathbb{E}[\mathbf{T}]\mathbb{E}[\mathbf{T}]^{\mathsf{T}} + \mathrm{Cov}(\mathbf{T}, \mathbf{T}) - \mu_{\mathbf{T}}\mu_{\mathbf{T}}^{\mathsf{T}}\right]$$

$$\boxed{\mathrm{Cov}(\mathbf{T}, \mathbf{T}) = \hat{\mathbb{E}}\left[\mathbb{E}[\mathbf{T}]\mathbb{E}[\mathbf{T}]^{\mathsf{T}}\right] + \hat{\mathbb{E}}[\mathrm{Cov}(\mathbf{T}, \mathbf{T})] - \mu_{\mathbf{T}}\mu_{\mathbf{T}}^{\mathsf{T}}} \tag{4}$$

The different terms in Equation 4 have particular interpretations. First, we recognize that the term $\hat{\mathbb{E}}\left[\mathbb{E}[\mathbf{T}]\mathbb{E}[\mathbf{T}]^{\mathsf{T}}\right]$ is the same as performing regular PCA on the means of each of the distributions. The second term $\hat{\mathbb{E}}[\mathrm{Cov}(\mathbf{T},\mathbf{T})]$ computes the average covariance matrix over all random vectors:

$$\hat{\mathbb{E}}[\mathrm{Cov}(\mathbf{T},\mathbf{T})] = \frac{1}{N}\sum_{i=1}^{N}\mathrm{Cov}(\mathbf{t}_i,\mathbf{t}_i) \qquad (5)$$

It reflects the uncertainty that each random vector has and how these uncertainties influence the overall covariance in the dataset—it is also the major difference between our method and regular PCA, which cannot handle probability distributions. The last term is called centering matrix and also part of regular PCA. It consists of the outer product of the empirical mean $\mu_{\mathbf{T}}$ of our dataset. The empirical mean of our dataset can be computed as follows:

$$\mu_{\mathbf{T}} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[\mathbf{t}_i]$$

Algorithm 1 provides the corresponding pseudocode for Equation 4. The proof that Equation 4 yields a symmetric, positive semi-definite matrix and therefore is an actual covariance matrix can be found in the Appendix of this document.

4.3 PCA Framework and Diagonalization

Now that we have constructed the covariance matrix while respecting the uncertainty, we can continue with the remaining steps of the PCA algorithm. After setting up the covariance matrix, we retrieve its eigenvalues λ_d and corresponding eigenvectors $\vec{v}_d$. This can be done using *eigenvalue decomposition*:

$$\mathrm{Cov}(\mathbf{T},\mathbf{T})\vec{v} = \lambda\vec{v}$$

Let q be the desired target number of dimension for our dimensionality reduction. We then choose the q largest $\vec{v}_d$ by their corresponding eigenvalue λ_d, yielding q principal components $\mathbf{W} = \{\mathbf{w}_1,\ldots,\mathbf{w}_q\}$. We can then project each distribution onto the subspace $\langle\mathbf{W}\rangle$ that is spanned by these principal components $\Phi(\mathbf{t}_n) \in \langle\mathbf{W}\rangle$, where $\Phi(\cdot)$ is a linear projection that can be described using a linear transformation.

It is important to note that eigenvalues and eigenvectors have certain characteristics that complicate their analysis. The orientation of $\vec{v}_d$ is not completely defined, therefore $\vec{v}_d \hat{=} -\vec{v}_d$. In practice, the computation of $(\lambda_d,\vec{v}_d)$ is performed numerically, which can lead to instabilities and rounding errors. We will discuss the impact of this on the analysis of linear projections in Section 5.

4.4 Linear Transformation of MVN Distributions

Now that we have defined the projection $\Phi(\cdot)$, we need to transform each distribution into the subspace $\langle\mathbf{W}\rangle$. In the following, we will describe how this can be carried out for multivariate normal distributions, as they are often used to model errors or uncertainty in the data. As mentioned in Section 2, several existing techniques already model uncertainty using MVN distributions. In these works, the distributions are usually described using an error model, which means that a measurement $\vec{x}$ is disturbed by an error term ε. This is commonly written as:

$$\mathbf{t}_n = \vec{x}_n + \varepsilon_n, \quad \varepsilon_n \sim N(\vec{0},\Psi_n)$$

To retrace the closed-form derivation of the covariance matrix that we described in Section 4.2, it is easier to think of this error in terms of a single random vector $\mathbf{t}_n$ that can be equivalently defined as follows:

$$\mathbf{t}_n \sim N(\vec{x}_n,\Psi_n).$$

Figure 2 shows examples of different error models that can be created depending on the shape of Ψ_n. We also visualize the corresponding principal components of the dataset, determined by using our method.

The dimensionality of $\Phi(\mathbf{t})$ is $\dim(\langle\mathbf{W}\rangle)$. To perform the actual projection, we assume that $\mathbf{w}_q$ are unit vectors, and write them in a column matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix}\mathbf{w}_1 \ldots \mathbf{w}_q\end{bmatrix}$$

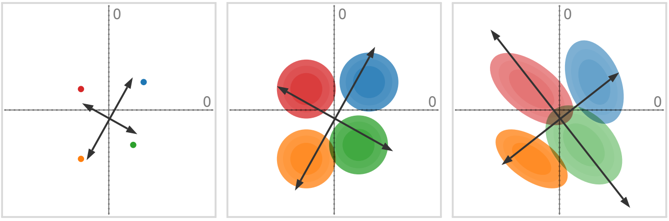
It is important to note that a projection Φ is an affine transformation, as defined in Section 3. Accordingly, we can project a normal distribution as follows:

$$\Phi(\mathbf{t}_n) = N(\mathbf{A}^{\mathsf{T}}\mu_n,\, \mathbf{A}^{\mathsf{T}}\Psi_n\mathbf{A})$$

The resulting distribution remains multivariate normally distributed.

