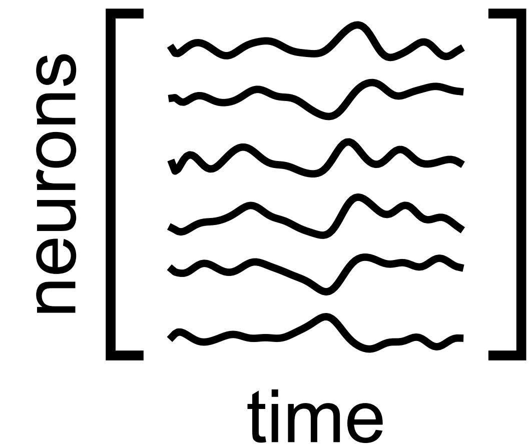
Agenda

https://github.com/ahwillia/mit-tensor

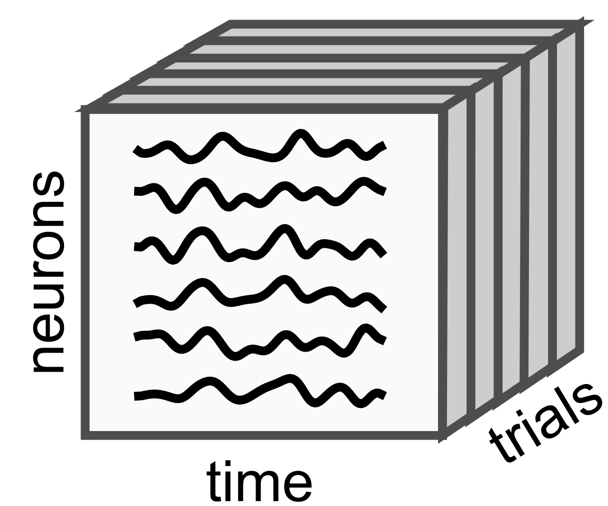
Talk on matrix decomposition **(45 mins)**

