Representation Learning of Compositional Data

Marta Avalos-Fernandez[†] **Richard Nock**[‡§♮] **Cheng Soon Ong**[‡§] **Julien Rouar**[†] **Ke Sun**[‡] [*]
[†]Université de Bordeaux, [‡]Data61,
[§]the Australian National University and [♮]the University of Sydney
first.last@{u-bordeaux.fr,data61.csiro.au}

Abstract

We consider the problem of learning a low dimensional representation for compositional data. Compositional data consists of a collection of nonnegative data that sum to a constant value. Since the parts of the collection are statistically dependent, many standard tools cannot be directly applied. Instead, compositional data must be first transformed before analysis. Focusing on principal component analysis (PCA), we propose an approach that allows low dimensional representation learning directly from the original data. Our approach combines the benefits of the log-ratio transformation from compositional data analysis and exponential family PCA. A key tool in its derivation is a generalization of the scaled Bregman theorem, that relates the perspective transform of a Bregman divergence to the Bregman divergence of a perspective transform and a *remainder* conformal divergence. Our proposed approach includes a convenient surrogate (upper bound) loss of the exponential family PCA which has an easy to optimize form. We also derive the corresponding form for nonlinear autoencoders. Experiments on simulated data and microbiome data show the promise of our method.

1 Introduction

Compositional data analysis (CoDA) is a subfield of statistics introduced more than three decades ago [3, 2, 1, 29]. Compositional data consist of a collection of nonnegative measurements that sum to a constant value, typically, proportions that sum to 1. Because knowing the sum, one component can be determined from the sum of the remainder, the parts that make up the composition are mathematically and statistically dependent. This distinct structure complicates analysis and does not allow standard statistical analyses. Ignoring the underlying nature of the data studied might give rise to misleading conclusions.

Among others, [1] and [13] provided a framework to perform CoDA by mapping data from the constrained simplex space to the Euclidian space using nonlinear log-ratio transforms. In this paper, we focus on Principal Components Analysis (PCA), one of the main tools for exploratory analysis of compositional data. Just like in standard Euclidean data, it is particularly useful when the first few principal components explain enough variability to be considered as representative. Unfortunately, any operation of centering or scaling destroys the compositional nature of the data, which complicates a direct application of PCA.

Our motivation for studying CoDA comes from the recent explosion of microbiome studies [14, 15]. Indeed, spectacular advances in 16S rRNA gene sequencing of the bacterial component of the human microbial community (microbiota) have enabled researchers to investigate human health and disease, leading to new insights into the role of these microbial communities. The microbiota sequencing data are measured as read counts interpreted as a species' abundance in a microbial community. To make the microbial abundance comparable across samples, data are normalized to the relative abundances

[*] Authors in alphabetical order

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

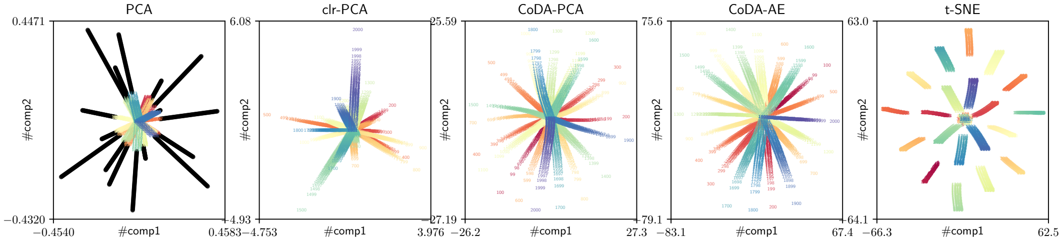

Figure 1: 2D Visualization of the low dimensional representation A on the `arms` dataset.

of all bacteria observed. On the other hand, because high-throughput experiments produce large amounts of data, multivariate analysis is indispensable [27, 21]. There is a stress to understand the soundness of models [5].

In this paper, we propose to learn a low dimensional representation of CoDA from the *original* data. To account for the nonlinearity due to the compositional nature of the data, we start from exponential family PCA [12] that we augment with the compositional constraint and then simplify the loss to be optimized via a generalization of a recent result [25] on Bregman divergences, which may be of independent interest. We also propose a nonlinear autoencoder (AE) version to learn the low dimensional representation.

Let us examine a toy example to illustrate our approach. We generate the `arms` dataset in $\mathfrak{S}^{19}$ by evenly interpolating the simplex center and each of the 20 vertices with 100 points, therefore yielding a matrix $\mathsf{X}_{20\times 2000}$. Figure 1 shows the 2D representation $\mathsf{A}_{2\times 2000}$ computed by five methods: the standard `PCA`; `clr-PCA` computes the standard PCA after performing clr; `CoDA-PCA` and `CoDA-AE` are our proposed methods; t-SNE [32] is a popular nonlinear dimensionality reduction method and is applied on X directly. In the `PCA` plot, the black segments indicate that the PCA reconstruction is outside of the simplex. PCA cannot be directly adapted to CoDA because the projection on the principal components may go beyond the convex hull of the vertices. It is clear that only `CoDA-PCA` and `CoDA-AE` uncover the true structure, where all the arms are clearly presented, and their connections are faithfully presented.

2 Compositional Data Analysis

We briefly review some definitions of CoDA. Compositional data are proportions: X is a compositional dataset if and only if $\mathsf{X} \in \Re^{d\times m}$ such that $\forall i \in [1, ..., m]$ the vector column $\boldsymbol{x}_i$ of X is in the simplex $\mathfrak{S}^d = \left\{\boldsymbol{x} \in \Re^d \ : \ \forall j, x_j > 0;\ \sum_{j=1}^d x_j = \kappa\right\}$, where $\kappa > 0$ is a constant, classically 1. Here the superscript d does not denote the dimensionality as $\dim(\mathfrak{S}^d) = d - 1$. For a dataset X' which contains counts of strictly positive values, we reduce it to a compositional dataset by dividing out the totals, that is we compute the CoDA set X such that: $\boldsymbol{x}_i = \boldsymbol{x}'_i \frac{1}{\sum_{j=1}^d x'_{ji}}$ is the vector of proportions for individual i.

Using Bregman divergences makes explicit a dual affine [6] coordinate space which is in fact the log coordinates of Aitchison [1]. It is in this space that we have affine constraints, which are therefore non-linear in the "primal", ambient space. To manage this nonlinear structure, it has been proposed [4] to first apply a log-ratio transformation to transpose the data into real Euclidean space. For instance, the additive log-ratio transformation (alr) applies log-ratio between a component and a reference component; the centered log-ratio transformation (clr) scales each subject vector by its geometric mean; and the isometric log-ratio transformation (ilr) is associated with an orthogonal coordinate system in the simplex. Afterwards, standard PCA is performed.

By definition, the clr transformation is

$$\boldsymbol{c}_{\mathrm{KL}}(\boldsymbol{x}) = \log\left(\frac{\boldsymbol{x}}{g(\boldsymbol{x})}\right) = \mathsf{C}^{\mathrm{clr}}\log(\boldsymbol{x}) = \log(\boldsymbol{x}) - \overline{\log(\boldsymbol{x})}\mathbf{1}_d, \tag{1}$$

where $g(\boldsymbol{x}) = (\prod_{j=1}^d x_j)^{1/d}$ the geometric mean of $\boldsymbol{x}$, $\overline{\boldsymbol{x}} = \frac{1}{d}\sum_{j=1}^d x_j$ is the arithmetic mean of $\boldsymbol{x}$, $\mathbf{1}_d$ is the $1 \times d$ vector of all ones, and $\mathsf{C}^{\mathrm{clr}} = \mathsf{I}_d - \frac{1}{d}\mathbf{1}_d\mathbf{1}_d^\intercal$. The purpose of the log-ratio transformation (centered or not) is to go back to $\Re^d$ from $\mathfrak{S}^d$ without losing information. Notice that $\log(x_j)$, $\overline{\log(\boldsymbol{x})} \in (-\infty, 0)$ so the compositional data is embedded in $\Re^d$ under the clr transformation. The reverse operation: $\boldsymbol{x} = \boldsymbol{c}_{\text{KL}}^{-1}(\boldsymbol{x'}) = \exp(\boldsymbol{x'}) \prod_{j=1}^{d} \exp(\frac{1}{1-d} x'_j)$ embeds $\Re^d$ into $\mathfrak{S}^d$. See the table below for a comparison of clr, alr and ilr, presented as different transformations $\mathsf{C} \log(\boldsymbol{x})$. They are equivalent up to linear transformations. Without loss of generality we focus on the clr.

clr	alr	ilr
$\mathsf{C}^{\text{clr}}_{d \times d} = \mathsf{I}_d - \frac{1}{d} \mathbf{1}_d \mathbf{1}_d^\top$	$\mathsf{C}^{\text{alr}}_{(d-1) \times d} = [\mathsf{I}_{d-1}, -\mathbf{1}_{d-1}]$	$\mathsf{C}^{\text{ilr}}_{(d-1) \times d} \in \left\{ \mathsf{R}\mathsf{C}^{\text{clr}} \, : \, \mathsf{R}\mathsf{R}^\top = \mathsf{I}_{d-1} \right\}$

However interpreting the resulting coordinates is still challenging [13, 24]: alr transformation is no distance-preserving; clr leads to degenerate distributions and singular covariance matrices; ilr avoids the precedent drawbacks, but still, results from complicated nonlinear transformations are difficult to interpret. Currently, there seems to be no consensus about the best practices ([16] versus [31]) and, in all cases, log-transforming is not a remedy for all the difficulties arisen by CoDA [20].