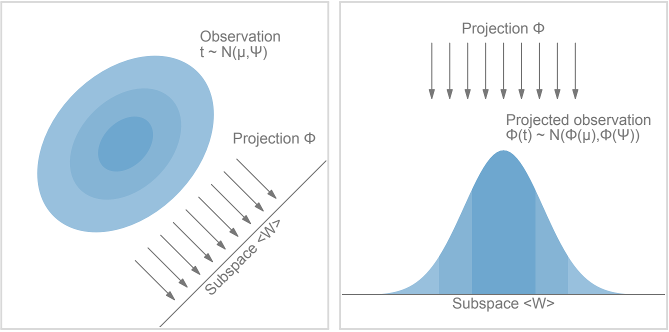
Algorithm 1: Covariance matrix of random vectors

```
Input  : List of $d$-variate distributions $\mathbf{T}$, scaling factor $s = 1$
Output : Covariance matrix $K_{\mathbf{TT}}$
$\mu_{\mathbf{t}} \leftarrow d$-dimensional vector initialized to 0
foreach $\mathbf{t} \in \mathbf{T}$ do
    $\mu_{\mathbf{t}}$ += $\mathbf{t}.mean()$
end
$\mu_{\mathbf{t}}$ /= $\mathbf{T}.length()$
$K_{\mathbf{TT}} \leftarrow d \times d$ matrix initialized to 0
foreach $\mathbf{t} \in \mathbf{T}$ do
    $\vec{m} \leftarrow \mathbf{t}.mean()$
    $K_{\mathbf{TT}}$ += $\vec{m}\vec{m}^{\mathsf{T}} + s^2 \cdot \mathbf{t}.cov() - \mu_{\mathbf{t}}\mu_{\mathbf{t}}^{\mathsf{T}}$
end
$K_{\mathbf{TT}}$ /= $\mathbf{T}.length()$
return $K_{\mathbf{TT}}$
```



(a) (b) (c)

Fig. 2. Different types of input data: (a) Regular PCA without uncertainty. (b) Isometric error model as used by previous work where PCA has been described as an optimization problem; the directions of the principal components are the same, but the lengths differ. (c) Our method: it works on arbitrary distributions and can result in drastically different principal components.



High-dimensional space Low-dimensional subspace

Fig. 3. A linear projection $\Phi(\cdot)$ of a normal distribution $\mathbf{t} \sim N(\mu,\Psi)$ results in a modified multivariate normal distribution $\Phi(\mathbf{t})$ in the lower dimensional subspace $\langle\mathbf{W}\rangle$. Because of this, we can propagate the uncertainty directly through linear dimensionality reduction techniques.

4.5 Reduction to Regular PCA

In this section, we will show that our method is a mathematical generalization of conventional PCA. The main difference between the two algorithms lies in the setup of the covariance matrix, as described by

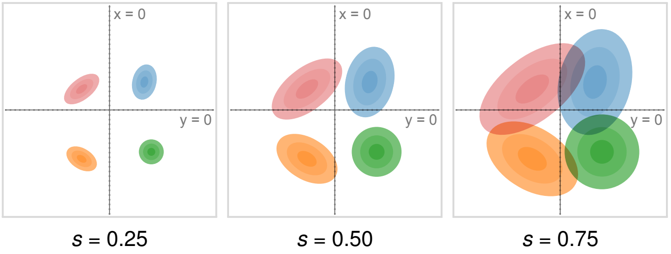


Fig. 4. Different levels of uncertainty can be achieved by scaling the covariances of each distribution with a factor s. By letting $s \to 0$ we can emulate traditional principal component analysis, as the distributions converge toward single points.

Equation 4. Our method includes an additional term that reflects the uncertainty of each input (Equation 5). To reduce our formulation to regular PCA, we will scale the covariance of each of the distributions by a constant factor s. This decreases the spread of the covariance matrix, and because of this, implicitly reduces the amount of uncertainty within each distribution.

To scale the covariance matrices, we will again make use of the properties of affine transformations for covariance matrices, as discussed in Section 3. Let S be a scale matrix that has the form $S = \operatorname{diag}(s)$. We can now use Equation 2 to scale $K_{\mathbf{TT}} = \hat{\mathbb{E}}[\operatorname{Cov}(\mathbf{T},\mathbf{T})]$:

$$S\,(K_{\mathbf{TT}})\,S^{\mathsf{T}}$$

In practice, we can make use of the fact that a scale matrix S is always a diagonal matrix. In our case, each diagonal entry is equal to s, therefore $S = \operatorname{diag}(s)$. This allows us to simplify the equation above even further:

$$\boxed{S\,(K_{\mathbf{TT}})\,S^{\mathsf{T}} = s^2 \cdot K_{\mathbf{TT}}} \tag{6}$$

Figure 4 shows a set of multivariate normal distributions, all scaled with different weights. Another property of this description is that we can use s to interpolate between the certain and uncertain representation of our data. Algorithm 1 shows how to incorporate the scaling factor into the computation of the covariance matrix. In the next section, we will use this fact to investigate how much the uncertainty influences the resulting projection.

5 Sensitivity Analysis

We have shown in previous sections that uncertainty in the input can have a strong influence on the resulting set of principal components. Therefore, to better understand this relationship, we investigate to what amount the dimensionality reduction depends on the shape of each of the probability distributions. In Section 4.5, we have shown that our method is a generalized formulation of conventional PCA. We achieved this by scaling the covariances of each distribution with a factor s that describes the importance of the uncertainty. Now, we will leverage this model to show how the fitted projection varies for different scaling factors in the interval $s \in [0, \infty)$. This interval can be split up in two parts to investigate two different scenarios. For $0 \le s \le 1$, we can interpolate between uncertainty-aware PCA and regular PCA. Conversely, by choosing $1 < s < \infty$ we can extrapolate what the projection would look like if the uncertainty were higher. In the following, we propose a novel visualization technique that is tailored to analyze the effects of different scaling factors s and hence influences of different levels of uncertainty.

5.1 Factor Traces

Factor Analysis shares many similarities with PCA and is often used for the explanatory analysis of multi-dimensional datasets. The individual latent factors, akin to principal components, are usually represented using *factor maps*. To create a factor map, the unit vectors of each dimension in feature space are projected according to the latent factors of the data [34]. We extend this technique to enable the exploration of the effects of uncertainty on PCA.

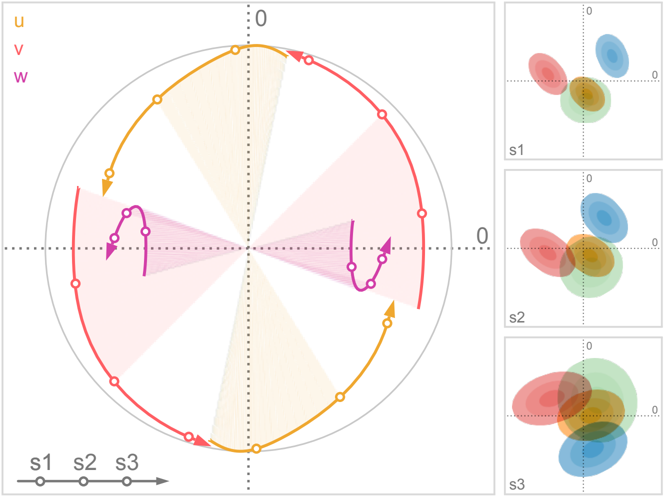


Fig. 5. Progression of a factor trace (left) as the uncertainty increases. Both unit vectors u and v rotate around w. Accordingly, the projected distributions (right) rotate around the mean as well.

