Density Separation with Tensor Factorization

2 Dec 2024 | CMS 2024 | Richmond, BC

Nicholas Richardson
Department of Mathematics

Naomi Graham
Department of Computer Science

Michael P. Friedlander
Departments of Computer Science & Mathematics

THE UNIVERSITY OF BRITISH COLUMBIA

Canadian Mathematical Society
Société mathématique du Canada

INSTITUTE OF APPLIED MATHEMATICS
University of British Columbia

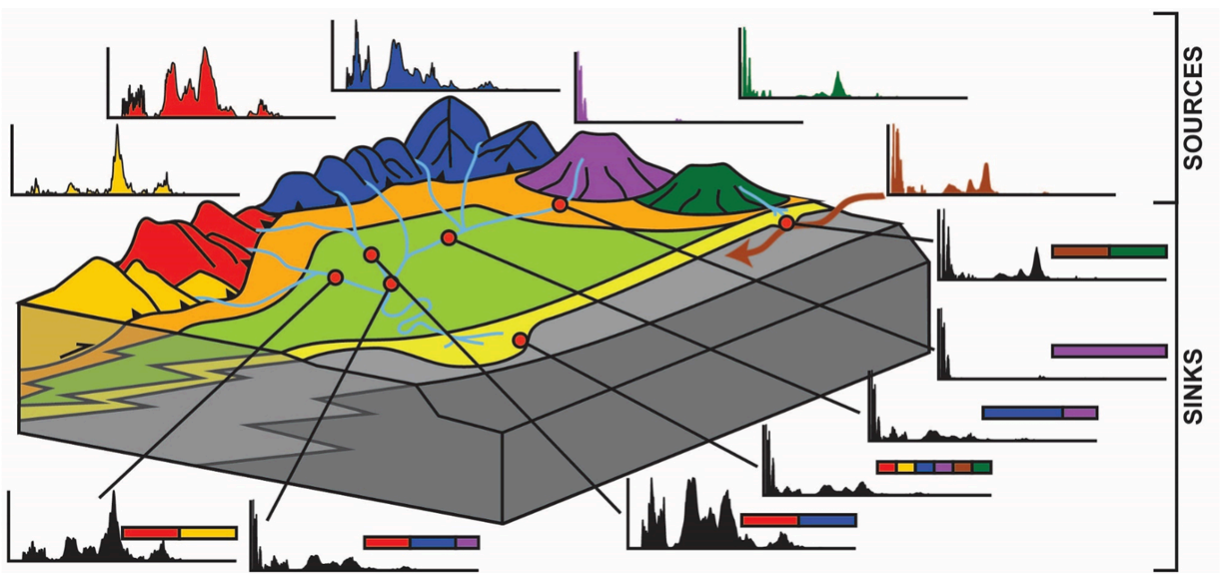
Talk Overview

- Motivation & Examples
- Generic Problem
- Solution Method
- Results

Motivation & Examples

Application 1: Geology[1]

- Source locations have their own distribution of minerals
- Rocks from these sources are mixed & deposited downstream
- Take scoops of rocks at locations downstream, called “sinks”

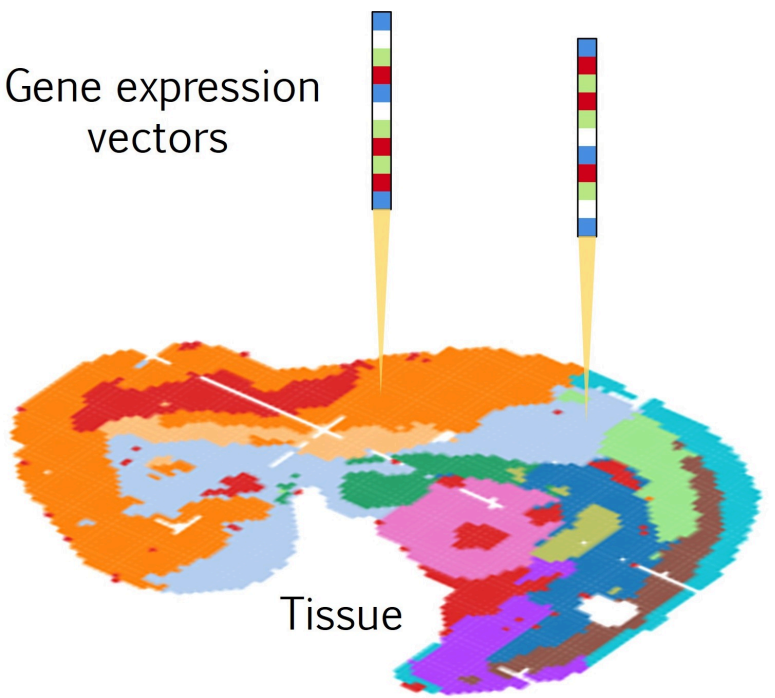


Modeling the Physical Problem

- Want to “un-mix” the measured sink distribution
- **Goal**: Estimate the mixing proportions and source distributions

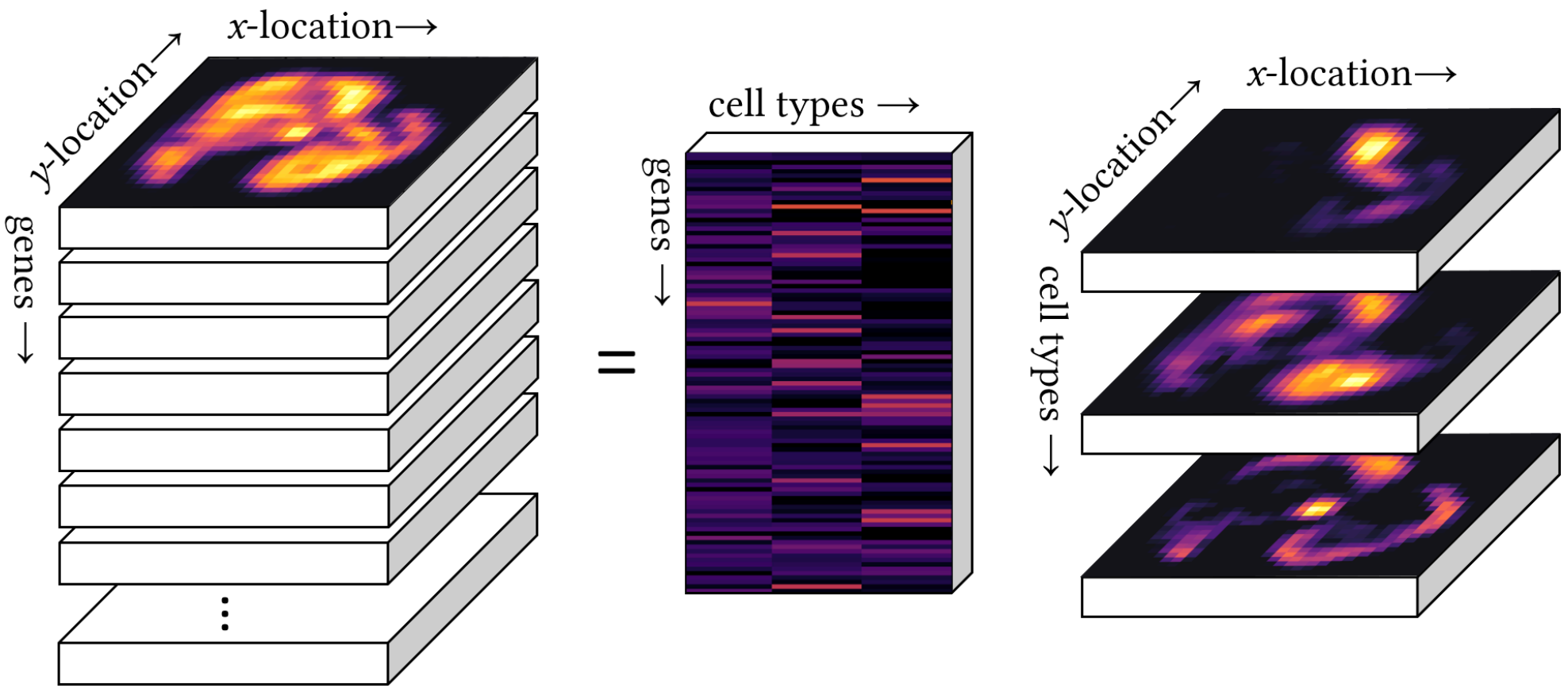
Application 2: Transcriptomics

- Perform single-cell sequencing for multiple embryos throughout development
- Label cells (heart, brain, etc.) at each time point
- **Goal**: Given gene expressions at each cell, cluster cells that have similar gene expressions