Questions & exercise **(15 mins)**



Talk on tensor decomposition **(45 mins)**

Questions & exercise **(15 mins)**



Large-scale data analysis via matrix and tensor decompositions

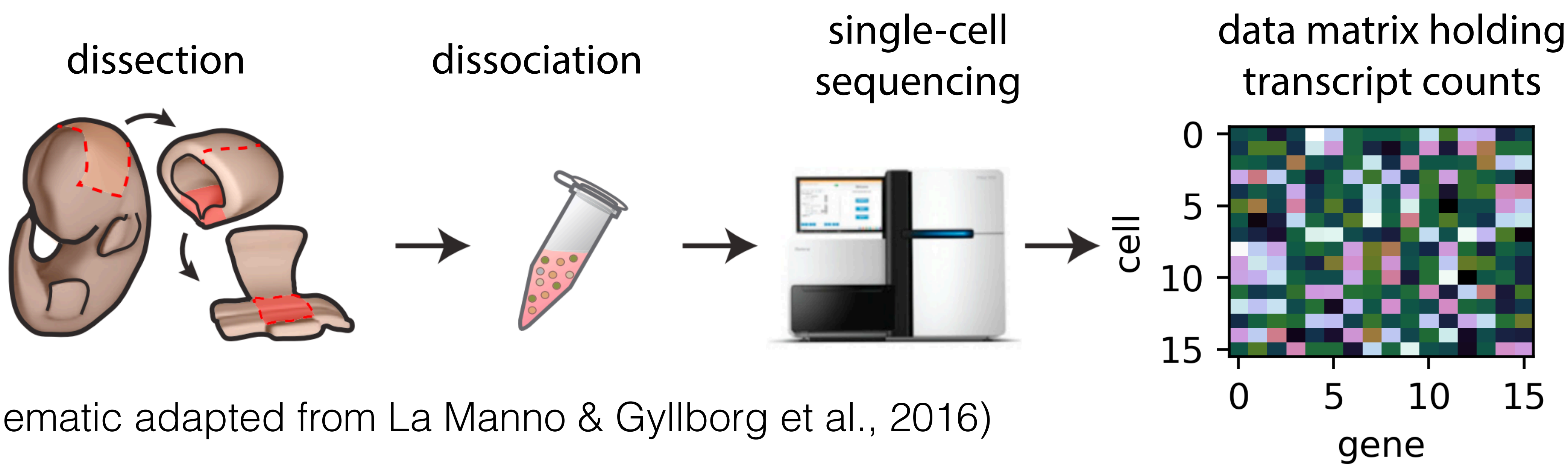
Part 1: Matrix decomposition