Factor maps visualize static latent information that is hidden in the input data. However, we are interested in visualizing the progression of uncertainty. We do this by looking at the *factor traces* that are described by the change of principal components under the varying degree of uncertainty. In particular, we perform sensitivity analysis by continuously scaling the covariances of the distributions from the original dataset using s as a scaling factor, as described above. Figure 5 shows an example of factor traces of a three-dimensional dataset. For each $s \in [0, \infty)$ a different subspace is chosen. As a result, the projected unit vectors describe a trace in the image space. Thereby, we obtain a compact representation of the analogous transformation of the feature space coordinate system. As we mentioned before, there are two intervals for s that are of interest for the analysis of the sensitivity with respect to the uncertainty. The interval $0 \le s \le 1$ is highlighted by shading the area under the trace. In contrast, for the interval $1 < s < \infty$ we only show the trace to avoid visual clutter, and we use an arrowhead to represent $s \to \infty$.

In practice, we progressively sample s in the interval using a hyperbolic function. At the heart of principal component analysis is the decomposition of the covariance matrix into its eigenvalues and eigenvectors. This entails various challenges for the interpretation of the projection. While the eigenvectors of a positive semi-definite matrix are always orthogonal to each other, their orientation is ambiguous as their sign can change. In the resulting sequence, it can happen that the sign of $\vec{v}_i$ and $\vec{v}_{i+1}$ flips. This, in return, leads to a mirrored projection. We account for this in factor traces by providing both projections of the unit vectors of the original axes. For example, this becomes apparent when looking at the purple trace in Figure 5. We discuss the limitations of this approach in Section 8.

5.2 Interpretation

Factor traces simultaneously visualize different properties of the original dataset with respect to the corresponding projection: the length of each trace describes how strongly each original axis is affected by the uncertainty in the data, whereas the distance of each part of the trace to the center depicts the linear combination of the original unit vectors that define the projection. Factor traces also offer a way to analyze the robustness of the resulting projections with respect to uncertainty. The covariance matrices and the overall shape of the data determine the corresponding eigenvalues. Because the principal components are sorted by their eigenvalues and only the q largest eigenvalues are chosen, their respective values also have a large effect on the resulting projection. Figure 6 shows factor traces of two different datasets, together with plots of their eigenvalues. With an increasing s, sometimes the distance between two eigenvalues λ_i, λ_j decreases more and more. In some cases, it appears that the eigenvalues will cross, but instead, they will eventually start to move away from each other again. This effect closely resembles *avoided crossings*, a quantum phenomenon [41]. The reason for this effect is that two eigenvalues coalesce as they end up with the same length [29]. Eigenvalues that avoid crossing manifest in distinctive bumps in their corresponding eigenvalue plots, which can be seen in Figure 6d. The first dataset in Figure 6 does not contain any avoided crossings. By contrast, Figure 6c and Figure 6d show a three-dimensional dataset with two bumps (highlighted by the dotted lines). Avoided crossings make it difficult to reason about the behavior of the eigenvectors and consequently the resulting projection in these points. In some cases—Figure 6c, for example—we can observe sharp turns in the corresponding factor traces. Here, the avoided crossing is between λ_2 and λ_3.

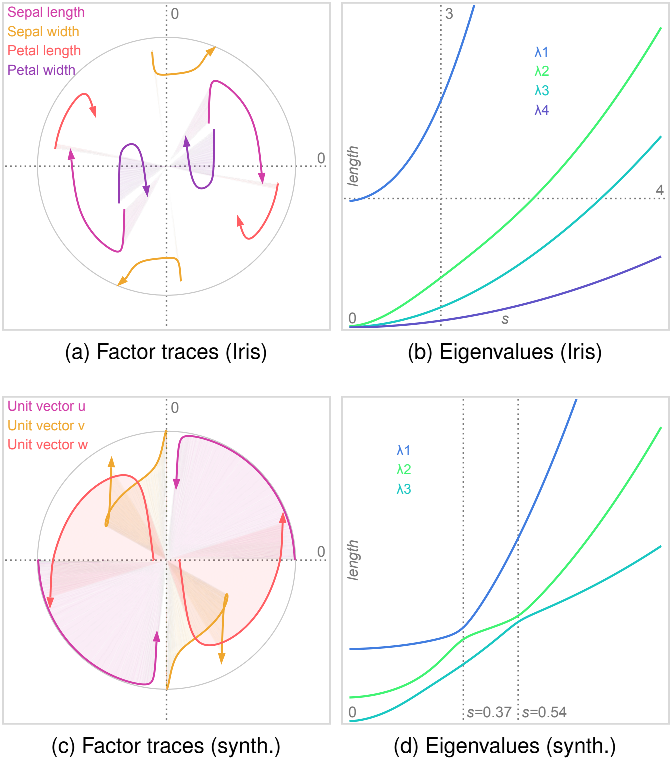
In conjunction with PCA, factor traces can aid the exploratory analysis of datasets by giving insights into the behavior of the principal components under uncertainty. Apart from showing how the projection changes under uncertainty, factor traces can help gauge how robust and hence how trustful the projected view of the dataset is. While our approach can aid in assessing projections, the visualization of high-dimensional data involving a large variety of distributions remains a difficult challenge. Generally, factor traces work well for datasets with up to six original dimensions. Above this limit, the representation becomes more difficult to understand due to overplotting. As shown in Figure 6, the interpretation of factor traces can be further enhanced by taking the corresponding eigenvalue plot into account. Depending on the dataset, we see the possibility to encode this information directly onto the factor trace, either by thickness or color.

6 Examples

Our method can handle various types of data uncertainty. Following the classification of Skeels et al. [31], we will take a look at examples from the measurement precision level and the completeness level. Measurement precision can play a substantial role in the analysis of datasets, especially for qualitative studies and experiments, where it is hard to assign certain values to responses. One way to deal with this uncertainty is to assign fuzzy numbers or even explicitly encoded probability distributions to each of the data points, as we will show in Section 6.1. Furthermore, we will look at different types of aggregations as sources for uncertainty on the completeness level. Apart from these examples, we see potential use cases for our method in visualizing preprocessed data for real-time analysis, or data that has been aggregated to protect the privacy of individuals, such as medical data. Regarding aggregation, Section 8 gives more details about the computational complexity of our approach. Please note that in the following examples, we use different representations for the distributions to highlight the projections found by our method.