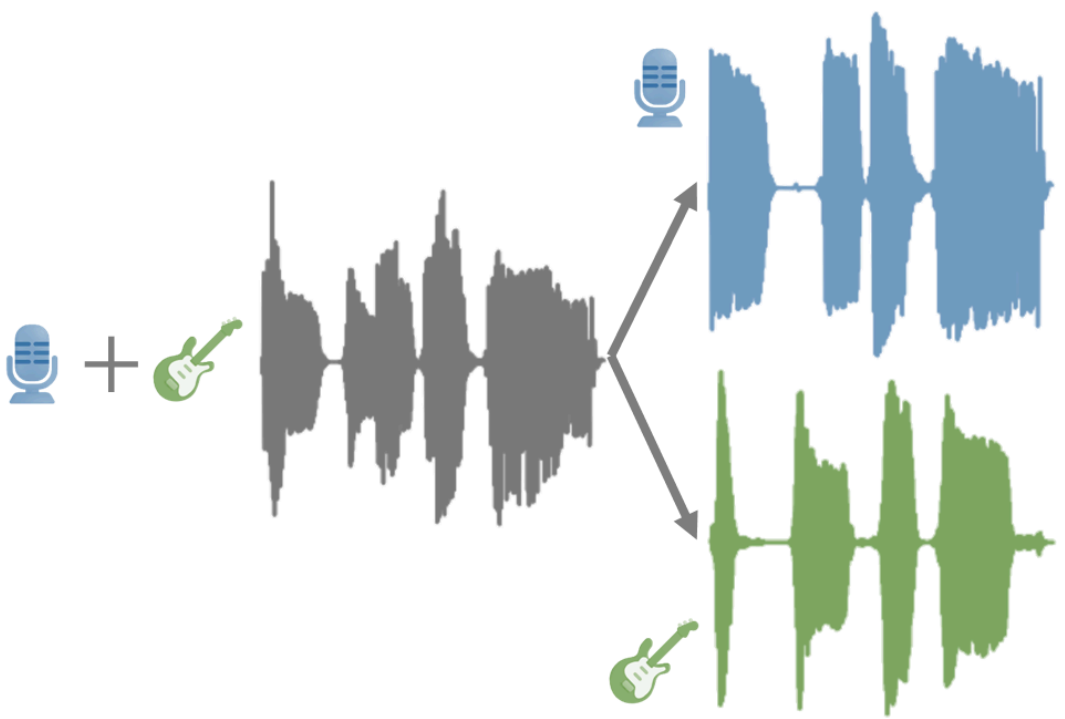
Modeling the problem

- **Goal**: Learn gene expressions and distribution of each cell type



Application 3: Music Decomposition

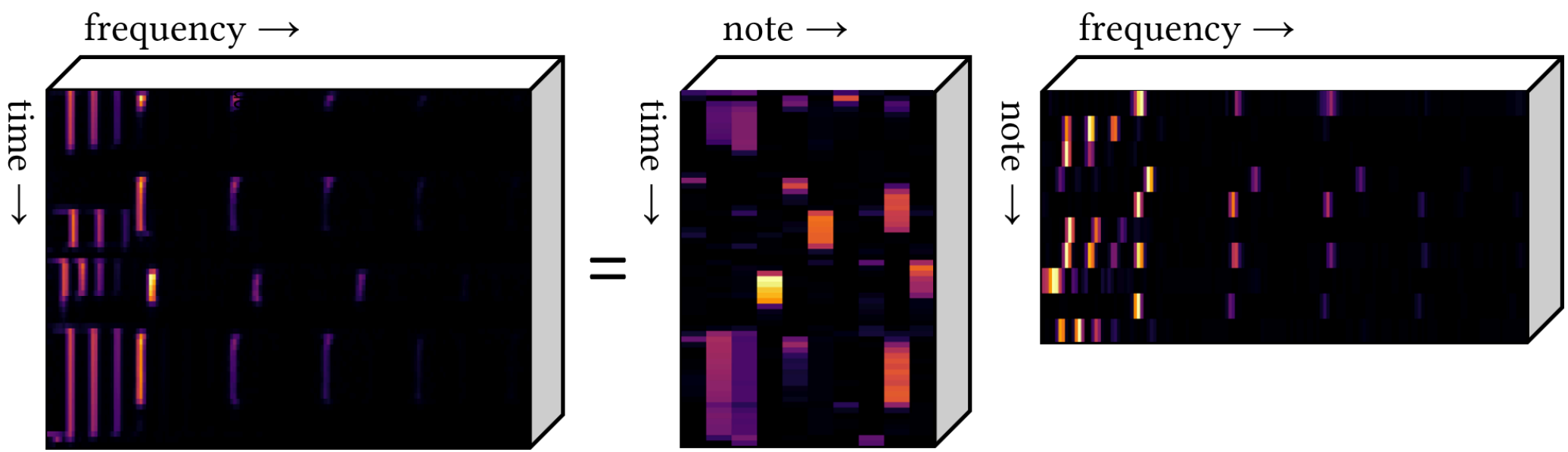
- Rebalance or isolate instruments from a single recording
- Learn what each note sounds like (frequency spectrum)
- **Goal**: Separate audio mixture by frequency spectrums



Density Separation with Tensor Factorization

Modeling Music Decomposition

- Take short-time Fourier transform of audio recording
- **Goal**: Learn the frequencies and amplitudes of each note



Generic Problem

Joint Signal Decomposition

- Given mixtures $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_I$
- Find mixing coefficients a_{ij} and sources $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_R$

$$
\begin{aligned}
\mathbf{y}_1 &= a_{11}\mathbf{b}_1 + a_{12}\mathbf{b}_2 + \cdots + a_{1R}\mathbf{b}_R \\
\mathbf{y}_2 &= a_{21}\mathbf{b}_1 + a_{22}\mathbf{b}_2 + \cdots + a_{2R}\mathbf{b}_R \\
&\vdots \\
\mathbf{y}_I &= a_{I1}\mathbf{b}_1 + a_{I2}\mathbf{b}_2 + \cdots + a_{IR}\mathbf{b}_R
\end{aligned}
$$

Other Decomposition Approaches

Parametrised Basis

- Fixed basis: $y = \sum_{b \in \mathcal{B}} a_b b$
  - Fourier and short-time Fourier transform (Gröchenig 2001)
  - Wavelets (Benedetto and Frazier 1993; Daubechies et al. 2011)
  - Frame decomposition (Benedetto and Frazier 1993, Ch. 7)
  - Atomic decomposition (Chen et al. 2001; Fan et al. 2022)
  - Dictionary learning* (Tošić and Frossard 2011)
- Random basis: $y = \sum_r a_r b_r$ where $b_r \sim \mathcal{B}$
  - Random feature models $b_r(x) = f(\langle \omega_r, x \rangle + \phi_r)$, where $(\omega_r, \phi_r) \sim \Omega \times \Phi$ (Rahimi and Recht 2008; Hashemi et al. 2023)
  - Sparse random mode decomposition (Richardson et al. 2023)

Other Decomposition Approaches

Parametrised Basis

- Great if you know a *good* basis for your data!
  - Data becomes sparse in your basis
  - Basis elements having meaningful interpretations (e.g. Fourier frequencies)
- Otherwise, sources b_r may not be meaningful

Other Decomposition Approaches

Data-driven

- Intrinsic Mode Functions: $y = \sum_r b_r$ where $b_r(t) = a_r(t)\sin(\phi_r(t))$
  - Empirical mode decomposition (Huang et al. 1998)
  - Empirical wavelet transform (Gilles 2013)
  - Variational mode decomposition (Dragomiretskiy and Zosso 2014)
- Supervised learning: learn parameters θ so that $F_\theta(\mathbf{y}) = (\mathbf{b}_r)_r$
  - Convolutional neural networks (Zhu et al. 2019)
  - Long short-term memory networks (Cao et al. 2019)
  - Autoencoders (Karamatlı et al. 2019)
- Dictionary learning (Tošić and Frossard 2011)

Other Decomposition Approaches

Data-driven

- Intrinsic Mode Functions
  - Effective for frequency-amplitude modulated sources b_r
- Supervised learning
  - ⭐ State-of-the-art ⭐
  - Need lots of data with known decompositions
  - Must retrain on data for each application

Joint Signal Decomposition

- Given mixtures $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_I$
- Find mixing coefficients a_{ij} and sources $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_R$

Joint Signal Decomposition

- Given mixtures $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_I$
- Find mixing coefficients a_{ij} and sources $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_R$

$$
\begin{aligned}
\mathbf{y}_1 &= a_{11}\mathbf{b}_1 + a_{12}\mathbf{b}_2 + \cdots + a_{1R}\mathbf{b}_R \\
\mathbf{y}_2 &= a_{21}\mathbf{b}_1 + a_{22}\mathbf{b}_2 + \cdots + a_{2R}\mathbf{b}_R \\
&\vdots \\
\mathbf{y}_I &= a_{I1}\mathbf{b}_1 + a_{I2}\mathbf{b}_2 + \cdots + a_{IR}\mathbf{b}_R
\end{aligned}
$$

Joint Signal Decomposition

- Given mixtures $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_I$
- Find mixing coefficients a_{ij} and sources $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_R$

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_I \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1R} \\ a_{21} & a_{22} & \ldots & a_{2R} \\ \vdots & & & \vdots \\ a_{I1} & a_{I2} & \ldots & a_{IR} \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_R \end{bmatrix}$$