Alex Williams

MIT, 09/05/2017

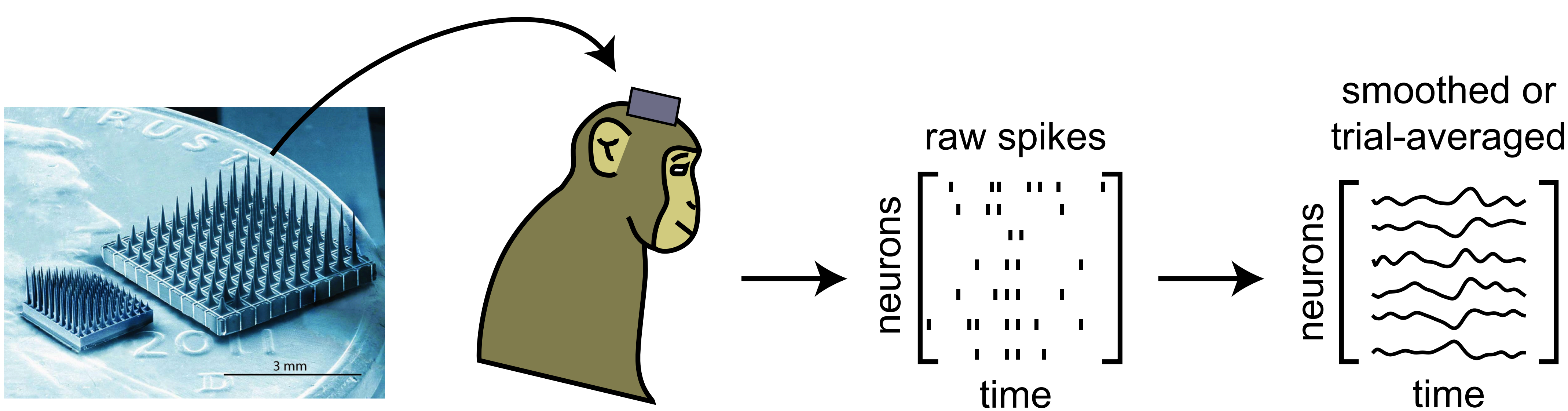
Examples of Matrix-Encoded Data

1. Gene Expression



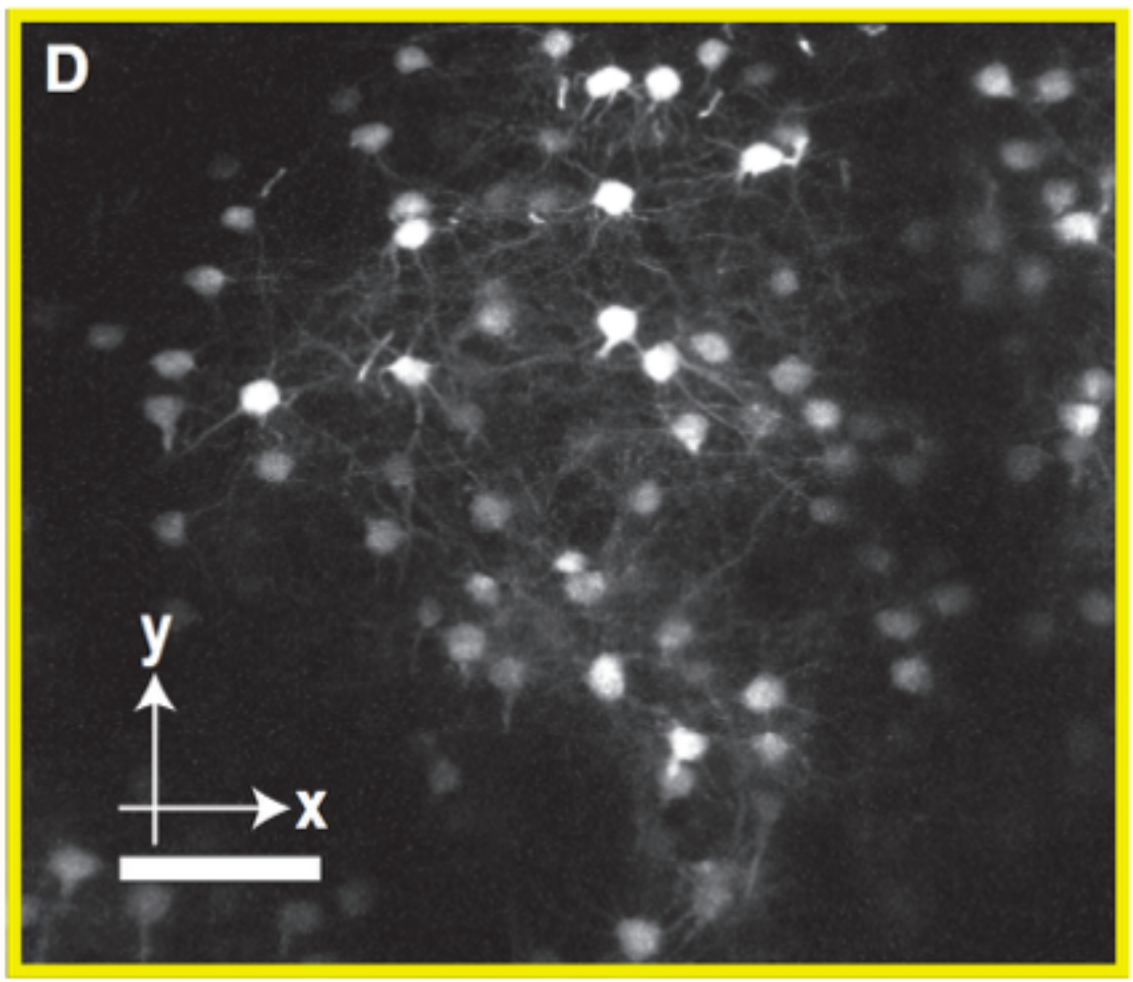
(schematic adapted from La Manno & Gyllborg et al., 2016)