6.1 Student Grades

Our uncertainty-aware PCA method can be used to perform dimensionality reduction on data with explicitly encoded uncertainty. Amongst others, such data can be found in the domain of fuzzy systems. As an example, we adopt the synthetic student grade dataset established by Denoeux and Masson [5]. It consists of four test results (*M1*, *M2*, *P1*, *P2*) for each of six students. The possible marks for the tests range from 0 to 20, and the dataset is highly heterogenous: grades can be represented either as real numbers, such as 15, without any uncertainty, or as intervals, such as $[10,12]$. Furthermore, many grades are given by qualitative statements like *fairly good* or *bad*. Both intervals and linguistic labels contain uncertainty, modeled using uniform distributions and trapezoidal distributions, respectively. The original paper also contains one *unknown* value. We model the missing value using a normal distribution $N(14, 5.7^2)$, which we extract from prior information: the mean is similar to previous test results, and the variance represents realistic deviations in both directions from this mean. Figure 7 shows the PCA on this dataset. It is important to note that PCA performed solely on the means of the input, as shown in Figure 7a, fails to capture important uncertainty information in the data. Our method (Figure 7b) appropriately depicts the uncertainty that is present in *P1* of *Tom* and *Bob*. This draws a very different picture from the result of regular PCA because the topology changes: it is quite possible that *Tom* performed similar to *Jane*—a fact that is not readily visible from Figure 7a. The importance of *P1* on the resulting projection can also be seen in the factor trace (Figure 7c) for this dataset: with an increasing amount of uncertainty factored into our method, the trace of *P1* moves toward the outside of the unit circle. The interpretation for this is that most of the information of this axis is preserved after projection.



(a) Factor traces (Iris)

(b) Eigenvalues (Iris)

(c) Factor traces (synth.)

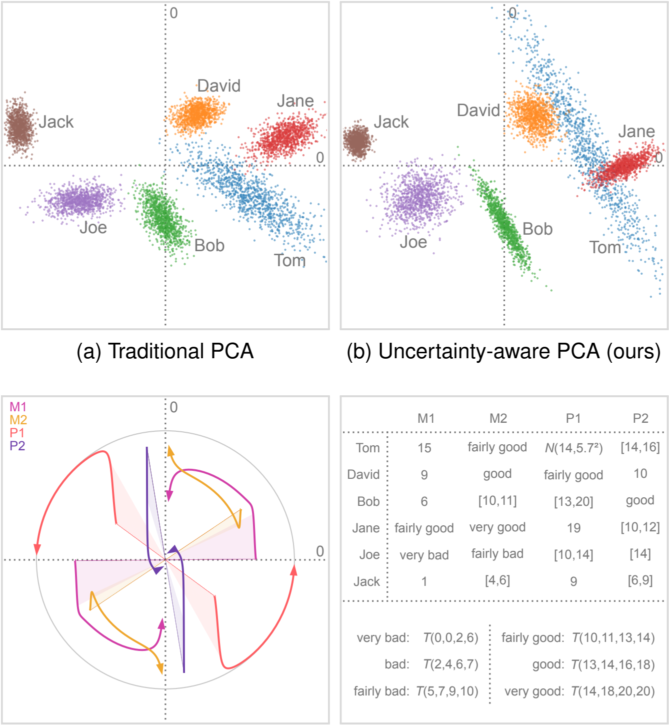
(d) Eigenvalues (synth.)

Fig. 6. Factor traces for two different datasets: (a) The 4D Iris dataset with (b) corresponding plot of the eigenvalues. (c) A 3D synthetic dataset, also with eigenvalues (d). Whereas the Iris dataset has no avoided crossing eigenvalues, the synthetic dataset has two avoided crossings (d) represented by bumps in the plot ($s \in \{0.37, 0.54\}$). The factor traces are projected onto a 2D subspace—as a consequence only the second bump manifests in the traces: at $s \approx 0.54$ the orange trace v forms a loop, while the purple trace u curves inward.

6.2 Iris Dataset

The Iris dataset[1] has widely been used to study projection and machine learning algorithms. It is four-dimensional and consists of 150 specimen of the Iris plants. Additionally, each instance can be attributed to one of three classes, and the instances are distributed equally among the classes. The clusters of the Iris dataset can be well described using multivariate normal distributions. We aggregate the data into three distributions, by their class label, on which we then perform uncertainty-aware PCA. The result of this can be seen in Figure 8a. For comparison, we also perform conventional PCA and color each point according to its class label—the results are shown in Figure 8b. Both projections are almost identical. This shows that our method can find projections with only a fraction of the original 4D data: three multivariate normal

[1] https://archive.ics.uci.edu/ml/datasets/iris



(a) Traditional PCA

(b) Uncertainty-aware PCA (ours)

(c) Factor traces

	M1	M2	P1	P2
Tom	15	fairly good	$N(14,5.7^2)$	[14,16]
David	9	good	fairly good	10
Bob	6	[10,11]	[13,20]	good
Jane	fairly good	very good	19	[10,12]
Joe	very bad	fairly bad	[10,14]	[14]
Jack	1	[4,6]	9	[6,9]

very bad:	T(0,0,2,6)	fairly good:	T(10,11,13,14)
bad:	T(2,4,6,7)	good:	T(13,14,16,18)
fairly bad:	T(5,7,9,10)	very good:	T(14,18,20,20)

(d) Dataset

Fig. 7. Importance of respecting the uncertainty in the input data. (a) Traditional PCA on the mean values of the student grades. (b) The projection found by our method shows the uncertainty in the data more faithfully. (c) The corresponding factor traces allow us to analyze the role of the original axes. In this case, *P1* approaches unit length, which means that its information is present even after projection. (d) The dataset in tabular form, providing the trapezoidal distributions for the linguistic labels. A trapezoidal distribution *T(a,b,c,d)* is defined by its bounds *a*, *d* and its discontinuities *b*, *c*.

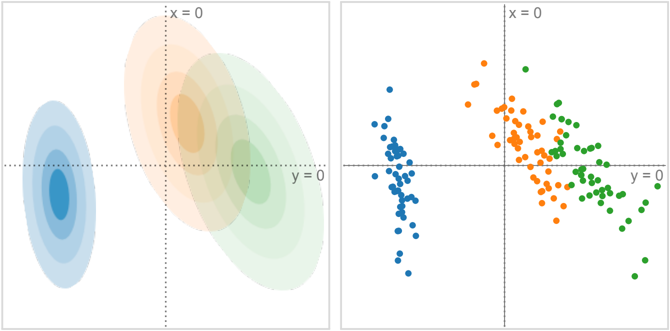
distributions instead of 150 points.