Joint Signal Decomposition

- Given mixtures $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_I$
- Find mixing coefficients a_{ij} and sources $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_R$

$$\begin{bmatrix} \leftarrow & \mathbf{y}_1^\top & \rightarrow \\ \leftarrow & \mathbf{y}_2^\top & \rightarrow \\ \leftarrow & \vdots & \rightarrow \\ \leftarrow & \mathbf{y}_I^\top & \rightarrow \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1R} \\ a_{21} & a_{22} & \ldots & a_{2R} \\ \vdots & & & \vdots \\ a_{I1} & a_{I2} & \ldots & a_{IR} \end{bmatrix} \begin{bmatrix} \leftarrow & \mathbf{b}_1^\top & \rightarrow \\ \leftarrow & \mathbf{b}_2^\top & \rightarrow \\ & \vdots & \\ \leftarrow & \mathbf{b}_R^\top & \rightarrow \end{bmatrix}$$

Our approach

Matrix and Tensor Factorization

- Principle component analysis (Pearson 1901; Abdi and Williams 2010)
- Independent component analysis (Comon 1994; Hyvarinen 1999)
- Nonnegative matrix factorization (Cohen and Rothblum 1993; Gillis 2020)
- **Tensor decompositions** (Kolda and Bader 2009)
- Why?
  - Interpretability
  - Minimal assumptions on sources b_r
  - No training step, so less data needed
  - *Unsupervised* learning

Tucker-1 Decomposition

- A generalization of rank-R matrix factorization

- $\mathbf{Y} = \mathbf{AB}$
- $Y_{ijk} = \sum_{r=1}^{R} A_{ir} B_{rjk}$
- $\mathbf{Y}_i = \sum_{r=1}^{R} A_{ir} \mathbf{B}_r$

Tucker-1 Decomposition

- A generalization of rank-R matrix factorization

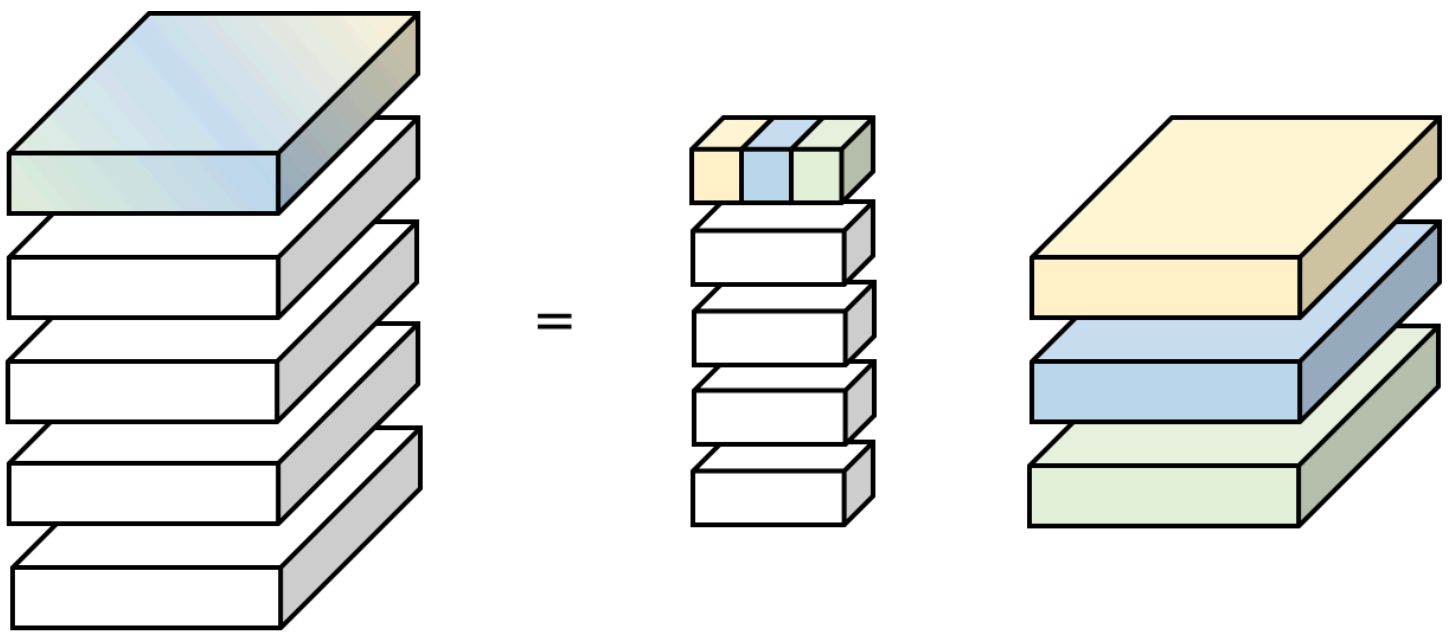
- $\mathbf{Y} = \mathbf{AB}$ • $Y_{ijk} = \sum_{r=1}^{R} A_{ir} B_{rjk}$ • $\mathbf{Y}_i = \sum_{r=1}^{R} A_{ir} \mathbf{B}_r$

$$\begin{bmatrix} \leftarrow & \mathbf{y}_1^\top & \rightarrow \\ \leftarrow & \mathbf{y}_2^\top & \rightarrow \\ \leftarrow & \vdots & \rightarrow \\ \leftarrow & \mathbf{y}_I^\top & \rightarrow \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1R} \\ a_{21} & a_{22} & \dots & a_{2R} \\ \vdots & & & \vdots \\ a_{I1} & a_{I2} & \dots & a_{IR} \end{bmatrix} \begin{bmatrix} \leftarrow & \mathbf{b}_1{}^\top & \rightarrow \\ \leftarrow & \mathbf{b}_2{}^\top & \rightarrow \\ & \vdots & \\ \leftarrow & \mathbf{b}_R{}^\top & \rightarrow \end{bmatrix}$$

Tucker-1 Decomposition

- A generalization of rank-R matrix factorization

- $\mathbf{Y} = \mathbf{AB}$
- $Y_{ijk} = \sum_{r=1}^{R} A_{ir} B_{rjk}$
- $\mathbf{Y}_i = \sum_{r=1}^{R} A_{ir} \mathbf{B}_r$



1-slices $Y[i, :, :]$ rows $A[i, :]$ 1-slices $B[r, :, :]$

Properties of the model

Should be *low rank*

- Need more mixtures $\mathbf{y}_i$ than sources $\mathbf{b}_r$ $(I > R)$
- Otherwise, there's too many solutions for a_r and $\mathbf{b}_r$
- For example,

$$\mathbf{y} = a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \cdots + a_R\mathbf{b}_R$$

- has the unhelpful solution $a_1 = 1$ and $\mathbf{b}_1 = \mathbf{y}$, where the rest of of the coefficients $a_r = 0$ for $r = 2, \ldots, R$

Properties of the model

Should be *scaled*

- Need to scale coefficients a_{ir} or sources $\mathbf{b}_r$ (or both)
- Otherwise, decomposition is not unique and unbounded
- For example,

$$\mathbf{Y} = \mathbf{AB} = (c\mathbf{A})\left(\tfrac{1}{c}\mathbf{B}\right) = (\mathbf{AC})\left(\mathbf{C}^{-1}\mathbf{B}\right)$$

- for all positive $c > 0$, or $R \times R$ invertible matrix $\mathbf{C}$
- Common scales include normalizing columns/rows of $\mathbf{A}$ or slices of $\mathbf{B}$

Solution Method

First, we need our densities y_i

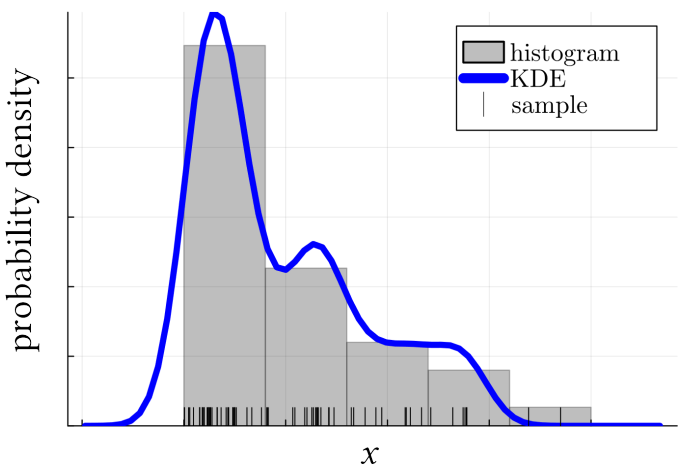
- Often cannot measure our mixture densities y_i directly
- Typical problem:
  - given i.i.d. samples $s_i^n \sim \mathcal{Y}_i$ for $n = 1, \ldots, N_i$
  - each distribution $\mathcal{Y}_i$ is a convex mixture of the (the same) source distributions $\mathcal{B}_R$: $\mathcal{Y}_i = a_{i1}\mathcal{B}_1 + \cdots + a_{iR}\mathcal{B}_R$
  - estimate mixing coefficients a_{ir} and probability density functions b_r for the distributions $\mathcal{B}_R$
- In other words, "$\mathbb{P}\left(s \sim \mathcal{B}_r | s \sim \mathcal{Y}_i\right) = a_{ir}$"