3 Exponential Family Principal Component Analysis

Another way to apply dimension reduction is to perform a generalized PCA on crude count data. Based on the same ideas as the generalised linear model, [12] described a generalized PCA model for distributions from the exponential family. We first recall the standard PCA setting.

3.1 Principal Component Analysis

For simplicity suppose that the data matrix X is already centered that can be easily achieved by appending to A matrix a row of ones.

(Traditional PCA) We have a dataset $\mathsf{X} \in \Re^{d \times m}$ that we approximate as $\mathsf{X} \sim \mathsf{V}^\top \mathsf{A}$ by minimizing the following loss wrt the constraints $\mathsf{A} \in \Re^{\ell \times m}$, $\mathsf{V} \in \Re^{\ell \times d}$, $\mathsf{V}\mathsf{V}^\top = \mathsf{I}_\ell$:

$$\ell_{\text{PCA}}(\mathsf{X}; \mathsf{A}, \mathsf{V}) \doteq \|\mathsf{X} - \mathsf{V}^\top \mathsf{A}\|_{\text{F}}^2 \ . \tag{2}$$

Hence, observations are column-wise. $\mathsf{V} : \Re^d \to \Re^\ell$ is surjective with $\mathsf{V}^\top \mathsf{V}$ defining a rank-ℓ projection, assuming in general $\ell < d$. A is the representation of data points. The goodness of fit of the representation is measured by the squared Frobenious norm. We summarise the different transformations and loss functions in Table 1.

Observe that instead of finding a linear representation A and its corresponding linear loadings V, we can consider nonlinear functions for encoding and decoding the latent representation. When the nonlinear encoder and decoder are implemented as feed-forward neural networks, we arrive at the autoencoder setting.

3.2 Bregman Divergence and φ-PCA

As mentioned in the introduction, compositional data do not live in Euclidean space. Count data are naturally linked to the Poisson distribution, and therefore we should consider an exponential family model for count data. From the Bayesian viewpoint, the PCA goal is to minimize a distance (L^2 for usual PCA) which is equivalent to a Bregman divergence minimization (or a Likelihood function maximization).

Definition 1 (Bregman Divergence) *Let $\varphi : \Re^d \to \Re$ convex differentiable. The Bregman divergence D_φ with generator φ is*

$$D_\varphi(\boldsymbol{x} \,\|\, \boldsymbol{x}') \doteq \varphi(\boldsymbol{x}) - \varphi(\boldsymbol{x}') - (\boldsymbol{x} - \boldsymbol{x}')^\top \nabla \varphi(\boldsymbol{x}'). \tag{3}$$

A Bregman divergence is just a truncation of the Taylor expansion of a function. It can therefore be defined for any differentiable function, not just the convex ones. If φ is *not* convex, we call D_φ a *Bregman distortion*, which is a signed dissimilarity. We denote by $\varphi^\star(\boldsymbol{x}) \doteq \sup_{\boldsymbol{y}} \{\boldsymbol{x}^\top \boldsymbol{y} - \varphi(\boldsymbol{y})\}$ the convex conjugate of the generator φ [9].

Table 1: Summary of methods in this paper

Method	Original	Reconstruction	Distortion	Notes
PCA	X	$\mathsf{V}^\intercal\mathsf{A}$	$\Vert \cdot - \cdot \Vert_{\mathrm{F}}^2$	classical PCA(2)
φ-PCA	X	$\nabla^*\varphi(\mathsf{V}^\intercal\mathsf{A})$	$D_\varphi(\cdot \Vert \cdot)$	exponential family PCA (4)
clr-PCA	$\boldsymbol{c}_{\text{KL}}(\mathsf{X})$	$\mathsf{V}^\intercal\mathsf{A}$	$\Vert \cdot - \cdot \Vert_{\mathrm{F}}^2$	CoDA with clr (5)
gauged-φ-PCA	$\nabla\varphi(\check{\mathsf{X}})$	$\mathsf{V}^\intercal\mathsf{A}$	$D_{\varphi^\star}(\cdot \Vert \cdot)$	General Bregman PCA (8)
CoDA PCA	$\boldsymbol{c}_{\text{KL}}(\mathsf{X})$	$\mathsf{V}^\intercal\mathsf{A}$	$D_{\exp}(\cdot \Vert \cdot)$	(11) is a special case of (8)
S-CODA-PCA	$\check{\boldsymbol{x}}_i$	$\nabla\check{\mathrm{KL}}(\exp(\mathsf{V}^\intercal\boldsymbol{a}_i))$	inner product	upper bound (17)
CoDA AE	X	$g_\theta \circ h_\Phi(\mathsf{X})$	$D_{\exp}(\cdot \Vert \cdot)$	neural networks g_θ and h_Φ

PCA has been generalized to the exponential families in a way that makes fitting occur in the *natural parameter* space [12, 19] (and references therein). The optimization problem is non-convex. The algorithmic strategy proposed by [12] is to use an alternating sequence of convex minimizations under constraints. Alternatively, [19] proposed maximizing the deviance (as a generalized notion of variance) and [10] proposed maximizing the likelihood function via a variational algorithm and gradient descent.

We denote exponential family PCA as φ-PCA, where φ is the cumulant of the exponential family, which is strictly convex differentiable with convex conjugate $\varphi^\star$, and uniquely determines the exponential family under mild conditions [7]. Note that for φ-PCA, X is not neccessarily in a vector space (e.g. $\mathsf{X} \not\subseteq \Re^{d\times m}$).

(φ-PCA) We have a dataset $\mathsf{X}_{d\times m}$ that we approximate as $\mathsf{X} \sim \nabla\varphi^\star(\mathsf{V}^\intercal\mathsf{A})$ with $\mathsf{A} \in \Re^{\ell\times m}$, $\mathsf{V} \in \Re^{\ell\times d}$, $\mathsf{V}\mathsf{V}^\intercal = \mathsf{I}_\ell$, through minimizing the Bregman loss

$$\ell_{\varphi\text{-PCA}}(\mathsf{X};\mathsf{A},\mathsf{V}) \doteq \sum_i D_\varphi(\boldsymbol{x}_i \,\|\, \nabla\varphi^\star(\mathsf{V}^\intercal\boldsymbol{a}_i)) = D_\varphi(\mathsf{X} \,\|\, \nabla\varphi^\star(\mathsf{V}^\intercal\mathsf{A})). \tag{4}$$

Vectors are column-vectors: $\boldsymbol{x}_i, \boldsymbol{a}_i$ are respectively column observation i in the ambient and principal spaces, respectively. This formulation has a major advantage that linear algebra may be used to fit A, V while X may not lie in a vector space, see for example [12, 19] and references therein. We remark that because of the dual symmetry of Bregman divergences, we have $D_\varphi(\mathsf{X} \,\|\, \nabla\varphi^\star(\mathsf{V}^\intercal\mathsf{A})) = D_{\varphi^\star}(\mathsf{V}^\intercal\mathsf{A} \,\|\, \nabla\varphi(\mathsf{X}))$ [8]. Notice there exists a little "hole" in the φ-PCA definition, as X is not necessarily easy to center when it is not in a vector space.

φ-PCA includes standard PCA as a special case as when $\varphi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_{\mathrm{F}}^2$ and the corresponding Bregman divergence becomes $D_\varphi(\boldsymbol{x} \,\|\, \boldsymbol{x}') = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|_{\mathrm{F}}^2$.

4 Exponential family PCA on Compositional Data

CoDA has found a workaround for the centering problem, *centered log-ratio coordinates*. From [3, Def. 4.6, Chap. 8] the associated loss is the standard PCA loss on clr transformed data:

$$\ell_{\text{clr-PCA}}(\mathsf{X};\mathsf{A},\mathsf{V}) \;\doteq\; \frac{1}{2}\|\boldsymbol{c}_{\text{KL}}(\mathsf{X}) - \mathsf{V}^\intercal\mathsf{A}\|_{\mathrm{F}}^2 \;=\; D_\varphi(\boldsymbol{c}_{\text{KL}}(\mathsf{X}) \,\|\, \nabla\varphi^\star(\mathsf{V}^\intercal\mathsf{A})), \tag{5}$$

where $\boldsymbol{c}_{\text{KL}}(\mathsf{X})$ is the centered log-ratio transform defined in Equation (1) and $\varphi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_{\mathrm{F}}^2$. Recall from the previous section that we could deal with crude count data by using exponential family PCA. However if we wish to perform PCA on the crude count data, while maintaining the clr transform, we need an additional normalization term, which requires us to obtain a gauged version of the Bregman divergence.

4.1 Scaled Bregman Theorem with Remainder

In this section we generalize the Scaled Bregman Theorem from [25, Theorem 1] to allow for a remainder term. We use it in this paper to deal with the perspective transform required for CoDA, but it may be of independent interest. Recall that φ is the generator of the Bregman distortion (Definition 1). We additionally define a perspective (or gauge) function g to deal with the fact that we are considering data on the simplex. Whenever φ and g are differentiable, the following is immediate from [25, Theorem 1].