This example also illustrates two different ways to visualize data that has additional labels. To convey the class information, we need to support the visual aggregation of each cluster. When using conventional projection methods, this aggregation is usually performed in the image space. Figure 8b, for example, uses color. Another technique that is commonly used for aggregation in the image space is kernel density estimation. For clusters that roughly follow a normal distribution, our method provides a different approach: it allows aggregation in the feature space, where all the information is still present, and subsequent projection of the aggregated information. Subsequently, no further aggregation has to be performed in the image space. In Section 7, we provide a more detailed comparison to sampling-based strategies.

Figure 6a shows the factor traces for the Iris dataset. Here, we can see that *petal width* moves closest to the center of our visualization. This means that the dimensionality reduction, projects along this axis, especially for $s \to \infty$. Furthermore, *sepal width* and *petal length* have almost no shaded area. Because we use the shaded area to encode and highlight the interval $s \in [0,1)$, this illustrates that the projection of these two axes remains almost the same while interpolating between regular PCA and our method.

6.3 Anuran Calls Dataset

The Anuran Calls dataset[2] contains acoustic sound features extracted from frog recordings. In total, there are 7195 instances of such calls, and they are grouped by family, genus, and species labels. Again, we perform aggregation of the instances, in this case, by looking at the family class label. However, the interesting aspect of this dataset is that, in contrast to the Iris dataset (Section 6.2), there is a different amount of instances per class. There are calls from four different frog families in this dataset—the numbers of instances per class are 4420, 2165, 542, and 68. These families can further be subgrouped by genus, yielding eight distinct clusters. Furthermore, it is important to note that many groups do not follow a normal distribution and exhibit varying modality, as can been seen in Figure 9.



(a) PCA on distributions

(b) PCA on orginal points

Fig. 8. Comparison between (a) our approach and (b) performing PCA on the original set of points of the Iris datasets. In (a) the aggregation into clusters has been performed before the projection, while in (b) the aggregation into the different clusters is performed visually through color.

So far, we have assumed that all aggregated distributions represent the same amount of instances. This can lead to overemphasized clusters if their original sample count is small. Concerning this dataset, this would mean that the family with 4420 instances would receive the same amount of weight as the family with 68 instances. To achieve a better fit to the actual data that these distributions stand for, we can adapt our method to take class weights into account by slightly modifying Equation 4. In particular, it suffices to use the weighted average to evaluate $\hat{\mathbb{E}}[\cdot]$. The computation of the sample mean needs to be adjusted accordingly.

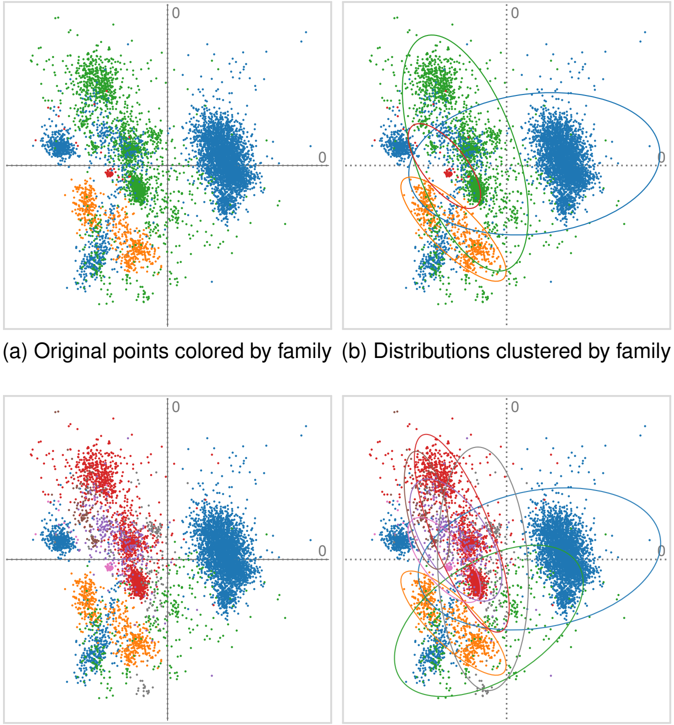
Figure 9 shows the comparison of our method, adapted to handle cluster weights, to regular PCA on the original set of points. For Figure 9ab, the data is clustered by *family*, yielding four distributions. Figure 9bc was aggregated by *genus*, which results in eight distinct discrete probability distributions. For the projections that were created using our method, we show the covariances that were extracted from each of the different clusters. This demonstrates that even if the clusters do not follow a simple distribution, such as the blue cluster in Figure 9b, our technique is still able to reconstruct the original PCA. The projections that are found for the point data and the aggregated data are visually the same. Assigning weights to each cluster according to the amount of data that it represents is an obvious application of this extension to our method. However, we can also imagine that this technique can be used in a more exploratory setting, for example, by investigating the effect of one cluster on the resulting principal components.

7 Comparison To Sampling

In this section, we provide a comparison of our method with another strategy that could be used to construct the covariance matrix for uncertain data: sampling. Instead of directly computing $\mathrm{Cov}(\mathbf{T},\mathbf{T})$ on the distributions, we can draw samples from each of them. If we concatenate the resulting set of points, we can use the conventional way for computing the covariance matrix as specified by Equation 1.

To compare the resulting covariance matrices, we need a suitable distance metric. We choose the Hellinger distance. It is commonly used to compare the results of linear models [37]. This distance metric is typically used to compare two multivariate normal distributions p

[2] https://archive.ics.uci.edu/ml/datasets/Anuran+Calls+(MFCCs)



(a) Original points colored by family (b) Distributions clustered by family

(c) Original points colored by genus (d) Distributions clustered by genus

Fig. 9. Comparison of projections resulting from conventional PCA and our method. Projections of the extracted covariances are shown as ellipses. Clustering by family leads to four clusters, while clustering by genus results in eight clusters. Although the clusters have a large variance in the number of instances (a), our weighted approach matches the projection of the original dataset well. The projection also remains stable for clustering by a different class label, here, by genus (b). Overall our method (d) performs well, even though not all clusters in the original dataset follow a normal distribution (c).

and q. It is based on the Bhattacharyya coefficient, which can be used to describe the overlap between p and q:

$$BC(p,q) = \int \sqrt{p(x)q(x)}\,dx$$

The Mahalanobis distance is a special case of the Bhattacharyya distance $(-\ln(BC(p,q))$ for distributions that share the same covariance. Using the definition of the Bhattacharyya coefficient, the Hellinger distance is defined as

$$H(p,q) = \sqrt{1 - BC(p,q)}$$

To apply this distance metric to the problem of comparing the results from principal component analysis, it is important to note that PCA is completly defined by its sample mean and overall covariance matrix. Together, we interpret these two artifacts as a multivariate normal distribution. The resulting distribution can then be compared using the Hellinger distance. In contrast to a description based on eigenvalues and eigenvectors, our method is invariant against flipping and no further preprocessing has to be performed.