Density Estimation from samples

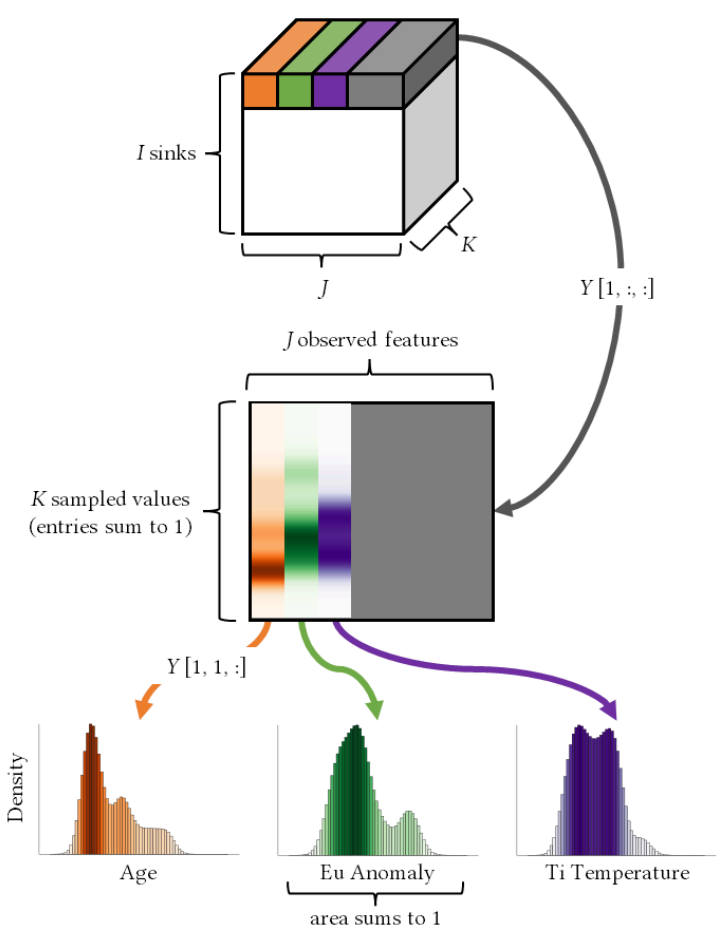
- *if* we knew what the sources look like…
- …parametrize sources and optimize parameters with
  - expectation maximization (Dempster et al. 1977)
  - method of moments (Pearson 1936)
  - e.g, Gaussian mixture models (Lindsay and Basak 1993)
- *otherwise* we need a nonparametric estimation

Density Estimation from samples

- Kernel density estimation (Rosenblatt 1956; Parzen 1962)
- $y(x) = \frac{1}{N_i} \sum_{n=1}^{N_i} k\left(\frac{x - s^n}{h}\right)$ is the KDE from samples s^n
- k is the *kernel* and h is the *bandwidth*

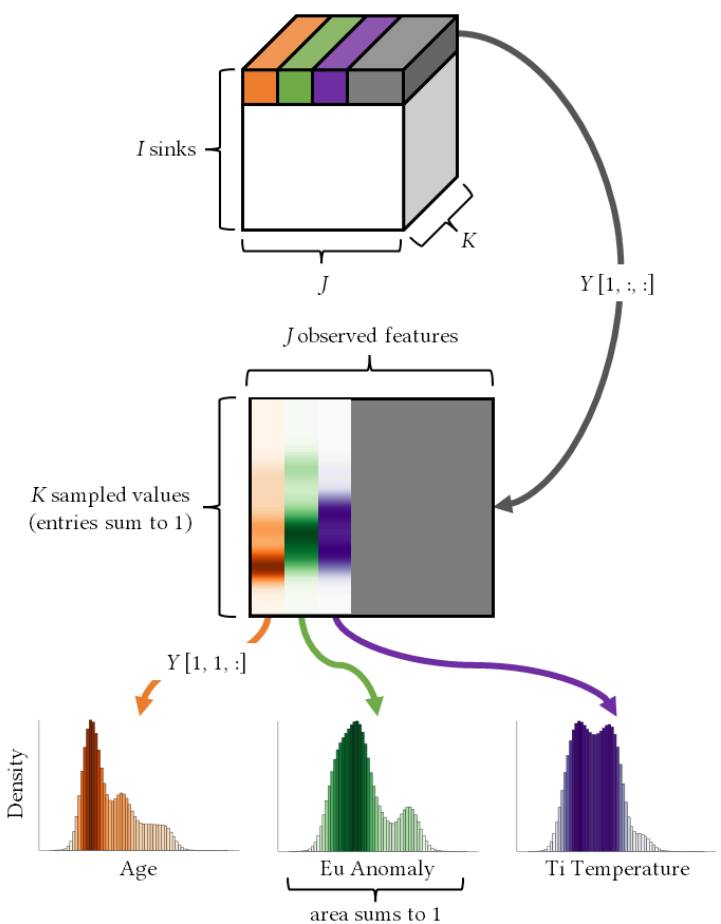


Higher Order KDE & Tensors



- If *independent* features
  - store discretized 1-dim KDEs for each sink i and feature j in $\mathbf{Y}_{ij:}$
- Doing it this way, reduces the size of the data
  - J-dim KDEs[1] (I of them) is more expensive to compute than IJ cheeper 1-dim KDEs
  - IK^J to IJK tensor entries

Higher Order KDE & Tensors



Contributions / advancements

- Perform this source seperation on all features jointly
- Model that scales well to arbitrary number of features J
- Size of problem is independent of number of samples collected

Source separation model

$$\min_{\mathbf{A},\mathbf{B}} \left\{ \ell(\mathbf{A},\mathbf{B}) := \tfrac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{B}\|_F^2 \,\middle|\, \mathbf{A} \in \Delta_{\mathbf{A}},\ \mathbf{B} \in \Delta_{\mathbf{B}} \right\}$$

$\|\cdot\|_F^2$ squared Frobenius norm (sum-of-squares)

$\Delta_{\mathbf{A}}$ non-negative entries & rows sum to 1

$\Delta_{\mathbf{B}}$ non-negative entries & fibres $\mathbf{B}_{ij:}$ sum to 1

- Nonnegativity implies it's NP hard[1] 🙁
- Simplex implies a bounded feasible set 😊
- Not convex, but *block*-convex and smooth

Block Coordinate Descent

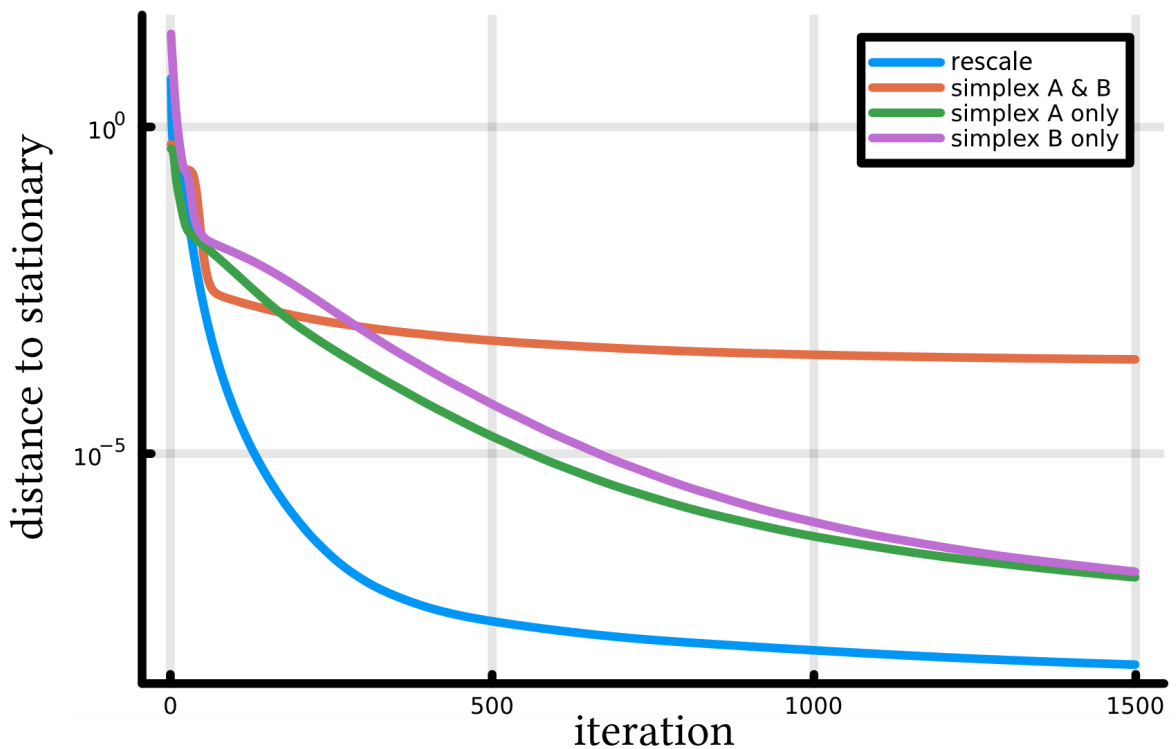
- Alternatly update $\mathbf{A}$ and $\mathbf{B}$ via projected gradient descent[1]
  - $\mathbf{A}^{t+1} = P_{\Delta_\mathbf{A}}(\mathbf{A}^t - \frac{1}{L_\mathbf{A}} \nabla_\mathbf{A} \ell(\mathbf{A}^t, \mathbf{B}^t))$
  - $\mathbf{B}^{t+1} = P_{\Delta_\mathbf{B}}(\mathbf{B}^t - \frac{1}{L_\mathbf{B}} \nabla_\mathbf{B} \ell(\mathbf{A}^{t+1}, \mathbf{B}^t))$
- Convergence to nash equilibria $(\mathbf{A}^*, \mathbf{B}^*)$
  - *block-wise* min: $\ell(\mathbf{A}^*, \mathbf{B}^*) \leq \min\left(\ell(\mathbf{A}^*, \mathbf{B}), \ell(\mathbf{A}, \mathbf{B}^*)\right)$
- Also stationary: $0 \in \partial(\ell + \delta_{\Delta_\mathbf{A} \times \Delta_\mathbf{B}})(\mathbf{A}^*, \mathbf{B}^*)$