Theorem 2 (Scaled Bregman Theorem with Remainder) *For any $\varphi : \mathcal{X} \rightarrow \Re$ and $g : \mathcal{X} \rightarrow \Re_*$ ($\Re_* = \Re \setminus \{0\}$) that are both differentiable, denoting*

$$\check{\boldsymbol{x}} \doteq \frac{\boldsymbol{x}}{g(\boldsymbol{x})} \quad \textit{and} \quad \check{\varphi}(\boldsymbol{x}) \doteq g(\boldsymbol{x}) \cdot \varphi\left(\frac{\boldsymbol{x}}{g(\boldsymbol{x})}\right) \ , \tag{6}$$

the following holds true:

$$g(\boldsymbol{x}) \cdot D_{\varphi}\left(\check{\boldsymbol{x}} \,\|\, \check{\boldsymbol{y}}\right) = D_{\check{\varphi}}\left(\boldsymbol{x} \,\|\, \boldsymbol{y}\right) + R_{\varphi,g}(\boldsymbol{x} \,\|\, \boldsymbol{y}) \ , \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X} \ , \tag{7}$$

where $R_{\varphi,g}(\boldsymbol{x} \,\|\, \boldsymbol{y}) \doteq \varphi^{\star}\left(\nabla\varphi(\check{\boldsymbol{y}})\right) \cdot D_g(\boldsymbol{x} \,\|\, \boldsymbol{y})$ is called the remainder.

We can abstract Theorem 2 by saying that for any φ, g differentiable, we have

$$\text{perspective-Bregman}(\varphi, g) = \text{Bregman}(\text{perspective}(\varphi)) + \text{conformal-Bregman}(g, \varphi),$$

where "$\text{perspective}(\varphi)$" is $\check{\varphi}$ in (6), and conformal divergences are defined and analyzed in [26]. General classes of perspective transforms of convex functions are introduced in [22, 23]. The notion of perspective transform of a Bregman divergence was introduced in [25]. In [25, Theorem 1], conditions are assumed that make $R_{\varphi,g}(\boldsymbol{x} \,\|\, \boldsymbol{y}) = 0$, resulting in the scaled Bregman theorem. Notice that D_{φ} is a Bregman distortion but not necessarily a Bregman divergence if φ is not convex. For reasons explained in [25], we call g a *gauge*. In the following we assume that φ is separable, so that we can use both notations $\nabla\varphi$ and φ' to denote the gradient and derivatives involving φ.

By Theorem 2, as long as $g(\boldsymbol{x})$ is homogeneous of degree one, $D_{\varphi}\left(\check{\boldsymbol{x}} \,\|\, \check{\boldsymbol{y}}\right)$ and $\frac{1}{g(\boldsymbol{x})}\left[D_{\check{\varphi}}\left(\boldsymbol{x} \,\|\, \boldsymbol{y}\right) + R_{\varphi,g}(\boldsymbol{x} \,\|\, \boldsymbol{y})\right]$ are both invariant to re-scaling of $\boldsymbol{x}$ and $\boldsymbol{y}$ and can therefore be used to deal with compositional data. A general formulation of g satisfying this condition can be $g(\boldsymbol{x}) = \prod_{j=1}^{d} x_j^{w_j}$, where $\forall j, w_j \geq 0$ and $\sum_{j=1}^{d} w_j = 1$. In this paper, we focus on the special case $\forall j, w_j = \frac{1}{d}$ so that $D_{\varphi}\left(\check{\boldsymbol{x}} \,\|\, \check{\boldsymbol{y}}\right)$ can be expressed in terms of the widely used clr transformation. Setting $\boldsymbol{w}$ to be a one-hot vector $(1, 0, \cdots, 0)$ can express $D_{\varphi}\left(\check{\boldsymbol{x}} \,\|\, \check{\boldsymbol{y}}\right)$ with the alr. This latter case will be omitted here.

4.2 Exponential Family CoDA

We are now in a position to derive the exponential family version of the loss in (5). Let $\check{\mathsf{X}}$ denote the matrix of the column vectors $\check{\boldsymbol{x}}_i$. It turns out that in the same way as (2) is an approximation of (4), the loss in (5) is an approximation of the gauged loss:

$$\ell_{\text{gauged-}\varphi\text{-PCA}}(\mathsf{X}; \mathsf{A}, \mathsf{V}) \doteq D_{\varphi^{\star}}(\mathsf{V}^{\intercal}\mathsf{A} \,\|\, \nabla\varphi(\check{\mathsf{X}})) \ = \ D_{\varphi}(\check{\mathsf{X}} \,\|\, \nabla\varphi^{\star}(\mathsf{V}^{\intercal}\mathsf{A})). \tag{8}$$

Note that the above expression is in terms of the normalised matrix $\check{\mathsf{X}}$. To unpack it in terms of the original data X, we apply Theorem 2. In the CoDA case, $\varphi^{\star}(z) \doteq \exp z$, the convex dual of $\varphi(z) \doteq z \log z - z$. Indeed, after remarking that $\nabla\varphi(\check{\mathsf{X}}) = \boldsymbol{c}_{\text{KL}}(\mathsf{X})$, it follows

$$\begin{aligned}\ell_{\text{gauged-KL-PCA}}(\mathsf{X}; \mathsf{A}, \mathsf{V}) &= D_{\exp}(\mathsf{V}^{\intercal}\mathsf{A} \,\|\, \boldsymbol{c}_{\text{KL}}(\mathsf{X})) = D_{\text{KL}}\left(\check{\mathsf{X}} \,\|\, \exp(\mathsf{V}^{\intercal}\mathsf{A})\right) \\ &= \mathbf{1}^{\intercal}\exp(\mathsf{V}^{\intercal}\mathsf{A})\mathbf{1} - \mathsf{trace}\left(\check{\mathsf{X}}^{\intercal}\mathsf{V}^{\intercal}\mathsf{A}\right) + \text{constant}.\end{aligned} \tag{9}$$

In other words, the CoDA PCA is in fact fitting natural parameters from centered log-ratios being natural coordinates as well. From (9) we observe that both of them live in the same space. Therefore $\mathsf{V}^{\intercal}\mathsf{A}$ is centered in the same way as $\boldsymbol{c}_{\text{KL}}(\mathsf{X})$, and so

$$\mathsf{V}\mathbf{1}_d \in \ker(\mathsf{A}^{\intercal}) \quad \Leftrightarrow \quad \mathsf{A}^{\intercal}\mathsf{V}\mathbf{1} = \mathbf{0}_m \ . \tag{10}$$

Remark that a centering assumption is also explicit in [3, Chapter 8, Eq. 8.1].

Hence, we can define the CoDA PCA problem as follows.

(CoDA PCA) We have a dataset $\mathsf{X} \in (\mathfrak{S}^d)^m$ that we approximate as $\boldsymbol{c}_{\text{KL}}(\mathsf{X}) \sim \mathsf{V}^\top\mathsf{A}$ by minimizing the following loss wrt the constraints $\mathsf{A} \in \Re^{\ell \times m}$, $\mathsf{V} \in \Re^{\ell \times d}$, $\mathsf{V}\mathsf{V}^\top = \mathsf{I}_\ell$, $\mathsf{A}^\top\mathsf{V}\mathbf{1} = \mathbf{0}$:

$$\ell_{\text{CoDA-PCA}}(\mathsf{X}; \mathsf{A}, \mathsf{V}) = D_{\exp}(\mathsf{V}^\top\mathsf{A} \,\|\, \boldsymbol{c}_{\text{KL}}(\mathsf{X})). \tag{11}$$

Regarding $\boldsymbol{c}_{\text{KL}}(\boldsymbol{x})$, $\check{\boldsymbol{x}}$ and $\boldsymbol{x}$ as different coordinate systems of $\mathfrak{S}^d$, we use the Fisher information metric (FIM) [6], whose formulation is well studied on the $\boldsymbol{x}$ coordinates, to define the corresponding *pullback metric* $\mathcal{G}$ under the $\boldsymbol{c}_{\text{KL}}(\boldsymbol{x})$ and $\check{\boldsymbol{x}}$ coordinates, meaning that these metrics correspond to the same underlying geometry of $\mathfrak{S}^d$. We have the following proposition (proof omitted; see [30] for similar derivations).

Proposition 3 *The FIM that uniquely defines the geometry of* $\mathbf{c} \in \left\{\boldsymbol{c}_{\text{KL}}(\boldsymbol{x}) \,:\, \boldsymbol{x} \in \mathfrak{S}^d\right\}$ *is given by* $\mathcal{G}_{ij}(\boldsymbol{c}) = \delta_{ij}\frac{\exp(c_i)}{\sum_{i=1}^d \exp(c_i)} - \frac{\exp(c_i+c_j)}{\left(\sum_{i=1}^d \exp(c_i)\right)^2}$*; the FIM under the coordinates* $\check{\boldsymbol{x}}$ *is given by* $\mathcal{G}_{ij}(\check{\boldsymbol{x}}) = \delta_{ij}\frac{1}{\check{x}_i \sum_{i=1}^d \check{x}_i} - \frac{1}{(\sum_{i=1}^d \check{x}_i)^2}$*, where* $\delta_{ij} = 1$ *if* $i = j$ *otherwise* $\delta_{ij} = 0$.

Intuitively, the metric $\mathcal{G}$ measures the local distance $\mathrm{d}\check{\boldsymbol{x}}^\top\mathcal{G}(\check{\boldsymbol{x}})\mathrm{d}\check{\boldsymbol{x}}^\top$ of a tiny shift $\mathrm{d}\check{\boldsymbol{x}}$. It is not everywhere identity as in a Euclidean space. Therefore the distance should not be measured by the Frobenious norm as in (5). In contrast, our loss $\ell_{\text{CoDA-PCA}}(\mathsf{X}; \mathsf{A}, \mathsf{V})$ is based on the KL divergence which locally agrees with the FIM [6].

4.3 Relating CoDA PCA to φ-PCA

We now define and analyze a generalized perspective transform of the generator of KL divergence: let $\check{\text{KL}}(\boldsymbol{x}) \doteq g(\boldsymbol{x}) \cdot \sum_{j=1}^d \varphi(x_j/g(\boldsymbol{x}))$ where $\varphi(z) \doteq z\log(z) - z$ and $g(\boldsymbol{x}) \doteq (\prod_j x_j)^{1/d}$.

