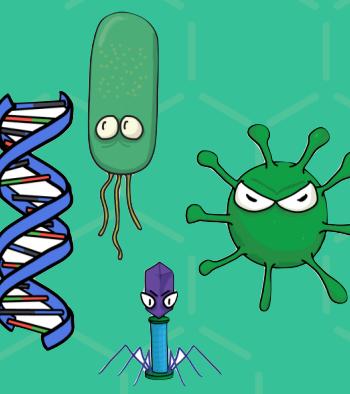


Representation Learning of Compositional Data



Marta Avalos-Fernandez, Richard Nock, Cheng Soon Ong, Julien Rouar, Ke Sun

tagline

Compositional Data analysis (CoDA) is a subfield of statistics analysing data on a simplex; clr-PCA is a standard CoDA tool for principal component analysis
We improve on clr-PCA from three standpoints: (i) Information Geometry (loss), (ii) Optimisation and (iii) Representation (deeper architectures)

Result of independent interest: **scaled Bregman Theorem (SBT) with remainder**

Definitions: let $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$ and $g : \mathbb{R}^d \rightarrow \mathbb{R}_*$ differentiable. Define **Bregman distortion** D_φ with generator φ : $D_\varphi(\mathbf{x} \parallel \mathbf{x}') \doteq \varphi(\mathbf{x}) - \varphi(\mathbf{x}') - (\mathbf{x} - \mathbf{x}')^\top \nabla \varphi(\mathbf{x}')$

(Bregman divergence if generator convex)

$$\text{Then for any } \mathbf{x}, \mathbf{y}, g(\mathbf{x}) \cdot D_\varphi(\check{\mathbf{x}} \parallel \check{\mathbf{y}}) = D_{\check{\varphi}}(\mathbf{x} \parallel \mathbf{y}) + \varphi^*(\nabla \varphi(\check{\mathbf{y}})) \cdot D_g(\mathbf{x} \parallel \mathbf{y})$$

(Bregman divergences if generators convex) remainder

generalizes Nock et al., NIPS'16

$$\check{\mathbf{x}} \doteq \frac{\mathbf{x}}{g(\mathbf{x})}, \varphi^* \text{ convex conjugate,} \\ \check{\varphi}(\mathbf{x}) \doteq g(\mathbf{x}) \cdot \varphi\left(\frac{\mathbf{x}}{g(\mathbf{x})}\right) \\ (\text{generalized perspective transform})$$

clr-PCA

Aitchison'86

(i) 2x data transformation (centered log ratio)

$$c_\varphi(\mathbf{x}) = \nabla \varphi\left(\frac{\mathbf{x}}{g(\mathbf{x})}\right) = C^{\text{clr}} \log(\mathbf{x}) = \log(\mathbf{x}) - \overline{\log(\mathbf{x})} \mathbf{1}_d$$

$$\varphi(\mathbf{x}) = \mathbf{x} \log \mathbf{x} - \mathbf{x}$$

$$\overline{\mathbf{x}} = \frac{1}{d} \sum_{j=1}^d x_j$$

$$C^{\text{clr}} = \mathbf{I}_d - \frac{1}{d} \mathbf{1}_d \mathbf{1}_d^\top$$

centering matrix

(other choices available)

(ii) Run PCA

$$\text{Min } \ell_{\text{PCA}}(c_\varphi(\mathbf{X}); \mathbf{A}, \mathbf{V}) \doteq \|c_\varphi(\mathbf{X}) - \mathbf{V}^\top \mathbf{A}\|_F^2$$

$$\text{s.t. } \mathbf{A} \in \mathbb{R}^{\ell \times m}, \mathbf{V} \in \mathbb{R}^{\ell \times d}, \mathbf{V} \mathbf{V}^\top = \mathbf{I}_\ell, \ell \ll d$$

CoDA-PCA

(Better metric characterisation)

(i) 1x data transformation

$$\check{\mathbf{x}} \doteq \frac{\mathbf{x}}{g(\mathbf{x})}, g(\mathbf{x}) = (\prod_{j=1}^d x_j)^{1/d}$$

(ii) Run constrained exponential families PCA

$$\text{Min } \ell_{\text{CoDA-PCA}}(\mathbf{X}; \mathbf{A}, \mathbf{V}) \doteq D_\varphi(\check{\mathbf{X}} \parallel \nabla \varphi^*(\mathbf{V}^\top \mathbf{A}))$$

$$\text{s.t. } \mathbf{A} \in \mathbb{R}^{\ell \times m}, \mathbf{V} \in \mathbb{R}^{\ell \times d}, \mathbf{V} \mathbf{V}^\top = \mathbf{I}_\ell, \mathbf{A}^\top \mathbf{V} \mathbf{1} = \mathbf{0}$$

Collins et al., NIPS'02

Better Information Geometry / Optimisation

Use SBT with remainder for CoDA-PCA: obtain a **surrogate loss** simple to optimise

$$\mathbf{A}^\top \mathbf{V} \mathbf{1} = \mathbf{0} \Rightarrow \ell_{\text{CoDA-PCA}}(\mathbf{X}; \mathbf{A}, \mathbf{V}) \leq \text{CST} - \sum_i \check{\mathbf{x}}_i^\top \nabla \text{KL}(\exp(\mathbf{V}^\top \mathbf{a}_i))$$

Inner product!