Rescaling trick

- Relax constraints to
  - $\mathbf{A} \geq 0, \mathbf{B} \geq 0$
  - $\frac{1}{J} \sum_{jk} B_{rjk} = 1$ for all r (vs $\sum_{k} B_{rjk} = 1$ for all r, j)
- Updates now look like:
  - $\mathbf{A}^{t+1/2} = (\mathbf{A}^t - \frac{1}{L_{\mathbf{A}}} \nabla_{\mathbf{A}} \ell(\mathbf{A}^t, \mathbf{B}^t))_+$
  - $\mathbf{B}^{t+1/2} = (\mathbf{B}^t - \frac{1}{L_{\mathbf{B}}} \nabla_{\mathbf{B}} \ell(\mathbf{A}^{t+1/2}, \mathbf{B}^t))_+$
  - $\mathbf{B}^{t+1} = \mathbf{C}^{-1}\mathbf{B}^{t+1/2}$ and $\mathbf{A}^{t+1} = \mathbf{A}^{t+1/2}\mathbf{C}$
  - where $C_{rr} = \frac{1}{J} \sum_{jk} B_{rjk}$

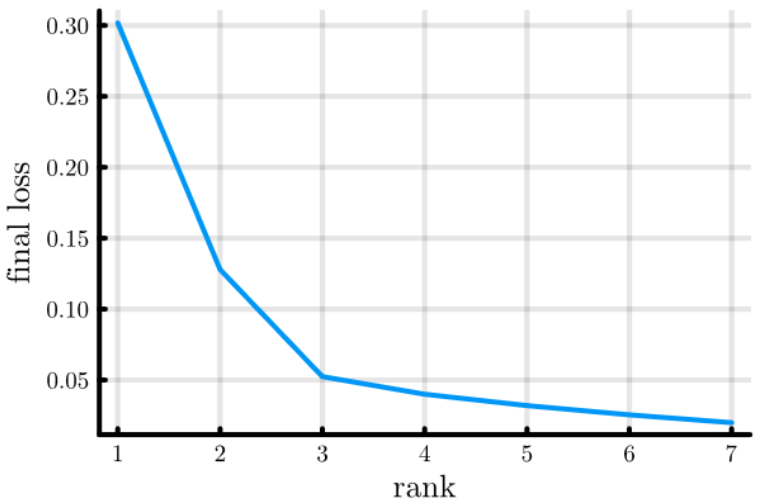
Rescaling vs simplex projection

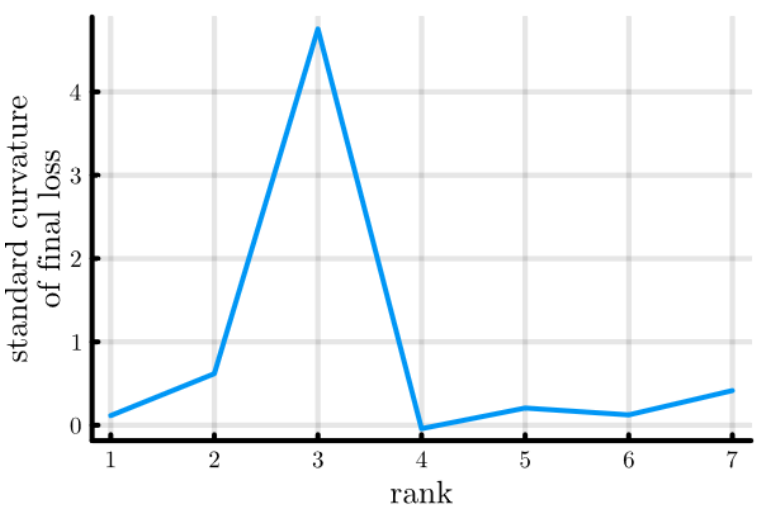
- Compare stationary condition dist $(0, \partial(\ell + \delta_{\geq 0})(\mathbf{A}, \mathbf{B}))$ every iteration for different constraint methods



Estimating the rank

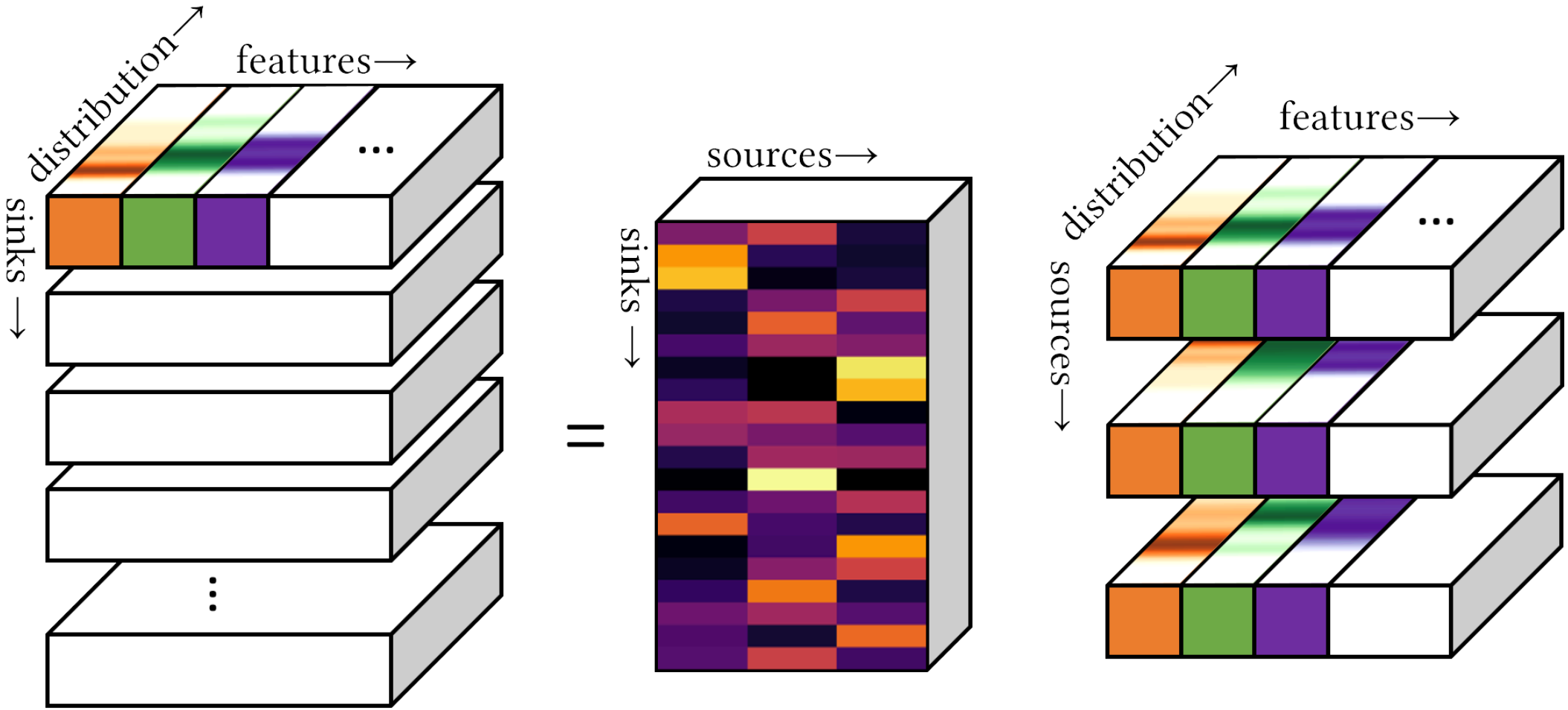
- Try many ranks $R = 1, \ldots, R_{\max}$
- Compare the objective as a function of the rank
  $\ell(R) = \|\mathbf{Y} - \mathbf{A}_R\mathbf{B}_R\|_F^2$
- *Occam's Razor*: Trade off between simple model (low rank R) and explanatory power (larger R)
- Select point of maximum curvature
  $\arg\max_R \ell''(R)/(1+\ell'(R)^2)^{1.5}$