Lemma 4 *(Properties of* $\check{\text{KL}}$*)* $\check{\text{KL}}$ *satisfies the following properties:*

(1) $\check{\text{KL}}$ *is convex;*

(2) the general term of the Hessian H *of* $\check{\text{KL}}$ *is*

$$\text{H}_{ij} \doteq \text{H}_{ij}(\check{\text{KL}}(\boldsymbol{x})) \;=\; \frac{1}{dx_j} \cdot \left\{ \begin{array}{cc} -u_{ji} & \textit{if} \quad j \neq i \\ \sum_{k \neq j} u_{kj} & \textit{otherwise} \end{array} \right. , \tag{12}$$

where $u_{ab} \doteq 1 + x_a/x_b$*. Furthermore,*

$$\boldsymbol{z}^\top\text{H}\boldsymbol{z} \;=\; \frac{1}{2d}\sum_{ij}(x_i + x_j) \cdot \left(\frac{z_i}{x_i} - \frac{z_j}{x_j}\right)^2, \forall \boldsymbol{z} \in \Re^d. \tag{13}$$

Hence, $\boldsymbol{z}^\top\text{H}\boldsymbol{z} \geq 0, \forall \boldsymbol{x} \in \Re^d_{++}, \forall \boldsymbol{z} \in \Re^d$ *and* $\boldsymbol{z}^\top\text{H}\boldsymbol{z} = 0$ *only when* $\boldsymbol{z} \propto \boldsymbol{x}$*;*

(3) function $\check{\text{KL}} \circ \exp$ *is 1-homogeneous on* $\text{span}(\{\mathbf{1}\})^\perp$.

(Proof in SM, Section D) A consequence of Theorem 2 is the following Corollary.

Corollary 5 *For any* A, V *such that* $\mathsf{A}^\top\mathsf{V}\mathbf{1} = \mathbf{0}$*, we have*

$$\ell_{\text{CoDA-PCA}}(\mathsf{X}; \mathsf{A}, \mathsf{V}) \;\leq\; \tilde{\ell} \doteq \sum_i \frac{1}{g(\boldsymbol{x}_i)} \cdot D_{\check{\text{KL}}}(\boldsymbol{x}_i \| \exp(\mathsf{V}^\top\boldsymbol{a}_i)). \tag{14}$$

Hence, the CoDA PCA loss is upperbounded by a weighted generalized φ*-PCA loss. Furthermore,*

$$D_{\check{\text{KL}}}(\boldsymbol{x}_i \| \exp(\mathsf{V}^\top\boldsymbol{a}_i)) \;=\; \check{\text{KL}}(\boldsymbol{x}_i) - \boldsymbol{x}_i^\top\nabla\check{\text{KL}}(\exp(\mathsf{V}^\top\boldsymbol{a}_i)) \tag{15}$$

Proof Since g is concave (Example 1 in Supplement), $D_g(\boldsymbol{x}\|\boldsymbol{y}) = -D_{-g}(\boldsymbol{x}\|\boldsymbol{y}) \leq 0$, and (14) follows from Theorem 7 and the fact that $r_i \geq 0, \forall i$, which shows (14). (15) is a consequence of the analytical construct of Bregman divergences (Definition 1) and point (3) in Lemma 4 and the fact that $\mathsf{V}^\top\boldsymbol{a}_i \in \text{span}(\{\mathbf{1}\})^\perp$ by assumption. ■

Remark In [3, Chapter 8], CoDA PCA is presented as a (centered) regular PCA over data that been subject to *two* transforms via the centered log-ratio coordinates. What Corollary 5 shows is that we can solve the problem via a surrogate formulation using *non transformed data* but minimizing a loss which is that of a φ-PCA transformed *twice*: first taking a perspective transform of the KL generator ($\check{\mathrm{KL}}$) and then having a weighted Bregman divergence minimization ($g^{-1}(.)$). We remark that weights can also be folded in the arguments as we have:

$$\tilde{\ell} \quad = \quad \sum_i \mathrm{KL}(\check{\boldsymbol{x}}_i) - \check{\boldsymbol{x}}_i^\top \nabla \check{\mathrm{KL}}(\exp(\mathsf{V}^\top \boldsymbol{a}_i)) \ . \tag{16}$$

□

Furthermore, the leftmost argument in (16) plays no role in its minimization, and therefore we get the Surrogate CoDA PCA (s-CoDA-PCA) by replacing (11) with a simple inner product:

$$\ell_{\text{s-CoDA-PCA}}(\mathsf{X}; \mathsf{A}, \mathsf{V}) \quad \doteq \quad -\sum_i \check{\boldsymbol{x}}_i^\top \nabla \check{\mathrm{KL}}(\exp(\mathsf{V}^\top \boldsymbol{a}_i)) \ . \tag{17}$$

5 Implementations

Both the CoDA-PCA in (11) and the s-CoDA-PCA in (17) can be equivalently written as the following unconstrained problems

$$(\text{CoDA-PCA}) \quad \underset{\mathsf{B},\mathsf{U}}{\operatorname{argmin}} \ \left[\mathbf{1}_d^\top \exp(\mathsf{Y})\mathbf{1}_m - \mathsf{trace}\left(\check{\mathsf{X}}^\top \mathsf{Y}\right)\right], \tag{18}$$

$$(\text{s-CoDA-PCA}) \quad \underset{\mathsf{B},\mathsf{U}}{\operatorname{argmin}} \ \mathsf{trace}\left[\check{\mathsf{X}}^\top \left(\exp(-\mathsf{Y}) \circ \frac{\mathbf{1}_d \mathbf{1}_d^\top}{d} \exp(\mathsf{Y}) - \mathsf{Y}\right)\right], \tag{19}$$

where "$\circ$" means element-wise product, $\exp(\cdot)$ is element-wise exponential, and $\mathsf{Y} = \mathsf{C}\mathsf{U}^\top\mathsf{B}$ with $\mathsf{C} \in \Re^{d\times d}$, $\mathsf{U} \in \Re^{\ell\times d}$, $\mathsf{B} \in \Re^{\ell\times m}$. C is a constant centering matrix satisfying $\operatorname{rank}(\mathsf{C}) = d - 1$, $\mathsf{C}^\top\mathbf{1} = \mathbf{0}$, so that Y's columns are automatically centered and $\mathsf{Y}^\top\mathbf{1} = \mathsf{B}^\top\mathsf{U}\mathsf{C}^\top\mathbf{1} = \mathbf{0}$. Any C satisfying this condition corresponds to a valid re-parametrization of the feasible space, for example $\mathsf{C} = \mathsf{I}_d - \frac{1}{d}\mathbf{1}_d\mathbf{1}_d^\top$ or $\mathsf{C} = \mathsf{I}_d - \mathsf{I}_d^\sharp$ ($\mathsf{I}_d^\sharp$ circularly raises the diagonal entries of I_d by 1 row). U's rows form a nonorthogonal basis of $\Re^d$. B's columns are the sample coordinates in such a basis. After optimization, we take the QR decomposition $\mathsf{C}(\mathsf{U}^\star)^\top = (\mathsf{V}^\star)^\top\mathsf{T}^\star$, where $\mathsf{V}^\star$'s rows are orthonormal. Therefore $\mathsf{C}(\mathsf{U}^\star)^\top\mathsf{B}^\star = (\mathsf{V}^\star)^\top\mathsf{T}^\star\mathsf{B}^\star$ and $\mathsf{A}^\star = \mathsf{T}^\star\mathsf{B}^\star$ is the corresponding coordinates. An optimal solution of the original constrained PCA problem is given by $(\mathsf{V}^\star, \mathsf{A}^\star)$.

Although the losses in (18) and (19) are non-convex, they are both *bi-convex*. Fixing U, the loss is a strictly convex function of B that is decomposed into a sum of per-sample convex functions of $\boldsymbol{b}_i$; fixing B, it is a strictly convex function of U. These convex functions have the general form $f(\boldsymbol{\xi}) = \sum_i \exp(\boldsymbol{\alpha}_i^\top\boldsymbol{\xi} + \beta_i) + \boldsymbol{\zeta}^\top\boldsymbol{\xi}$. Its gradient and Hessian are both in simple closed form: $\triangledown f = \sum_i \exp(\boldsymbol{\alpha}_i^\top\boldsymbol{\xi}+\beta_i)\boldsymbol{\alpha}_i+\boldsymbol{\zeta}$; $\triangledown^2 f = \sum_i \exp(2\boldsymbol{\alpha}_i^\top\boldsymbol{\xi}+2\beta_i)\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^\top$. One can apply an off-the-shelf convex optimizer, which in the simplest case can be the Newton method, to alternately minimize B and U until convergence. Our implementation simply uses L-BFGS [9] based on the gradient of the loss. In summary, we have the following result.