For our experiment, we applied PCA to a synthetic dataset with 10 distributions $\mathbf{T}_{Syn} = \{\mathbf{t}_1, \ldots, \mathbf{t}_{10}\}$, each following a normal distribution $\mathbf{t}_i \sim N(\mu_\mathbf{i}, \Psi_i)$. All the means μ_i are drawn from another overarching multivariate normal distribution:

$$\mu_\mathbf{i} \sim N(\vec{0}, \Sigma)$$

The covariance of each of the distributions Ψ_i is constant across the dataset. It is created by reversing the elements of Σ. Because of this, all covariances also share the same determinant.

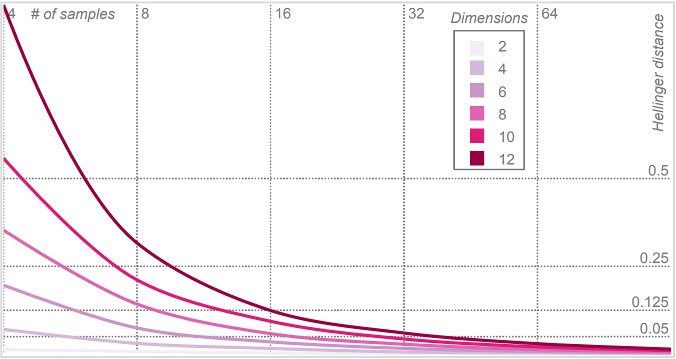


Fig. 10. Multiple comparison of our method to a sampling-based approach using the Hellinger distance for input data with two to 12 dimensions. The x axis shows the increasing number of samples that were used for the sampling strategy, while the y axis shows the distance to the result from our method. The required number of samples for a good result grows with the number of dimensions.

Figure 10 shows the results of our experiment. For each data point we performed 40 runs and chose the median outcome. We can draw several conclusions form our experiment. First, it shows that the sampling approach converges to our method with an increasing number of samples. This indicates that our method is a valid way to compute PCA on probability distributions. Second, it shows that our method scales far better than the sampling-based approach with a growing number of dimensions. We expect the curse-of-dimensionality to be the reason for this.

8 Discussion

In the following, we discuss the uncertainty-aware extension of PCA that we introduced, demonstrated, and assessed above from different perspectives. To begin with, we compare its computational complexity to traditional PCA. We follow up with more details on its application to interactive visualization, especially concerning scalability. Finally, we discuss the general limitations of PCA and how these carry over to our method.

8.1 Computational Complexity

In general, our method has the same computational complexity as regular PCA. For a dataset with N samples and D features, regular PCA has a computational complexity of $O(ND^2)$ for the computation of the covariance matrix. Retrieving the eigenvalues and eigenvectors has a complexity of $O(D^3)$.

With our method, samples N are D-dimensional probability distributions instead of points. In many cases, the probability density function of a random vector $\mathbf{t}_n$ is known analytically, and $\mathbb{E}[\mathbf{t}_n]$ as well as $\mathrm{Cov}(\mathbf{t}_n, \mathbf{t}_n)$ can be looked up in constant time $O(1)$. Our adapted computation of the global covariance matrix can be performed in $O(2 \cdot ND^2)$ since we additionally need to compute the average covariance matrix over all N distributions. Asymptotically, however, the constant factor 2 can be neglected. This results in a complexity of $O(ND^2)$ for determining the covariance matrix.

We share the extraction of the eigenvalues and eigenvectors with regular PCA. As mentioned above, this can be performed in $O(D^3)$. Thus, our technique is of similar complexity as standard PCA. Please note that in this analysis, we consider the aggregation of clusters as a preprocessing step (more details in the next section). Its complexity would add to the total complexity, but is not considered here. In the following section, we provide details on why preparing clusters is of special importance for the application of our technique to data visualization.

8.2 Interactive Visualization and Scalability

Big data is gaining relevance, and the amount of data that can be acquired and stored grows rapidly. For example, the Large Hadron

Collider (LHC) at CERN exceeded 200 Petabytes of collected sensor data already in 2017 [9]. At the same time, it often is critical to visualize such data for exploration, analysis, and knowledge generation [25]. Processing latencies are of significant concern for interactive visualization regarding big data. We tackle this problem by separating the computationally complex task of data aggregation from the projection and visualization tasks. Since our method is aware of the shape of the distributions, we can approximate the projection of clustered datasets by the projection of their respective distributions. For a large number of samples N in a D-dimensional feature space, this aggregation step is computationally costly since the covariance matrices have to be computed in $O(ND^2)$. The advantage of our method is that the aggregation can be done instantly during data acquisition and, in case memory demands are of concern, there is even no need to store raw data persistently [42]. In some fields, it is already common practice to aggregate data as a preprocessing step, for example, the *in-situ* analysis in large-data visualization [7]. Using our method, the characteristics of the data are preserved during the complete pipeline, and its influence on the projection can still be taken into account during the analysis process. Please note that when a cluster of multiple data points is aggregated by abstracting it as a normal distribution, the estimation of the covariance matrix is an inevitable step. To do so, the number of data points needs to be sufficient concerning the number of dimensions, and there must not be problems with (local) outliers [30]. Similarly, a small number of clusters can be a problem in high-dimensional space [14]. By scaling the uncertainty of each cluster depending on the number of data points, it contains, our method compensates for differences in cluster sizes, as outlined in Section 6.3. However, more research needs to be done in the direction of assessing whether the additional information provided by each clusters' weight and error covariance matrix can fully counter this problem.