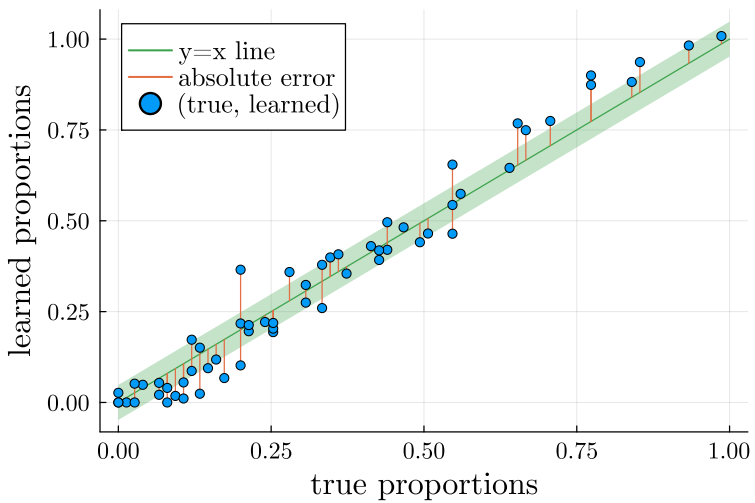


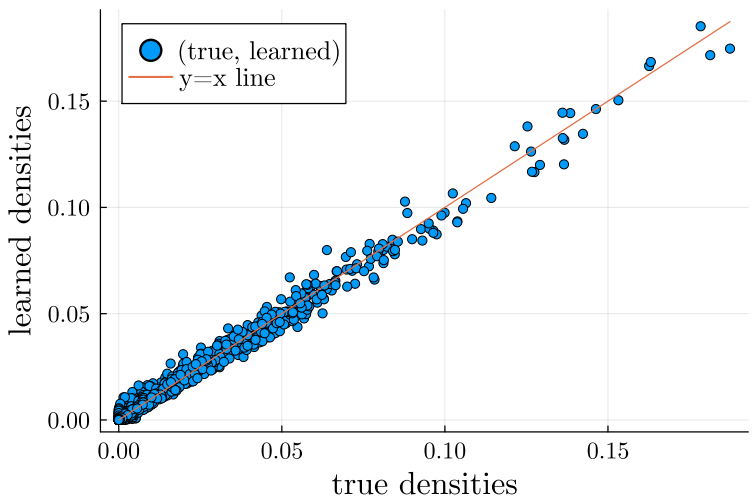
Results

Application 1: Geology

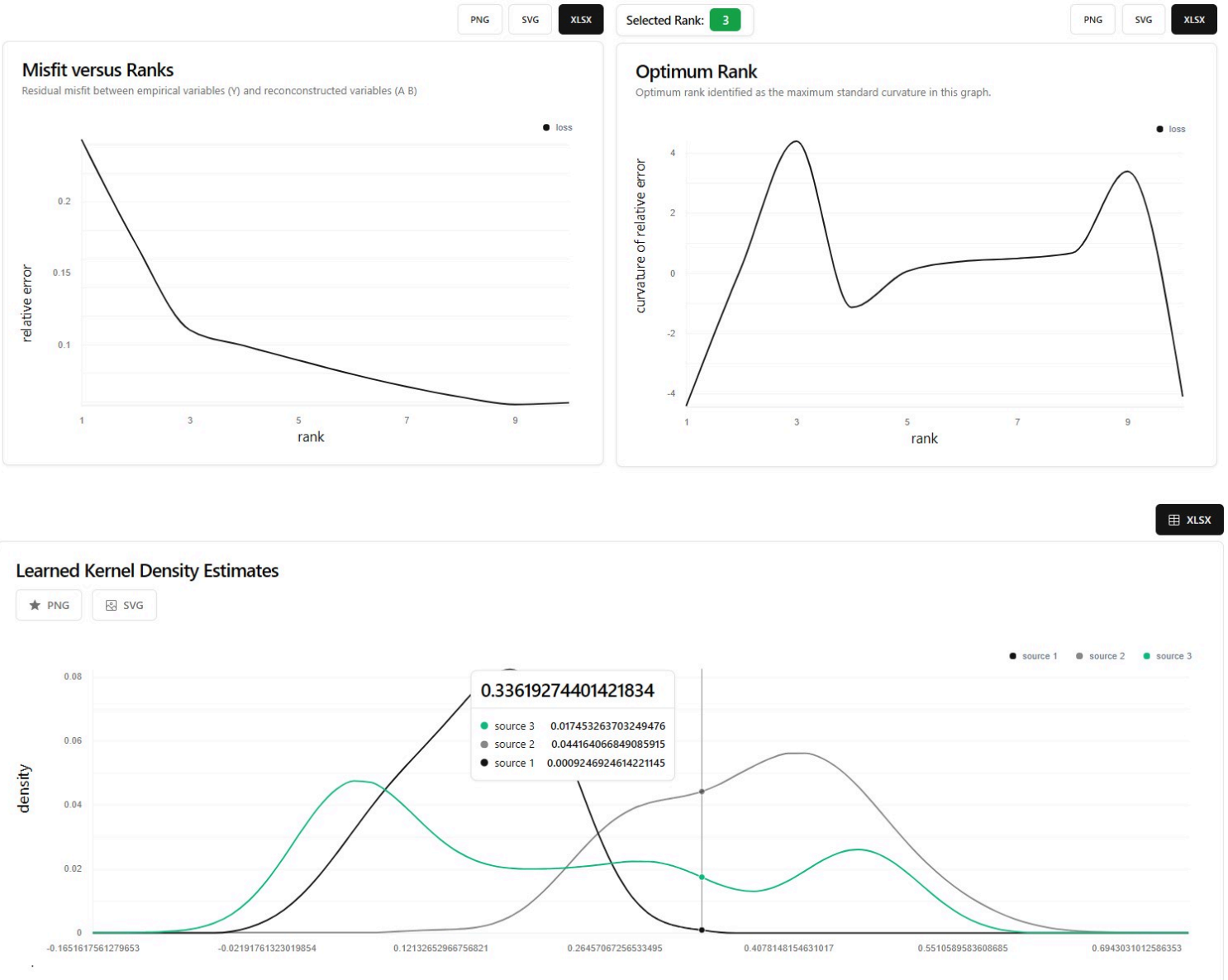


Application 1: Geology





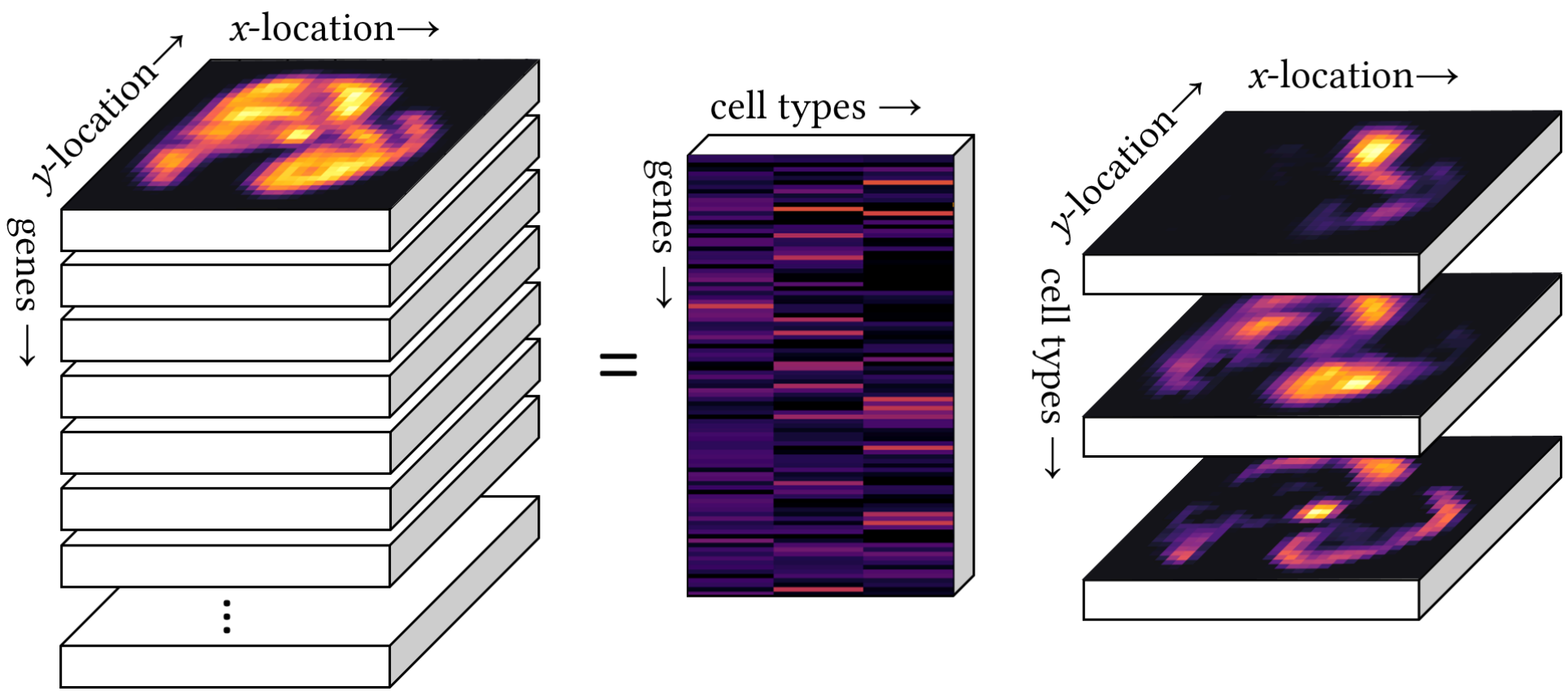
http://dzgrainalyzer.eoas.ubc.ca



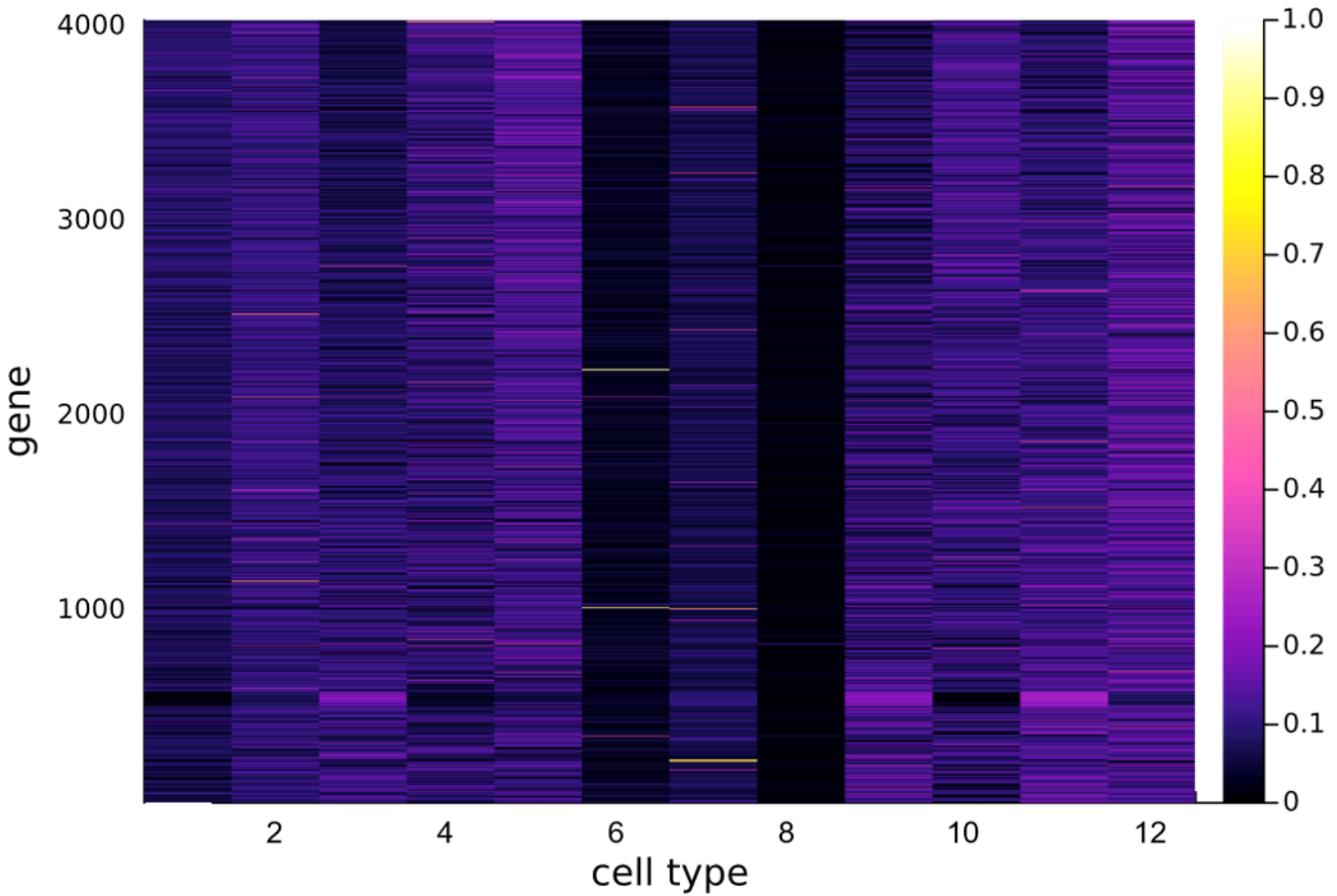
Density Separation with Tensor Factorization