Proposition 6 *The* CoDA-PCA *and the* s-CoDA-PCA *are both equivalent to an unconstrained bi-convex optimization problem.*

As an alternative implementation, we assume a parametric mapping $\boldsymbol{b}_i = g_\Theta(\boldsymbol{x}_i)$ that is the ℓ-dimensional output of a feed-forward neural network with input $\boldsymbol{c}_{\mathrm{KL}}(\boldsymbol{x}_i)$ and $\boldsymbol{x}_i$ (or $\check{\boldsymbol{x}}_i$) and connection weights Θ. Then we minimize the cost function in (18) with respect to U and Θ. If g_Θ is flexible enough, then the minimization recovers the CoDA-PCA projection. This approach could be favored as ① it learns an out-of-sample mapping $g_\Theta(\cdot)$ with a compact parametric structure that does not scale with the sample size m; and ② it can be adapted to an online learning scenario. However, it requires tuning of the neural network architecture and the optimizer. In our experiments, the encoding map is modeled by a feed-forward neural network with two hidden layers of ELU [11] units, each of size 100. To distinguish between the two implementations, the method to directly optimize U and B

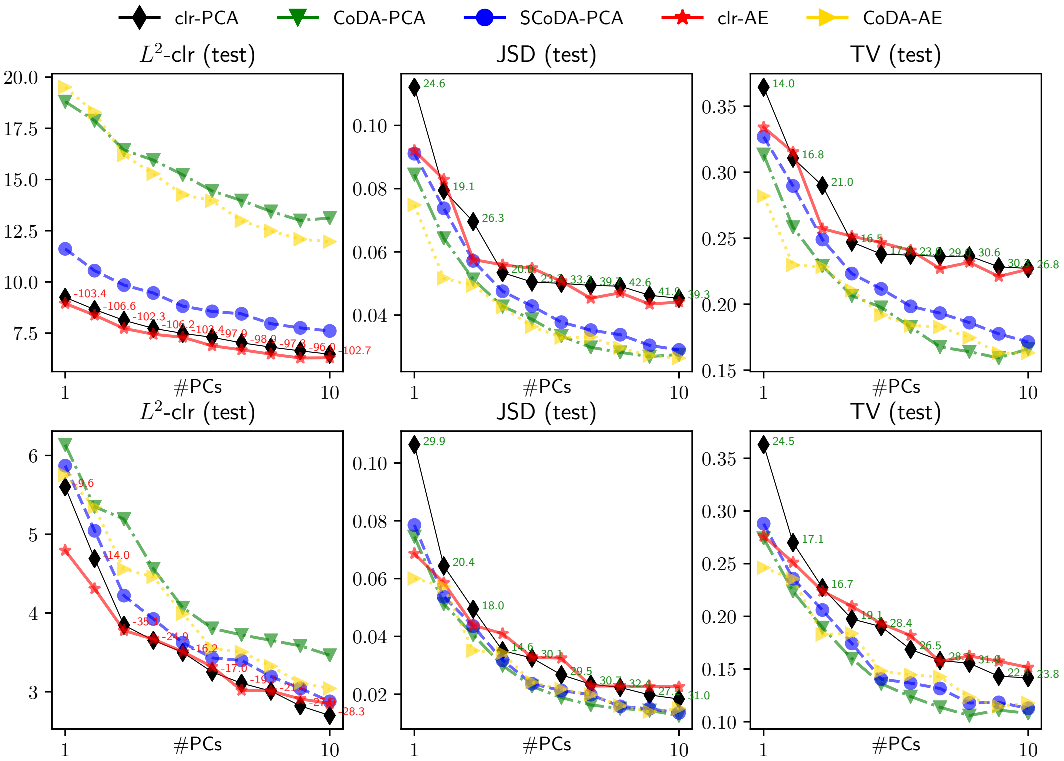

Figure 2: Testing errors (y-axis) against the number of principal components (x-axis) based on three different distance measures (from left to right) on the Atlas data (first row) and the diet swap data (second row). The numbers along the `clr-PCA` curves show the percentage of improvement (green) or disimprovement (red), comparing `CoDA-PCA` against `clr-PCA`.

without assuming the neural network mapping is called non-parametric CoDA-PCA, and the latter parametric version is simply called CoDA-PCA.

The above implementation resembles an auto-encoder structure: $\boldsymbol{x}_i \xrightarrow{\Theta} \boldsymbol{b}_i \xrightarrow{\mathsf{U}} \boldsymbol{y}_i$, where the decoder is simply a linear mapping $\boldsymbol{y}_i = \mathsf{C}\mathsf{U}^\intercal \boldsymbol{b}_i$. In the general case, we apply a non-linear decoder $\boldsymbol{b}_i \xrightarrow{\Phi} \boldsymbol{y}_i$ in the form $\boldsymbol{y}_i = \mathsf{C}h_\Phi(\boldsymbol{b}_i)$, where $h_\Phi(\cdot)$ is a neural network with parameters Φ and d output dimensions. At the same time, we add a small random noise to the encoder input so as to avoid overfitting. In this way we obtain a denoising CoDA-Autoencoder. In contrast to the CoDA-PCA, the CoDA-Autoencoder can only be trained by gradient-based optimizers.

In practice, the input matrix X may contain zeros that lie on the boundary of $\mathfrak{S}^d$. In this case $\boldsymbol{c}_{\text{KL}}(\boldsymbol{x})$ and $\check{\boldsymbol{x}}$ are undefined. A simple way to tackle the zero entries is to replace them with a small positive number $\epsilon > 0$. Alternatively, one can redefine the gauge as $g(\boldsymbol{x}) = \prod_{j:x_j>0}(x_j)^{1/\rho}$, where $\rho = |\{j : x_j > 0\}|$ so that $g(\boldsymbol{x})$ is always positive and $\check{\boldsymbol{x}}$ is well defined on $\mathfrak{S}^d \cup \partial\mathfrak{S}^d$.

6 Experiments

We compare the following methods: `clr-PCA` means PCA applied on the centered log-ratio coordinates; `CoDA-PCA` is the proposed CoDA-PCA in (11); `SCoDA-PCA` is the proposed S-CoDA-PCA in (17); `clr-AE` is an autoencoder with L^2 loss applied on the clr transformation; `CoDA-AE` is the proposed CoDA-Autoencoder in subsection 5. Both `clr-AE` and `CoDA-AE` use exactly the same structure with one hidden layer of 100 ELU [11] units in their decoders.

The baselines are assessed based on an array of measures including (L^2-clr) the L^2-distance $\|\boldsymbol{c}_{\text{KL}}(\boldsymbol{x}) - \boldsymbol{c}_{\text{KL}}(\boldsymbol{x}')\|_{\text{F}}$ between the input data $\boldsymbol{x} \in \mathfrak{S}^d$ and the reconstruction $\boldsymbol{x}' \in \mathfrak{S}^d$ in the clr space; (JSD) the Jensen-Shannon divergence $\frac{1}{2}\text{KL}(\boldsymbol{x} : \frac{\boldsymbol{x}+\boldsymbol{x}'}{2}) + \frac{1}{2}\text{KL}(\boldsymbol{x}' : \frac{\boldsymbol{x}+\boldsymbol{x}'}{2})$; (TV) the total variation distance

$\frac{1}{2}\sum_{i=1}^{d}|x_i - x'_i|$. These measurements are all invariant to scaling or permutation of $\boldsymbol{x}$ and $\boldsymbol{x}'$. See the supplementary material for more baselines and performance indicators.

We consider the following datasets available in the microbiome R package [18], each of which is randomly split into a training set (90%) and a testing set (10%). *The HITChip Atlas dataset [17]* contains 130 genus-level taxonomic groups that cover the majority of the known bacterial diversity of the human intestine. The data come from 1006 western adults from 15 western countries (Europe and the United States). Sample sets were analysed with three different DNA extraction methods. *The two-week diet swap study* between western (USA) and traditional (rural Africa) diets was reported in [28]. In this study, a two-week food exchange was performed in subjects from the same populations, where African Americans were fed a high-fibre, low-fat African-style diet and rural Africans a high-fat, low-fibre western-style diet. The group diet was indicated by HE (home environment days), DI (dietary intervention days) and ED (initial and final endoscopy days). Each subject served as his/her own control, given the known wide individual variation in colonic microbiota composition.

Fig. 2 shows the typical testing results. We observe that on most performance indicators `CoDA-PCA` and `CoDA-AE` show a much smaller testing error as compared to `clr-PCA` and `clr-AE`, respectively. The only exception is on L^2-clr, where `clr-PCA` and `clr-AE` appear to be favored against our `CoDA` variants. This is because L^2-clr is exactly the cost function of those two methods. We found that `CoDA-AE` is more robust against overfitting as compared to `clr-AE`. The performance of `SCoDA-PCA` is close to `CoDA-PCA` on most of the indicators and is better than `CoDA-PCA` on L^2-clr.

The source codes to reproduce our experimental results are available online[2].

7 Conclusion

We propose an approach for learning a low dimensional representation directly on raw count data, which is compositional in nature. Our proposed algorithm generalizes PCA in two ways, first by going to the exponential family via the Bregman divergence, and second by converting the normalization of data to a change in the Bregman divergence. The key theorem used for transforming the Bregman divergence generalizes a recent result, and may be of independent interest.

Acknowledgements

The authors gratefully thank Perrine Soret, Frank Nielsen, Xinhua Zhang, and the anonymous NIPS reviewers, for their helpful and constructive feedback. This work was done while MAF was visiting Data61, CSIRO in Canberra, Australia.

References

[1] J. Aitchison. The statistical analysis of compositional data (with discussion). *Journal of the Royal Statistical Society B*, 44(2):139–177, 1982.

[2] J. Aitchison. Principal component analysis of compositional data. *Biometrika*, 70(1):57–65, 1983.

[3] J. Aitchison. *The Statistical Analysis of Compositional Data*. Chapman and Hall, New York, 1986.

[4] J. Aitchison. Principles of compositional data analysis. *Multivariate Analysis and its Applications*, 24:73–81, 1994.

[5] J. Aitchison and J.-J. Egozcue. Compositional data analysis: Where are we and where should we be heading? *Mathematical Geology Journal*, 37:829–850, 2005.

[6] S.-I. Amari. *Information Geometry and Its Applications*. Springer-Verlag, Berlin, 2016.