8.3 Limitations of PCA

In practice, PCA is applied to all kinds of datasets, where it is commonly used as a tool for exploratory analysis. Conceptually, our approach yields a projection operator that is more aware of the uncertainty in the data. Just as with other linear methods, important information that is present in the non-principal components gets discarded due to the orthographic projection, which can guide the analysis into the wrong direction. Our method inherits this limitation. For regular PCA, methods have been developed to mitigate these effects—we provide an overview in Section 2. For one, this is because one of the terms of our method essentially performs PCA on the expected values of each of the distributions, as described in Section 4.2. With regard to the uncertainty in the data, a second limiting factor can arise: if the fraction of the covariance introduced by the uncertainty in the data is small in comparison to the covariance introduced by the expected values, and if the uncertainty happens to be orthogonal to the projection, it can also remain covert in the final representation. Future research may investigate how non-linear methods, which could alleviate this problem, can be generalized to probability distributions too.

Several other factors pose challenges to finding the correct principal components. The presence of outliers in the data can strongly influence the resulting projection. This stems from the quadratic term in the computation of the covariance matrix. When outliers are of concern, forms of *Robust PCA* (see Section 2), which rely on solving optimization problems, can be applied. It remains to be seen how similar approaches can be adapted to uncertainty-aware PCA. Although PCA was originally developed for real-valued data, it is often also used on datasets where some of the axes represent ordinal, and sometimes even categorical values. Naturally, these axes can contain uncertainty information as well. Furthermore, as of now, we do not explicitly model missing values. In the context of regular PCA, several techniques have been developed to deal with this—Dray and Josse [6] provide a summary of approaches that can be applied in this case. One straightforward way to handle these inputs in our framework nonetheless is imputation, as we have done in the student grade example provided in Section 6.1. With our method, these imputed values can even take the form of more complex distributions, which is why we see this as a practical workaround.

9 Conclusion

In this paper, we have presented a technique for performing principal component analysis on probability distributions. Unlike previous work, which mainly was concerned with non-correlated error models, our method works on arbitrary distributions. We achieve this by incorporating first and second moments of the uncertain input data into the calculation of the global covariance matrix. Our formulation of the global covariance matrix offers the potential for various extensions to traditional PCA. Particularly, in this paper, we have shown the application to aggregated datasets (Section 6.2 and Section 6.3) and datasets with explicitly encoded errors (Section 6.1).

Principal component analysis, and linear dimensionality reduction techniques in general, have the advantage over non-linear methods that the projections remain interpretable. The principal components found by PCA are linear combinations of the axes from the original data space. With our technique, scaling the influence of the covariances of each of the distributions allows us to perform sensitivity analysis concerning uncertainty. The factor traces we propose are a visual method to assess how uncertainty in the data is reflected by the contributions of each original dimension to the principal components. Further, our technique preserves the low computational complexity and clear algorithmic structure of traditional PCA. This enables the assessment of uncertainty induced differences to the projection by sampling different parameters for scaling the uncertainty. As a result, our technique constitutes a next step towards the earnest consideration of uncertainty in the analysis of high-dimensional data and forms the foundation for straightforward extensions in numerous directions.

Appendix

We show that our method, provided by Equation 4, indeed yields a covariance matrix by looking at the different terms of this equation. A matrix K is positive semi-definite if $\vec{u}^\mathsf{T} K \vec{u} \geq 0$, for every non-zero vector $\vec{x}$.

Theorem 1. *The outer product $\vec{x}\vec{x}^\mathsf{T} \in \mathbb{R}^{d \times d}$ of a vector $\vec{x} \in \mathbb{R}^d$ with itself always results in a symmetric, positive semi-definite matrix.*

Proof. Let $\vec{u} \in \mathbb{R}^d$ be a nonzero vector. Using the definition of positive semi-definitness from above,

$$\vec{u}^\mathsf{T}(\vec{x}\vec{x}^\mathsf{T})\vec{u} = (\vec{x}^\mathsf{T}\vec{u})^2 \geq 0.$$

The symmetry follows from the definition of matrix multiplication. □

Our method differs from regular PCA in one term, which is defined in Equation 5: In essence, this term computes the arithmetic mean of the covariance matrices $\mathrm{Cov}(\mathbf{t}_i, \mathbf{t}_i)$ of each distribution $\mathbf{t}_i$. A matrix is a covariance matrix if and only if it is *symmetric* and *positive semi-definite*. By definition, $\mathrm{Cov}(\mathbf{t}_i, \mathbf{t}_i)$ always satisfies this property.

Theorem 2. *Let $\mathbf{K} = \{K_1, \ldots, K_N\}, K_n \in \mathbb{R}^{d \times d}$ be a set of covariance matrices, then the arithmetic mean of this set $\frac{1}{N}\sum_{n=1}^{N} K_n$ is a covariance matrix.*

Proof. Let $\vec{u} \in \mathbb{R}^d$ be a nonzero vector and $A, B \in \mathbb{R}^{d \times d}$ positive semi-definite matrices. Both addition $A + B$, and multiplication with a scalar $kA, k \geq 0$ result in positive semi-definite matrices:

$$\vec{u}^\mathsf{T}(A+B)\vec{u} = \vec{u}^\mathsf{T}A\vec{u} + \vec{u}^\mathsf{T}B\vec{u}$$
$$\vec{u}^\mathsf{T}(kA)\vec{u} = k(\vec{u}^\mathsf{T}A\vec{u})$$

Because of this and the properties of symmetric matrices, it follows that the arithmetic mean of $\mathbf{K}$ is a symmetric and positive semi-definite matrix and therefore also a covariance matrix. □

Acknowledgments

We are very grateful to Sven Kosub, who helped us flesh out the mathematical model presented in Section 4. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 251654672 – TRR 161. Additionally, this work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825041.

References

[1] J.-H. Ahn and J.-H. Oh. A constrained EM algorithm for principal component analysis. *Neural Computation*, 15(1):57–65, 2003. doi: 10.1162/089976603321043694

[2] C. Bishop. Bayesian PCA. In *Advances in Neural Information Processing Systems*, vol. 11, pp. 382–388. MIT Press, 1999.

[3] C. J. C. Burges. Dimension reduction: A guided tour. *Foundation and Trends in Machine Learning*, 2(4):275–365, 2009. doi: 10.1561/2200000002

[4] J. P. Cunningham and Z. Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, 16(1):2859–2900, 2015.