Application 2: Transcriptomics

- constraint modification: horizontal slices are normalized
  $\sum_{j,k} B_{r,j,k} = 1 \; \forall r$

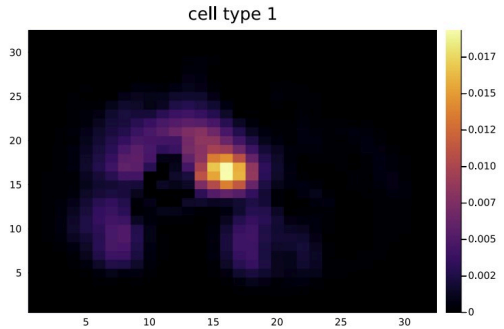


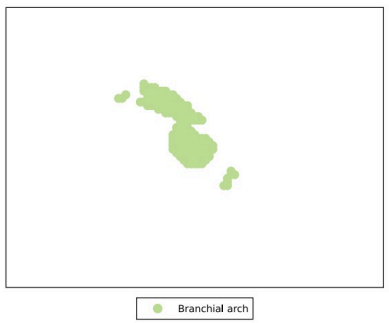
Application 2: Transcriptomics

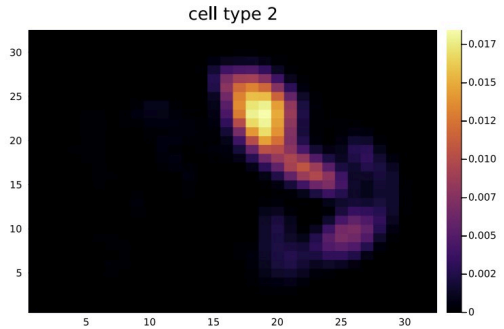


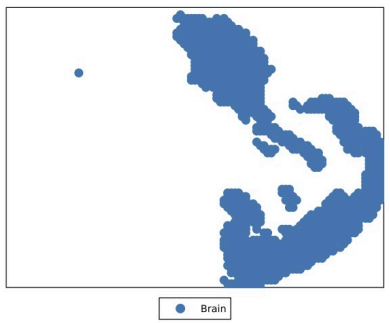
Density Separation with Tensor Factorization