[7] O. Barndorff-Nielsen. *Information and Exponential Families in Statistical Theory*. Wiley Publishers, 1978.

[2] `https://bitbucket.org/RichardNock/coda`

[8] J.-D. Boissonnat, F. Nielsen, and R. Nock. Bregman Voronoi diagrams. *Discrete Comput. Geom.*, 44(2):281–307, 2010.

[9] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[10] J. Chiquet, M. Mariadassous, and Stéphane Robin. Variational inference for probabilistic Poisson PCA. *Annals of Applied Statistics (to appear)*, 2018.

[11] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *4^{th} ICLR*, 2016.

[12] M. Collins, S. DasGupta, and R. Schapire. A generalization of principal components analysis to the exponential family. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *NIPS*15*, 2002.

[13] J.-J. Egozcue, V. Pawlowsky-Glahn, G. Mateu-Figueras, and C. Barceló-Vidal. Isometric logratio transformations for compositional data analysis. *Mathematical Geology Journal*, 35:279–300, 2003.

[14] G. Reid G. B. Gloor. Compositional analysis: a valid approach to analyze microbiome high-throughput sequencing data. *Can J Microbiol.*, 12:1–12, 2016.

[15] G. B. Gloor, J. R. Wu, V. Pawlowsky-Glahn, and J. J. Egozcue. It's all relative: analyzing microbiome data as compositions. *Ann Epidemiol.*, 26:322–9, 2016.

[16] M. Greenacre. *Compositional Data Analysis in Practice*. Chapman and Hall, New York, 2018.

[17] L. Lahti, J. Salojärvi, A. Salonen, M. Scheffer, and W. M. de Vos. Tipping elements in the human intestinal ecosystem. *Nat. Commun.*, 5:4344, 2014.

[18] L. Lahti, S. Sudarshan, T. Blake, and J. Salojarvi. Microbiome R package. version 1.1.10012, 2017.

[19] A. J. Landgraf. *Generalized Principal Component Analysis: Dimensionality Reduction through the Projection of Natural Parameters*. PhD thesis, Ohio State University, 2015.

[20] D. Lovell, W. Müller, J. Taylor, A. Zwart, and C. Helliwell. Caution! compositions! can constraints on omics data lead analyses astray? Technical Report EP10994, CSIRO, 2010.

[21] A.F. Andersson L.W. Hugerth. Analysing microbial community composition through amplicon sequencing: From sampling to hypothesis testing. *Frontiers in Microbiology*, 8:1561, 2017.

[22] P. Maréchal. On a functional operation generating convex functions, part 1: duality. *J. of Optimization Theory and Applications*, 126:175–189, 2005.

[23] P. Maréchal. On a functional operation generating convex functions, part 2: algebraic properties. *J. of Optimization Theory and Applications*, 126:375–366, 2005.

[24] J. A. Martín-Fernández, V. Pawlowsky-Glahn, J. J. Egozcue, and R. Tolosona-Delgado. The statistical analysis of compositional data (with discussion). *Math Geosci*, 50:273–298, 2018.

[25] R. Nock, A.-K. Menon, and C.-S. Ong. A scaled Bregman theorem with applications. In *NIPS*29*, pages 19–27, 2016.

[26] R. Nock, F. Nielsen, and S.-I. Amari. On conformal divergences and their population minimizers. *IEEE Trans. IT*, 62:1–12, 2016.

[27] V. Shankar O. Paliy. Application of multivariate statistical techniques in microbial ecology. *Molecular ecology*, 25:1032–57, 2016.

[28] S. J. D. O'Keefe, J. V. Li, L. Lahti, J. Ou, F. Carbonero, K. Mohammed, J. M. Posma, J. Kinross, E. Wahl, E. Ruder, K. Vipperla, V. Naidoo, L. Mtshali, S. Tims, P. G. B. Puylaert, J. DeLany, A. Krasinskas, A. C. Benefiel, H. O. Kaseb, K. Newton, J. K. Nicholson, W. M. de Vos, H. R. Gaskins, and E. G. Zoetendal. Fat, fiber and cancer risk in African Americans and rural Africans. *Nat. Commun.*, 6:6342, 2015.

[29] V. Pawlowsky-Glahn and A. Buccianti. *Compositional Data Analysis, theory and applications*. Wiley, 2011.

[30] Ke Sun and Stéphane Marchand-Maillet. An information geometry of statistical manifold learning. In *31st ICML*, pages 1–9, 2014.

[31] R. Tolosona-Delgado V. Pawlowsky-Glahn, J. J. Egozcue. Lecture notes on compositional data analysis. Technical report, Girona University, 2007.

[32] L. van der Maaten and G. E. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

A Convexity of divergence

Theorem 7 *For any* A, V *such that* $\mathsf{A}^\top \mathsf{V} \mathbf{1} = \mathbf{0}$, *letting* $D_{\exp}$ *the Bregman divergence with generator* $\varphi^\star(z) \doteq \exp z$ *and* KL *the generator for KL divergence,*

$$D_{\exp}(\mathsf{V}^\top \boldsymbol{a}_i \| \boldsymbol{c}_{\mathrm{KL}}(\boldsymbol{x}_i)) \quad = \quad q_i \cdot D_{\check{\mathrm{KL}}}(\boldsymbol{x}_i \| \exp(\mathsf{V}^\top \boldsymbol{a}_i)) + r_i \cdot D_g(\boldsymbol{x}_i \| \exp(\mathsf{V}^\top \boldsymbol{a}_i)), \forall i = 1, 2, ..., m) \qquad (20)$$

with $q_i \doteq 1/g(\boldsymbol{x}_i)$ *and* $r_i \doteq q_i \cdot \sum_j \exp\left(\boldsymbol{a}_i^\top \boldsymbol{v}_j\right)$, *satisfying* $r_i \geq d q_i$.

(Proof in SM, Section B)

We can remark that

$$\frac{1}{g(\boldsymbol{x}_i)} \cdot \sum_j \exp\left(\boldsymbol{a}_i^\top \boldsymbol{v}_j\right) \quad = \quad \sum_j \exp\left(\boldsymbol{a}_i^\top \boldsymbol{v}_j - x_{ij}\right) \check{x}_{ij}. \qquad (21)$$

To generalize the notion of geometric average, consider any strictly convex function $\varphi : \mathbb{R} \to \mathbb{R}$ at least three times differentiable and with invertible derivative. Define the φ-mean of $\boldsymbol{x}$ as

$$\mu_\varphi(\boldsymbol{x}) \quad \doteq \quad \varphi'^{-1}\left(\mathsf{E}_i \varphi'(x_i)\right). \qquad (22)$$

We now state a Lemma which will be essentially used for a particular case of φ but may be of independent interest for more general cases, as explained in the following examples.

Lemma 8 *Let* φ *convex and at least three times differentiable. Let*

$$\phi(x) \quad \doteq \quad -\frac{\varphi'''(x)}{(\varphi''(x))^2} \quad . \qquad (23)$$

Then the φ'*-mean* μ_φ *is convex (resp. concave) iff*

$$\phi(\mu_\varphi(\boldsymbol{x})) \quad \geq \textit{ (resp. } \leq \textit{)} \quad m \cdot \min_i \phi(x_i) \ \ , \forall \boldsymbol{x} \ \ . \qquad (24)$$

(Proof in SM, Section C)

Example 1 *Take for example* $F'(x) \doteq \ln x$ *(geometric average). In this case,* $\phi(x) = +1$ *and ineq. (24) brings* $1 \leq m$ *for the concavity of the mean and shows that* g *is concave.*

Example 2 *Consider* $F'(x) = -1/x$ *(harmonic average, over* $\mathbb{R}_+$*). In this case,* $\phi(x) = +x$*, so to prove the concavity of the mean, we need to show* $\mu_F(\boldsymbol{x}) \leq m \min_i x_i$*, which is equivalent to show*

$$\sum_i \frac{1}{x_i} \quad \geq \quad \max_i \frac{1}{x_i} \quad , \qquad (25)$$

and shows the concavity of the mean.

Example 3 *Consider* $F' = \exp$ *(normalized softmax). In this case,* $\phi(x) = -\exp(-x)$*, which shows that the mean cannot be concave. To show its convexity, we need to show equivalently*

$$\frac{1}{\sum_i \exp x_i} \quad \leq \quad \exp(-\max_i x_i) \quad , \qquad (26)$$

which is equivalent to $\sum_i \exp(x_i - \max_j x_j) \geq 1$*, and because of the non-negativity of the exponential, shows the convexity of the mean.*

B Proof of Theorem 7

First, letting $\boldsymbol{y}_i \doteq \exp(\mathsf{V}^\top \boldsymbol{a}_i)$ for any $i = 1, 2, ..., m$, and $\boldsymbol{v}_j$ the j-th row of V (as a column vector), we get

$$\varphi^\star\left(\nabla\varphi(\check{\boldsymbol{y}}_i)\right) \quad = \quad \sum_j \exp\left(\log\left(\exp\left(\boldsymbol{a}_i^\top \boldsymbol{v}_j\right)\right)\right) = \sum_j \exp\left(\boldsymbol{a}_i^\top \boldsymbol{v}_j\right). \qquad (27)$$