[5] T. Denoeux and M.-H. Masson. Principal component analysis of fuzzy data using autoassociative neural networks. *IEEE Transactions on Fuzzy Systems*, 12(3):336–349, 2004. doi: 10.1109/tfuzz.2004.825990

[6] S. Dray and J. Josse. Principal component analysis with missing values: a comparative survey of methods. *Plant Ecology*, 216(5):657–667, 2014. doi: 10.1007/s11258-014-0406-z

[7] S. Dutta, C.-M. Chen, G. Heinlein, H.-W. Shen, and J.-P. Chen. In situ distribution guided analysis and visualization of transonic jet engine simulations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):811–820, 2017. doi: 10.1109/tvcg.2016.2598604

[8] R. Faust, D. Glickenstein, and C. Scheidegger. DimReader: Axis lines that explain non-linear projections. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):481–490, 2019. doi: 10.1109/TVCG.2018.2865194

[9] M. Gaillard. CERN Data Centre passes the 200-petabyte milestone. `https://home.cern/news/news/computing/cern-data-centre-passes-200-petabyte-milestone`, 2017. [Online; accessed 22. Mar. 2019].

[10] P. Giordani and H. A. Kiers. A comparison of three methods for principal component analysis of fuzzy interval data. *Computational Statistics & Data Analysis*, 51(1):379–397, 2006. doi: 10.1016/j.csda.2006.02.019

[11] R. Hermida. The problem of allowing correlated errors in structural equation modeling: concerns and considerations. *Computational Methods in Social Sciences*, 3(1):05–17, 2015.

[12] G. E. Hinton and T. Shallice. Lesioning an attractor network: Investigations of acquired dyslexia. *Psychological Review*, 98(1):74–95, 1991. doi: 10.1037/0033-295X.98.1.74

[13] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley and Sons, 2001.

[14] I. M. Johnstone and A. Y. Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486):682–693, 2009. doi: 10.1198/jasa.2009.0121

[15] E. Kandogan. Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *Proceedings of the IEEE Information Visualization Symposium, Late Breaking Hot Topics*, pp. 9–12, 2000.

[16] Y. Koren and L. Carmel. Robust linear dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):459–470, 2004. doi: 10.1109/tvcg.2004.17

[17] D. J. Lehmann and H. Theisel. Optimal sets of projections of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):609–618, 2016. doi: 10.1109/TVCG.2015.2467132

[18] S. Liu, P.-T. Bremer, J. T. Jayaraman, B. Wang, B. Summa, and V. Pascucci. The Grassmannian atlas: A general framework for exploring linear projections of high-dimensional data. *Computer Graphics Forum*, 35:1–10, 2016. doi: 10.1111/cgf.12876

[19] S. Liu, D. Maljovec, B. Wang, P. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1249–1268, 2017. doi: 10.1109/TVCG.2016.2640960

[20] S. Nakajima, M. Sugiyama, and S. D. Babacan. On Bayesian PCA: Automatic dimensionality selection and analytic solution. In *International Conference on Machine Learning*, pp. 497–504, 2011.

[21] L. Nonato and M. Aupetit. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2650–2673, 2018. doi: 10.1109/TVCG.2018.2846735

[22] M. N. Nounou, B. R. Bakshi, P. K. Goel, and X. Shen. Bayesian principal component analysis. *Journal of Chemometrics*, 16(11):576–595, 2002. doi: 10.1002/cem.759

[23] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[24] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2005.

[25] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim. Knowledge generation model for visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1604–1613, 2014. doi: 10.1109/TVCG.2014.2346481

[26] G. Sanguinetti, M. Milo, M. Rattray, and N. D. Lawrence. Accounting for probe-level noise in principal component analysis of microarray data. *Bioinformatics*, 21(19):3748–3754, 2005. doi: 10.1093/bioinformatics/bti617

[27] C. Schulz, A. Nocaj, J. Görtler, O. Deussen, U. Brandes, and D. Weiskopf. Probabilistic graph layout for uncertain network visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):531–540, 2017. doi: 10.1109/tvcg.2016.2598919

[28] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In *Artificial Neural Networks — ICANN'97*, pp. 583–588. Springer, 1997.

[29] A. P. Seyranian, O. N. Kirillov, and A. A. Mailybaev. Coupling of eigenvalues of complex matrices at diabolic and exceptional points. *Journal of Physics A: Mathematical and General*, 38(8):1723–1740, 2005. doi: 10.1088/0305-4470/38/8/009

[30] G. Shevlyakov and P. Smirnov. Robust estimation of the correlation coefficient: An attempt of survey. *Australian Journal of Statistics*, 1 & 2:147–156, 2011.

[31] M. Skeels, B. Lee, G. Smith, and G. G. Robertson. Revealing uncertainty for information visualization. *Information Visualization*, 9(1):70–81, 2009. doi: 10.1057/ivs.2009.1

[32] C. Spearman. General intelligence, objectively determined and measured. *American Journal of Psychology*, 15:201–293, 1904.

[33] D. Spiegelhalter, M. Pearson, and I. Short. Visualizing uncertainty about the future. *Science*, 333(6048):1393–1400, 2011. doi: 10.1126/science.1191181

[34] J. H. Steiger. Principal Components Analysis, Feb 2015. [Online; accessed 9 Mar 2019].

[35] D. Streeb, R. Kehlbeck, D. Jäckle, and M. El-Assady. Distances, neighborhoods, or dimensions? Projection literacy for the analysis of multivariate data. `https://visxprojections.dbvis.de`, 2018. Workshop at IEEE VIS Conference.

[36] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999. doi: 10.1111/1467-9868.00196

[37] E. Torgersen. *Comparison of linear models*, p. 411–504. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1991. doi: 10.1017/CBO9780511666353.009

[38] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[39] S. Tripathi and R. S. Govindaraju. Engaging uncertainty in hydrologic data sets using principal component analysis: BaNPCA algorithm. *Water Resources Research*, 44(10), 2008. doi: 10.1029/2007WR006692

[40] N. Vaswani, Y. Chi, and T. Bouwmans. Rethinking principal component analysis (PCA) for modern data sets: Theory, algorithms, and applications. *Proceedings of the IEEE*, 106(8):1274–1276, 2018. doi: 10.1109/JPROC.2018.2853498

[41] J. von Neumann and E. P. Wigner. Über merkwürdige diskrete Eigenwerte. In *The Collected Works of Eugene Paul Wigner*, pp. 291–293. Springer, 1991. doi: 10.1007/978-3-662-02781-3_19

[42] Z. Wang, N. Ferreira, Y. Wei, A. S. Bhaskar, and C. Scheidegger. Gaussian cubes: Real-time modeling for visual exploration of large multidimensional datasets. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):681–690, 2017. doi: 10.1109/TVCG.2016.2598694

[43] M. Wattenberg, F. Viégas, and I. Johnson. How to use t-SNE effectively. *Distill*, 2016. doi: 10.23915/distill.00002

[44] T. D. Wickens. *The Geometry of Multivariate Statistics*. Psychology Press, 1994.