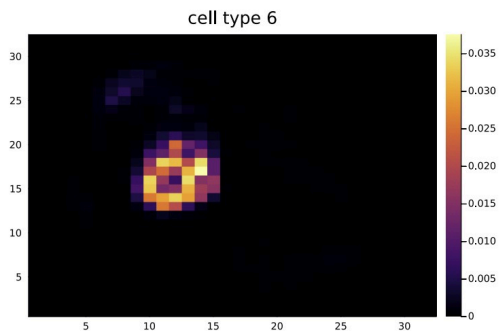
Application 2: Transcriptomics

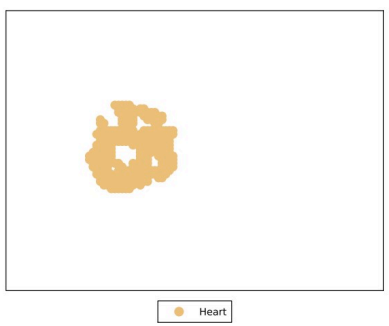


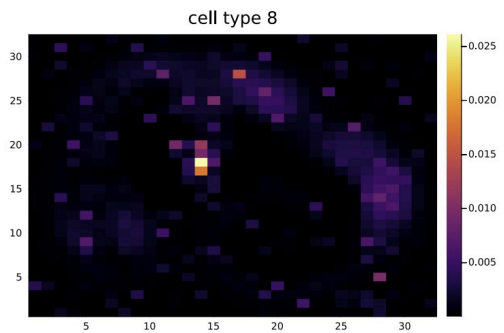


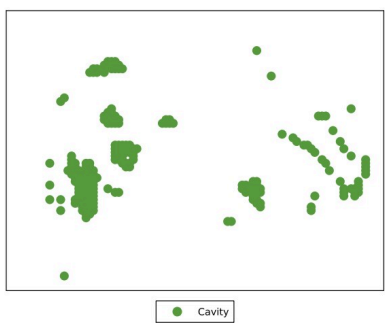






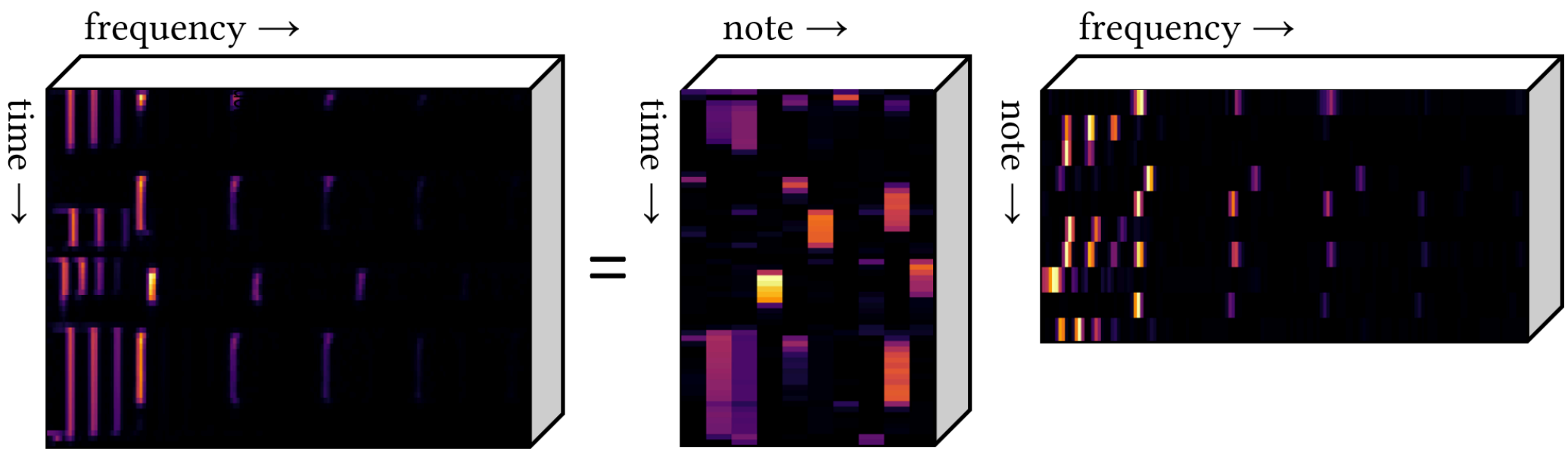






Application 3: Music Decomposition

- constraint modification: horizontal slices are max-normalized $\max_j B_{r,j} = 1 \; \forall r$



Summary

- Combine KDE with Tucker-1 factorization into a scalable **nonparametric density decomposition** method
- **Algorithm converges** to block-minimum and stationary point
  - Open-source Julia code on GitHub: MatrixTensorFactor.jl
- **Practical model** applicable to many areas
  - geology, genetics, music, etc.
- Expanding code for **more decompositions**, **faster convergence** along cts. dimensions

Paper & Code

References

Abdi H, Williams LJ (2010) Principal component analysis. WIREs Computational Statistics 2:433–459. https://doi.org/10.1002/wics.101

Benedetto JJ, Frazier MW (1993) Wavelets: Mathematics and Applications, 1st edn. CI Press, Boca Raton

Cao J, Li Z, Li J (2019) Financial time series forecasting model based on CEEMDAN : LSTM. Physica A: Statistical Mechanics and its Applications 519:127–139. https://doi.org/10.1016/j.physa.2018.11.061

Chen SS, Donoho DL, Saunders MA (2001) Atomic Decomposition by Basis Pursuit. SIAM Review 43:129–159. https://doi.org/10.1137/S003614450037906X

Cohen JE, Rothblum UG (1993) Nonnegative ranks, decompositions, and factorization nonnegative matrices. Linear Algebra and its Applications 190:149–168. https://doi.org/10.1016/0024-3795(93)90224-C

Comon P (1994) Independent component analysis, A new concept? Signal Processing 36:287–314. https://doi.org/10.1016/0165-1684(94)90029-9

Daubechies I, Lu J, Wu H-T (2011) Synchrosqueezed wavelet transforms: An empirica mode decomposition-like tool. Applied and Computational Harmonic Analysis 30:243–261. https://doi.org/10.1016/j.acha.2010.08.002

Dempster AP, Laird NM, Rubin DB (1977) Maximum Likelihood from Incomplete Da
Via the EM Algorithm. Journal of the Royal Statistical Society: Series B
(Methodological) 39:1–22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x
Dragomiretskiy K, Zosso D (2014) Variational Mode Decomposition. IEEE Transactio
on Signal Processing 62:531–544. https://doi.org/10.1109/TSP.2013.2288675
Fan Z, Jeong H, Joshi B, Friedlander MP (2022) Polar Deconvolution of Mixed Signal
IEEE Transactions on Signal Processing 70:2713–2727.
https://doi.org/10.1109/TSP.2022.3178191
Gilles J (2013) Empirical Wavelet Transform. IEEE Transactions on Signal Processing
61:3999–4010. https://doi.org/10.1109/TSP.2013.2265222
Gillis N (2020) Nonnegative Matrix Factorization. Society for Industrial; Applied
Mathematics, Philadelphia, PA
Graham N, Richardson N, Friedlander MP, Saylor J (2024) Tracing Sedimentary Origi
Multivariate Geochronology via Constrained Tensor Factorization. Preprint
Gröchenig K (2001) Foundations of Time-Frequency Analysis. Birkhäuser, Boston, M
Hashemi A, Schaeffer H, Shi R, et al (2023) Generalization bounds for sparse random
feature expansions. Applied and Computational Harmonic Analysis 62:310–330.
https://doi.org/10.1016/j.acha.2022.08.003
Huang NE, Shen Z, Long SR, et al (1998) The empirical mode decomposition and the
Hilbert spectrum for nonlinear and non-stationary time series analysis. Proceeding

the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences 454:903–995. https://doi.org/10.1098/rspa.1998.0193

Hyvarinen A (1999) Fast and robust fixed-point algorithms for independent component analysis. IEEE Transactions on Neural Networks 10:626–634. https://doi.org/10.1109/72.761722

Karamatlı E, Cemgil AT, Kırbız S (2019) Audio Source Separation Using Variational Autoencoders and Weak Class Supervision. IEEE Signal Processing Letters 26:13 1353. https://doi.org/10.1109/LSP.2019.2929440

Kolda TG, Bader BW (2009) Tensor Decompositions and Applications. SIAM Rev 51:455–500. https://doi.org/10.1137/07070111X

Lindsay BG, Basak P (1993) Multivariate Normal Mixtures: A Fast Consistent Method Moments. Journal of the American Statistical Association 88:468–476. https://doi.org/10.2307/2290326

O'Brien TA, Kashinath K, Cavanaugh NR, et al (2016) A fast and objective multidimensional kernel density estimation method: fastKDE. Computational Statistics & Data Analysis 101:148–160. https://doi.org/10.1016/j.csda.2016.02.01

Parzen E (1962) On Estimation of a Probability Density Function and Mode. The Anna of Mathematical Statistics 33:1065–1076. https://doi.org/10.1214/aoms/11777044

Pearson K (1901) LIII. On lines and planes of closest fit to systems of points in space. London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. https://doi.org/10.1080/14786440109462720

Pearson K (1936) Method of Moments and Method of Maximum Likelihood. Biometri
28:34–59. https://doi.org/10.2307/2334123
Rahimi A, Recht B (2008) Uniform approximation of functions with random bases. In:
2008 46th Annual Allerton Conference on Communication, Control, and Computi
pp 555–561
Richardson N, Graham N, Friedlander MP (2024a) MatrixTensorFactor.jl. GitHub
Richardson N, Graham N, Friedlander MP, Saylor J (2024b) SedimentAnalysis.jl. GitH
Richardson N, Schaeffer H, Tran G (2023) SRMD: Sparse Random Mode Decomposit
Communications on Applied Mathematics and Computation.
https://doi.org/10.1007/s42967-023-00273-x
Rosenblatt M (1956) Remarks on Some Nonparametric Estimates of a Density Functio
The Annals of Mathematical Statistics 27:832–837.
https://doi.org/10.1214/aoms/1177728190
Tošić I, Frossard P (2011) Dictionary Learning. IEEE Signal Processing Magazine 28:2
38. https://doi.org/10.1109/MSP.2010.939537
Vavasis SA (2010) On the Complexity of Nonnegative Matrix Factorization. SIAM Jou
on Optimization 20:1364–1377. https://doi.org/10.1137/070709967
Xu Y, Yin W (2013) A Block Coordinate Descent Method for Regularized Multiconvex
Optimization with Applications to Nonnegative Tensor Factorization and Complet
SIAM J Imaging Sci 6:1758–1789. https://doi.org/10.1137/120887795

Additional Slides

What features did we look at?

- age
- Eu anomaly
- Ti-based crystallization temperature
- Th-U ratio
- sum of light rare-earth elements over heavy rare-earth elements (ΣLREE/ΣHREE)
- Dy-Yb ratio
- normalized (Ce/ND)/Y ratio

Convergence Details

- when iterates are bounded iterates
  - there are limit points
  - know this because feasible set is bounded
- when the objective function is KL
  - sequence of iterates converges to a finite limit point

How should we format our data?

Example

- When you sample $s \sim \mathcal{Y}$, do the following
  - Sample r where $r = 1$ with probability a_1, and $r = 2$ with probability $a_2 = 1 - a_1$
  - Draw a sample from distribution $\mathcal{B}_r$
- The distribution of s is the same as $a_1\mathcal{B}_1 + a_1\mathcal{B}_2$
- So the density function y for distribution $\mathcal{Y}$ is equal to $y = a_1 b_1 + a_2 b_2$

Rank robustness