We recall $g(\boldsymbol{x}) = (\prod_k x_k)^{1/d}$, the geometric mean. So,

$$g(\exp(\mathsf{V}^\top \boldsymbol{a}_i)) \quad = \quad \exp\left(\frac{1}{d}\sum_j \boldsymbol{a}_i^\top \boldsymbol{v}_j\right) = \exp((\mathsf{A}^\top \mathsf{V}\mathbf{1})_i) = \exp(0) = 1, \tag{28}$$

because of the constraint $\mathsf{A}^\top \mathsf{V}\mathbf{1} = \mathbf{0}_m$. Hence, letting $\boldsymbol{y}_i \doteq \exp(\mathsf{V}^\top \boldsymbol{a}_i)$ for short, we obtain $\check{\boldsymbol{y}}_i = \boldsymbol{y}_i$. Also, the dual symmetry of Bregman divergences [8] yields:

$$D_{\exp}(\mathsf{V}^\top \boldsymbol{a}_i \| \boldsymbol{c}_{\text{KL}}(\boldsymbol{x}_i)) \quad = \quad D_{\text{KL}}(\check{\boldsymbol{x}}_i \| \exp(\mathsf{V}^\top \boldsymbol{a}_i)) \ . \tag{29}$$

We now use Theorem 2. It comes (letting again $\boldsymbol{y}_i \doteq \exp(\mathsf{V}^\top \boldsymbol{a}_i)$ for short) that for any $i = 1, 2, ..., m$,

$$\begin{aligned}
&\ g(\boldsymbol{x}_i) \cdot D_{\exp}(\mathsf{V}^\top \boldsymbol{a}_i \| \boldsymbol{c}_{\text{KL}}(\boldsymbol{x}_i)) \\
&= g(\boldsymbol{x}_i) \cdot D_{\text{KL}}(\check{\boldsymbol{x}}_i \| \exp(\mathsf{V}^\top \boldsymbol{a}_i)) \\
&= D_{\check{\text{KL}}}\left(\boldsymbol{x}_i \ \| \ \exp(\mathsf{V}^\top \boldsymbol{a}_i)\right) + \varphi^\star\left(\nabla\varphi(\check{\boldsymbol{y}}_i)\right) D_g(\boldsymbol{x}_i \| \boldsymbol{y}_i) \\
&= D_{\check{\text{KL}}}\left(\boldsymbol{x}_i \ \| \ \exp(\mathsf{V}^\top \boldsymbol{a}_i)\right) + \left(\sum_j \exp\left(\boldsymbol{a}_i^\top \boldsymbol{v}_j\right)\right) \cdot D_g(\boldsymbol{x}_i \| \exp(\mathsf{V}^\top \boldsymbol{a}_i)) \ ,
\end{aligned} \tag{30}$$

which brings the statement of the Theorem. The fact that $r_i \geq dq_i$ follows from the convexity of exp.

C Proof of Lemma 8

Let $\mu_\varphi(\boldsymbol{x}) \doteq \varphi'^{-1}((1/m) \cdot \sum_i \varphi'(x_i))$ the φ'-mean of the coordinates of $\boldsymbol{x}$. Then

$$\frac{\partial}{\partial x_i}\mu_\varphi(\boldsymbol{x}) \quad = \quad \frac{1}{m} \cdot \frac{\varphi''(x_i)}{\varphi''(\mu_\varphi(\boldsymbol{x}))} \ , \tag{31}$$

$$\frac{\partial^2}{\partial x_i \partial x_j}\mu_\varphi(\boldsymbol{x}) \quad = \quad \frac{1}{m\varphi''(\mu_\varphi(\boldsymbol{x}))} \cdot \left(\delta_{ij} \cdot \varphi'''(x_i) - \frac{\varphi'''(\mu_\varphi(\boldsymbol{x}))}{m(\varphi''(\mu_\varphi(\boldsymbol{x})))^2} \cdot \varphi''(x_i)\varphi''(x_j)\right) \tag{32}$$

$\boldsymbol{x}$ being fixed, let $\tilde{\boldsymbol{y}}$ the vector defined by $\tilde{y}_i \doteq y_i \cdot \varphi''(x_i)$ and let

$$\phi(x) \quad \doteq \quad -\frac{\varphi'''(x)}{(\varphi''(x))^2} \ . \tag{33}$$

Let Φ the diagonal matrix with $\Phi_{ii} \doteq \phi(x_i)$. The Hessian $\mathrm{H}_{\mu_\varphi(\boldsymbol{x})}$ satisfies

$$\begin{aligned}
\boldsymbol{y}^\top \mathrm{H}_{\mu_\varphi(\boldsymbol{x})} \boldsymbol{y} \quad &= \quad \frac{1}{m\varphi''(\mu_\varphi(\boldsymbol{x}))} \cdot \left(\frac{\phi(\mu_\varphi(\boldsymbol{x}))}{m} \cdot \tilde{\boldsymbol{y}}^\top \tilde{\boldsymbol{y}} - \tilde{\boldsymbol{y}}^\top \Phi \tilde{\boldsymbol{y}}\right) \\
&= \quad \frac{1}{m\varphi''(\mu_\varphi(\boldsymbol{x}))} \cdot \tilde{\boldsymbol{y}}^\top \mathrm{Diag}\left(\frac{\phi(\mu_\varphi(\boldsymbol{x}))}{m} \cdot \mathbf{1} - \boldsymbol{\phi}(\boldsymbol{x})\right) \tilde{\boldsymbol{y}}
\end{aligned} \tag{34}$$

We see that when φ'' is not zero everywhere, $\tilde{\boldsymbol{y}}$ spans the same set as $\boldsymbol{y}$, so if φ is convex, μ_φ is convex iff $(\phi(\mu_\varphi(\boldsymbol{x}))/m) \cdot \mathbf{1} - \boldsymbol{\phi}(\boldsymbol{x}) \geq \mathbf{0}$, that is

$$\phi(\mu_\varphi(\boldsymbol{x})) \quad \geq \quad m \cdot \phi(x_i) \ , \forall i \ , \tag{35}$$

which is equivalent to the Lemma's statement.

D Proof of Lemma 4

We have

$$\begin{aligned}
\check{\mathrm{KL}}(\boldsymbol{x}) &= \left(\sum_i x_i \log x_i - x_i\right) - \frac{1}{d} \cdot \sum_{ij} x_i \log x_j \ , && (36)\\
\frac{\partial}{\partial x_i}\check{\mathrm{KL}}(\boldsymbol{x}) &= \log x_i - \frac{1}{d} \cdot \sum_j \log x_j - \frac{1}{dx_i} \cdot \sum_j x_j && (37)\\
&= -\frac{1}{d} \cdot \left(\sum_j \frac{x_j}{x_i} + \log \frac{x_j}{x_i}\right) \ , && (38)\\
\frac{\partial^2}{\partial x_i \partial x_j}\check{\mathrm{KL}}(\boldsymbol{x}) &= -\frac{1}{d} \cdot \left(\frac{1}{x_i} + \frac{1}{x_j}\right), \forall j \neq i, && (39)\\
\frac{\partial^2}{\partial x_i^2}\check{\mathrm{KL}}(\boldsymbol{x}) &= \frac{1}{x_i} \cdot \left(1 - \frac{1}{d} + \frac{1}{d} \cdot \sum_{j \neq i} \frac{x_j}{x_i}\right) . && (40)
\end{aligned}$$

We then check that

$$(\mathrm{H}\boldsymbol{z})_i = \frac{z_i}{x_i} + \frac{1}{d} \cdot \sum_j \left(\frac{z_i x_j}{x_i^2} - \frac{z_j}{x_j} - \frac{z_j}{x_i}\right) \qquad (41)$$

and finally

$$\begin{aligned}
\boldsymbol{z}^\top \mathrm{H} \boldsymbol{z} &= \sum_i \frac{z_i^2}{x_i} + \frac{1}{d} \sum_{ij} \left(\frac{z_i^2 x_j}{x_i^2} - \frac{z_i z_j}{x_j} - \frac{z_i z_j}{x_i}\right)\\
&= \frac{1}{d} \sum_{ij} \left(\frac{z_i^2 x_i}{x_i^2} + \frac{z_i^2 x_j}{x_i^2} - \frac{z_i z_j}{x_j} - \frac{z_i z_j}{x_i}\right)\\
&= \frac{1}{d} \sum_{ij} \left(\frac{z_i^2 x_i x_j^2 + z_i^2 x_j^3 - z_i z_j x_i^2 x_j - z_i z_j x_i x_j^2}{x_i^2 x_j^2}\right)\\
&= \frac{1}{2d} \sum_{ij} \left(\frac{z_i^2 x_i x_j^2 + z_i^2 x_j^3 + z_j^2 x_i^2 x_j + z_j^2 x_i^3 - 2 z_i z_j x_i^2 x_j - 2 z_i z_j x_i x_j^2}{x_i^2 x_j^2}\right)\\
&= \frac{1}{2d} \sum_{ij} \left(\frac{z_i^2 x_j^2 (x_i + x_j) + z_j^2 x_i^2 (x_i + x_j) - 2 z_i z_j x_i x_j (x_i + x_j)}{x_i^2 x_j^2}\right)\\
&= \frac{1}{2d} \sum_{ij} (x_i + x_j) \cdot \left(\frac{z_i}{x_i} - \frac{z_j}{x_j}\right)^2 , && (42)
\end{aligned}$$

as claimed. This also shows the convexity of $\check{\mathrm{KL}}(\boldsymbol{x})$.