sCoDA-PCA

(i) 1x data transformation

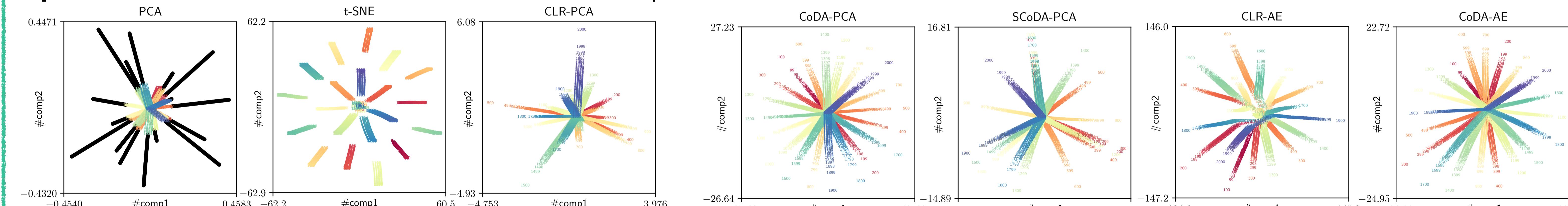
$$\check{\mathbf{x}} \doteq \frac{\mathbf{x}}{g(\mathbf{x})}, g(\mathbf{x}) = (\prod_{j=1}^d x_j)^{1/d}$$

(ii) Minimize constrained surrogate loss

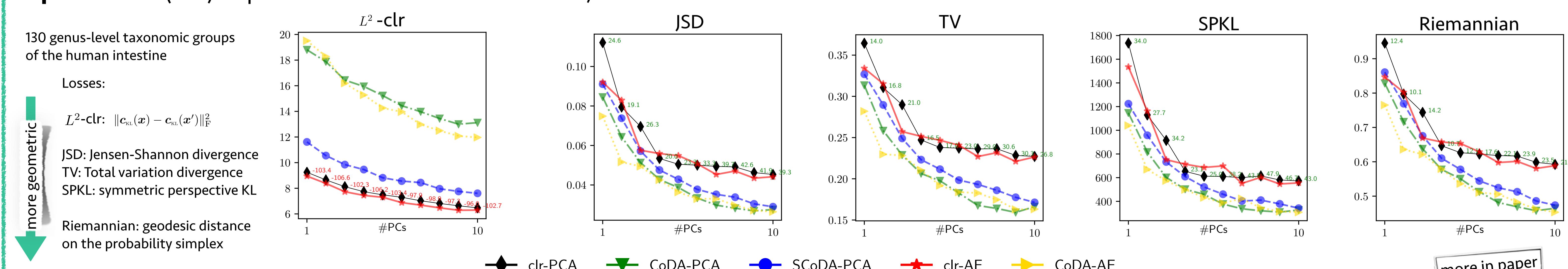
$$\text{Min } \ell_{\text{s-CoDA-PCA}}(\mathbf{X}; \mathbf{A}, \mathbf{V}) \doteq - \sum_i \check{\mathbf{x}}_i^\top \nabla \text{KL}(\exp(\mathbf{V}^\top \mathbf{a}_i))$$

$$\text{s.t. } \mathbf{A} \in \mathbb{R}^{\ell \times m}, \mathbf{V} \in \mathbb{R}^{\ell \times d}, \mathbf{V} \mathbf{V}^\top = \mathbf{I}_\ell, \mathbf{A}^\top \mathbf{V} \mathbf{1} = \mathbf{0}$$

Experiment – 20 arms from the center to the vertices of the simplex in \mathbb{R}^{20}



Experiment – (HIT)Chip Atlas dataset

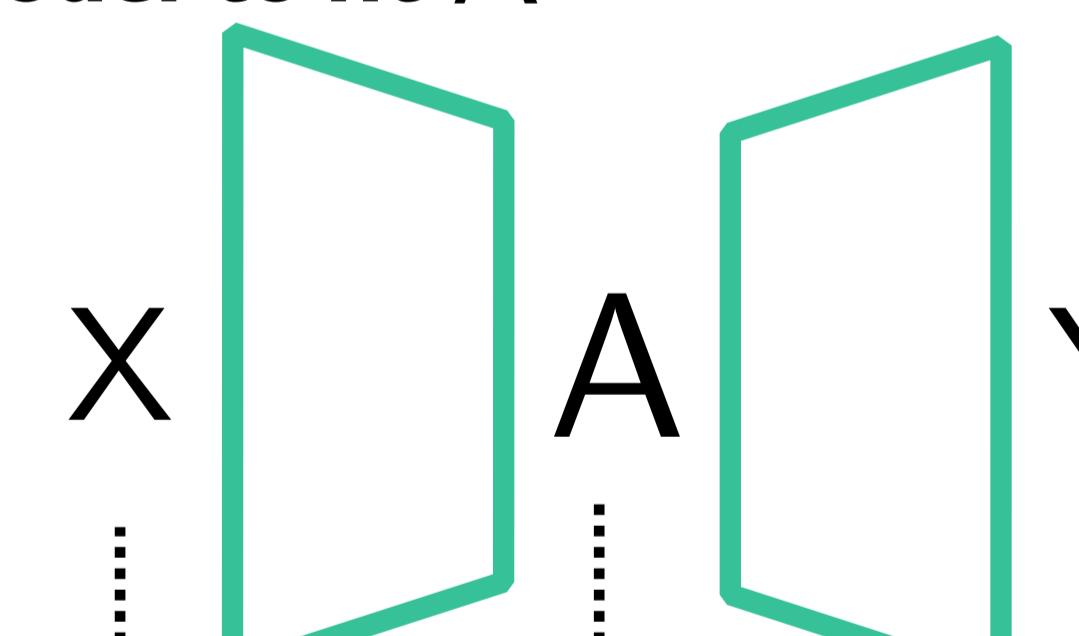


(iii) Use an autoencoder to fit \mathbf{A}

L: linear

NL: non linear

(NL: 1 hidden layer w/ ELU units)



clr-PCA

CoDA-PCA

sCoDA-PCA

clr-AE

CoDA-AE

"Deeper" representations