We now show that $\check{\mathrm{KL}} \circ \exp$ is 1-homogeneous on the subspace $\mathrm{span}(\{\boldsymbol{1}\})^\perp$, which means, by Euler's Theorem,

$$\check{\mathrm{KL}}(\exp(\boldsymbol{x})) = \exp(\boldsymbol{x})^\top \nabla \check{\mathrm{KL}}(\exp(\boldsymbol{x})), \forall \boldsymbol{x} \in \mathrm{span}(\{\boldsymbol{1}\})^\perp . \qquad (43)$$

To see this, we write

$$\begin{aligned}
&\check{\mathrm{KL}}(\exp(\boldsymbol{x})) - \exp(\boldsymbol{x})^\intercal \nabla \check{\mathrm{KL}}(\exp(\boldsymbol{x}))\\
=\;& \sum_j x_j \exp(x_j) - \exp(x_j)\\
&+\frac{1}{d}\cdot \sum_j \left(\exp(x_j)\cdot \sum_k \left(\exp(x_k - x_j) + (x_k - x_j)\right)\right)\\
=\;& \sum_j x_j\exp(x_j) \underbrace{-\sum_j \exp(x_j) + \sum_k \exp(x_k)}_{=0}\\
&+\frac{1}{d}\cdot\sum_j \exp(x_j)\underbrace{\sum_k x_k}_{=0} - \sum_j \exp(x_j)x_j\\
=\;& \sum_j x_j\exp(x_j) - \sum_j \exp(x_j)x_j\\
=\;& 0,
\end{aligned}\tag{44}$$

where we have used the fact that $\mathbf{1}^\intercal \boldsymbol{x} = 0$.

E Derivations of the unconstrained optimization in (18) and (19)

Consider that X is constant, therefore

$$\begin{aligned}
\ell_{\text{CoDA-PCA}}(\mathsf{X};\mathsf{A},\mathsf{V}) &= D_{\exp}(\mathsf{V}^\intercal\mathsf{A}\,\|\,\boldsymbol{c}_{\text{KL}}(\mathsf{X}))\\
&= \mathbf{1}^\intercal\left[\exp(\mathsf{V}^\intercal\mathsf{A}) - \exp(\boldsymbol{c}_{\text{KL}}(\mathsf{X})) - (\mathsf{V}^\intercal\mathsf{A} - \boldsymbol{c}_{\text{KL}}(\mathsf{X}))\circ\exp(\boldsymbol{c}_{\text{KL}}(\mathsf{X}))\right]\mathbf{1}\\
&= \mathbf{1}^\intercal\left[\exp(\mathsf{V}^\intercal\mathsf{A}) - \mathsf{V}^\intercal\mathsf{A}\circ\exp(\boldsymbol{c}_{\text{KL}}(\mathsf{X}))\right]\mathbf{1} + \text{constant}\\
&= \mathbf{1}^\intercal\left[\exp(\mathsf{V}^\intercal\mathsf{A}) - \mathsf{V}^\intercal\mathsf{A}\circ\check{\mathsf{X}}\right]\mathbf{1} + \text{constant}.
\end{aligned}$$

Let $\mathsf{Y} = \mathsf{V}^\intercal\mathsf{A}$, and we get (18).

By (38), if $\boldsymbol{y}_i$ is centered and $\boldsymbol{y}_i^\intercal\mathbf{1} = 0$, we have

$$\begin{aligned}
\triangledown\check{\mathrm{KL}}(\exp(\boldsymbol{y}_i)) &= -\frac{1}{d}\cdot\left[\exp(\mathbf{1}\boldsymbol{y}_i^\intercal - \boldsymbol{y}_i\mathbf{1}^\intercal) + \mathbf{1}\boldsymbol{y}_i^\intercal - \boldsymbol{y}_i\mathbf{1}^\intercal\right]\mathbf{1}\\
&= -\frac{1}{d}\exp(\mathbf{1}\boldsymbol{y}_i^\intercal - \boldsymbol{y}_i\mathbf{1}^\intercal)\mathbf{1} + \boldsymbol{y}_i.
\end{aligned}$$

Therefore

$$\begin{aligned}
\ell_{\text{s-CoDA-PCA}}(\mathsf{X};\mathsf{A},\mathsf{V}) &= \sum_i\left[\check{\boldsymbol{x}}_i^\intercal\frac{1}{d}\exp(\mathbf{1}\boldsymbol{y}_i^\intercal - \boldsymbol{y}_i\mathbf{1}^\intercal)\mathbf{1} - \check{\boldsymbol{x}}_i^\intercal\boldsymbol{y}_i\right]\\
&= \sum_i\check{\boldsymbol{x}}_i^\intercal\left[\exp(-\boldsymbol{y}_i)\exp(\boldsymbol{y}_i^\intercal)\frac{\mathbf{1}}{d} - \boldsymbol{y}_i\right],
\end{aligned}$$

which, in matrix form, is (19).

F More Experimental Results

We include another baseline `CoDA-PCA`* that is the non-parametric version of `CoDA-PCA` optimized by L-BFGS. Note that `CoDA-PCA`* does not learn an encoding map and therefore cannot provide an out-of-sample extension.

The baselines are further assessed based on (symmetric perspective KL divergence; SPKL) normalizing the geometric average of the input data p and the PCA reconstruction q and get respectively two positive measures $\check{p}$ and $\check{q}$, then computing the symmetric KL divergence $\frac{1}{2}\sum_i \left(\check{p}_i \log \frac{\check{p}_i}{\check{q}_i} + \check{q}_i \log \frac{\check{q}_i}{\check{p}_i}\right)$; ($L^2$) L^2-distance of $\bar{p}$ and $\bar{q}$ after the normalizing p and q into the probability simplex. (Riemannian) the Riemannian distance between the two probabilities $\bar{p}$ and $\bar{q}$ defined by the Fisher information metric, given by $2 \arccos\left(\sqrt{\bar{p}}^{\intercal}\sqrt{\bar{q}}\right)$.

Fig. 3 shows the training and testing errors. A key observation is that `CoDA-PCA`* is slightly better than `CoDA-PCA` because the embedding points are free parameters and are not constrained by a neural network. We also see that `clr-AE` shows small training errors but does not generalize as well as `CoDA-AE` on the testing set.

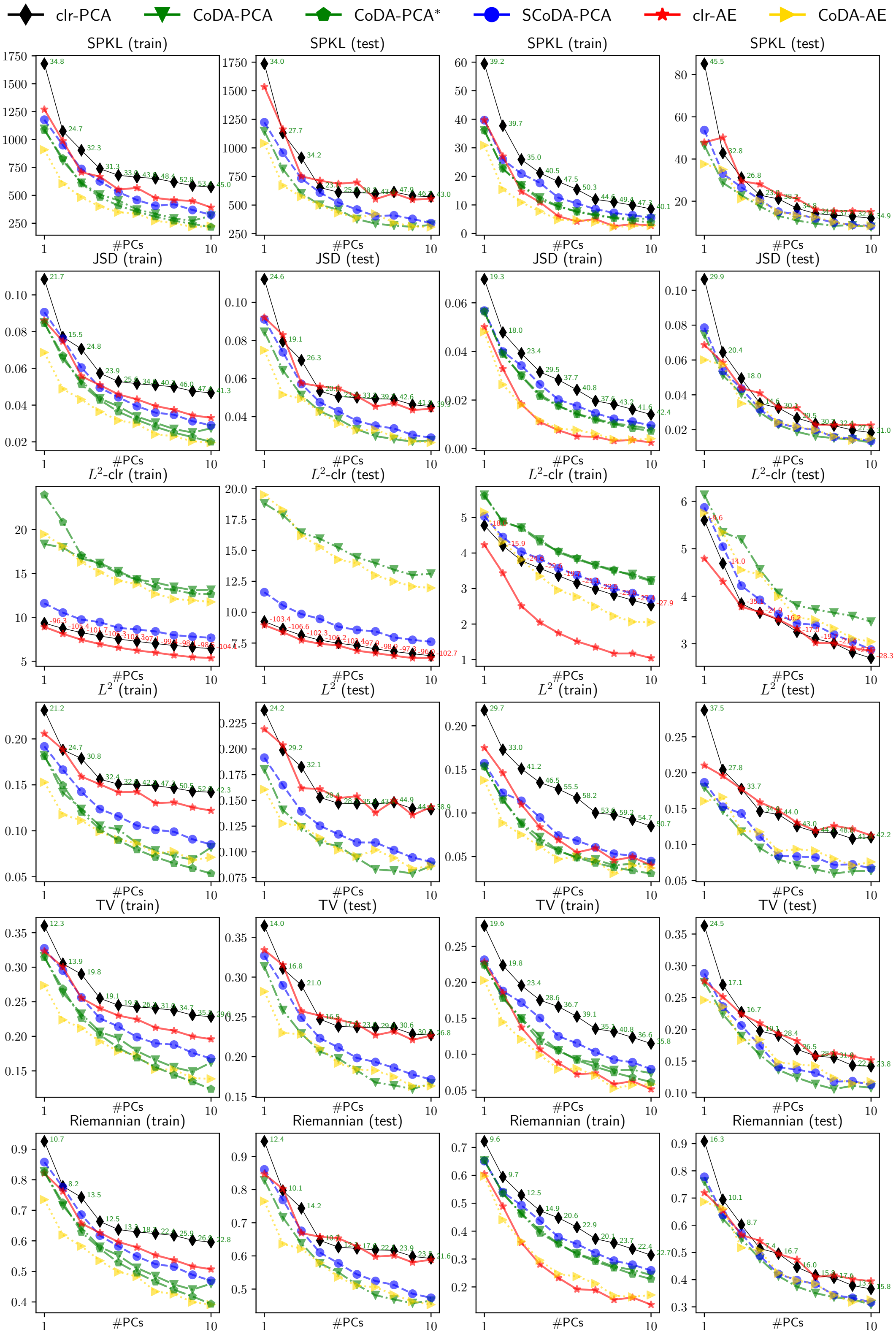

Figure 3: Training and testing errors measured by different distances against the number of principal components. The columns, from left to right, show training errors (Atlas), corresponding testing errors, training errors (diet swap) and corresponding test errors.