2. Neural Activity



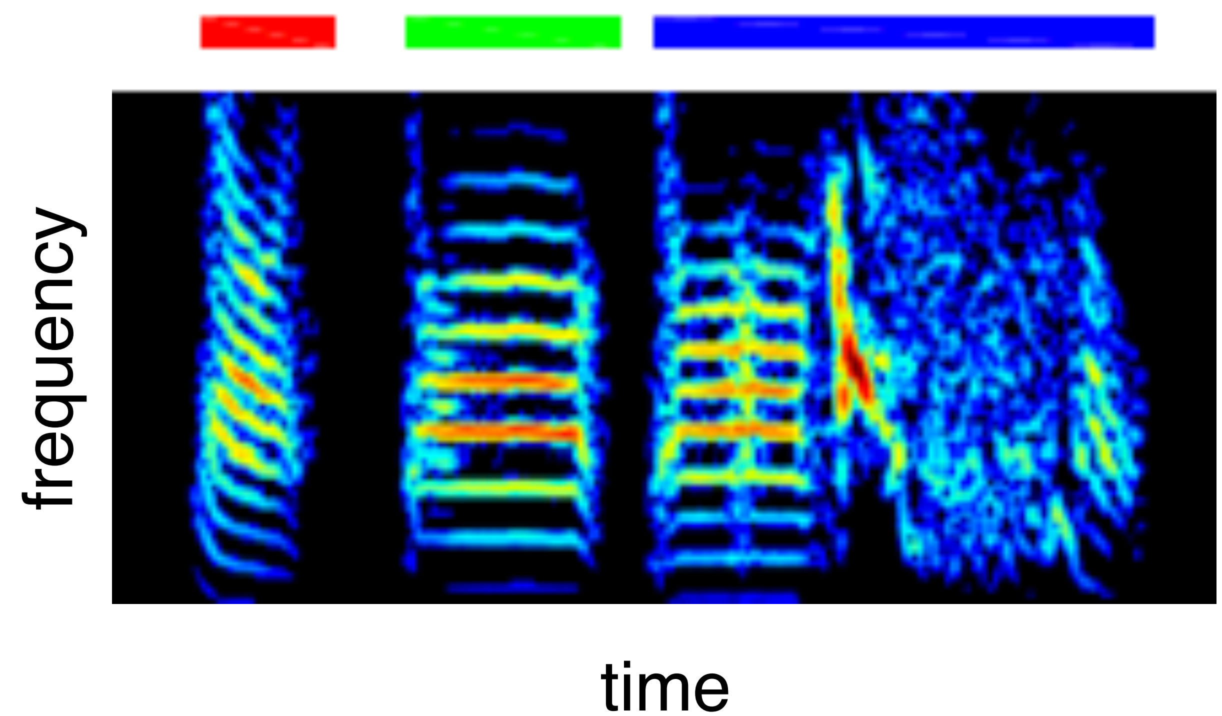
Examples of Matrix-Encoded Data

3. Fluorescence Images



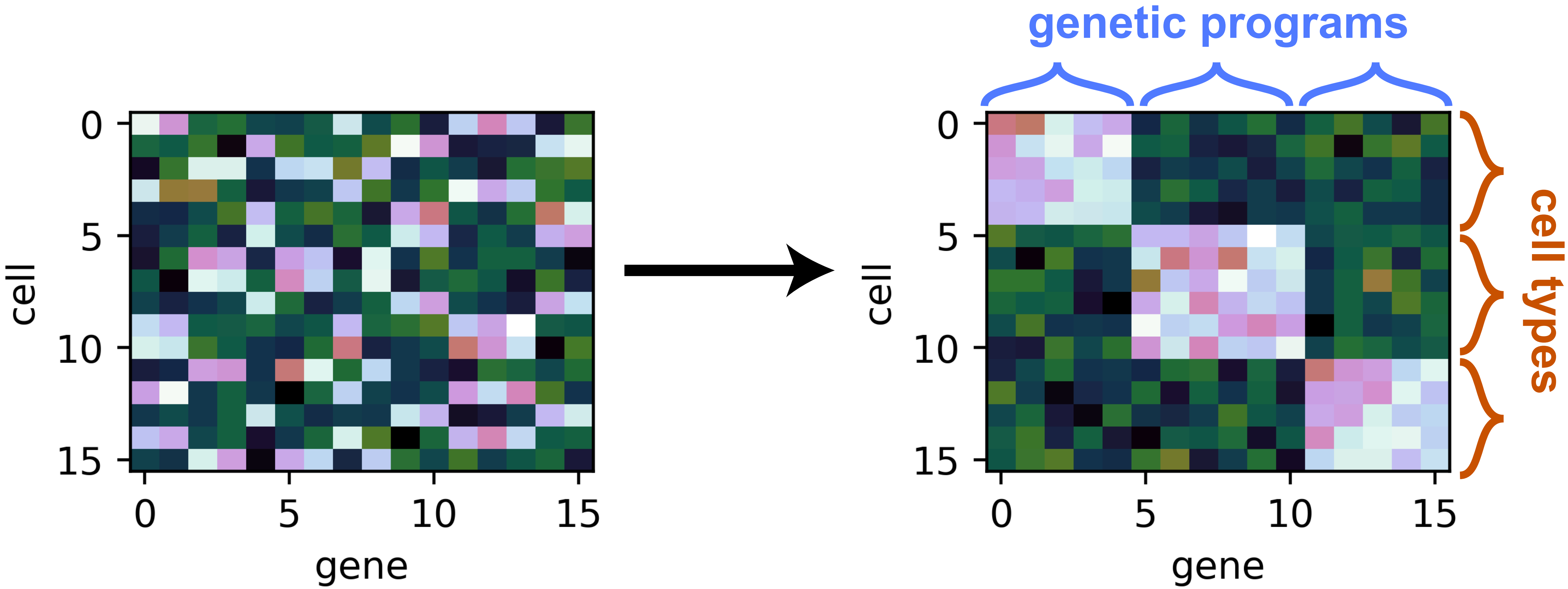
Cortical neurons expressing YFP
(Kim & Zhang et al., 2016)

4. Spectrograms



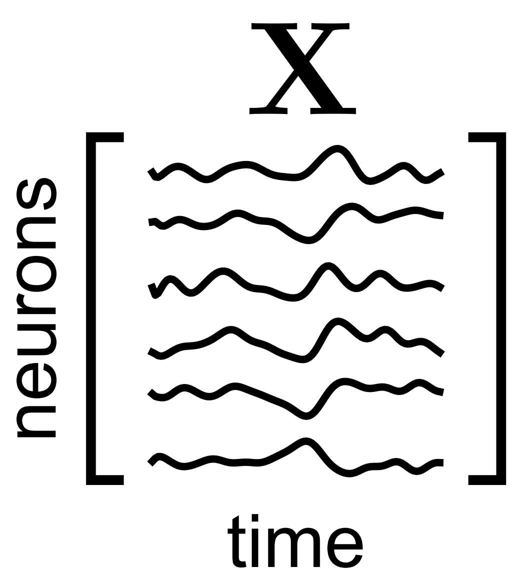
Zebra Finch courtship song
(Provided by Emily Mackevicius)

Goal: extract simple structure from these large-scale datasets



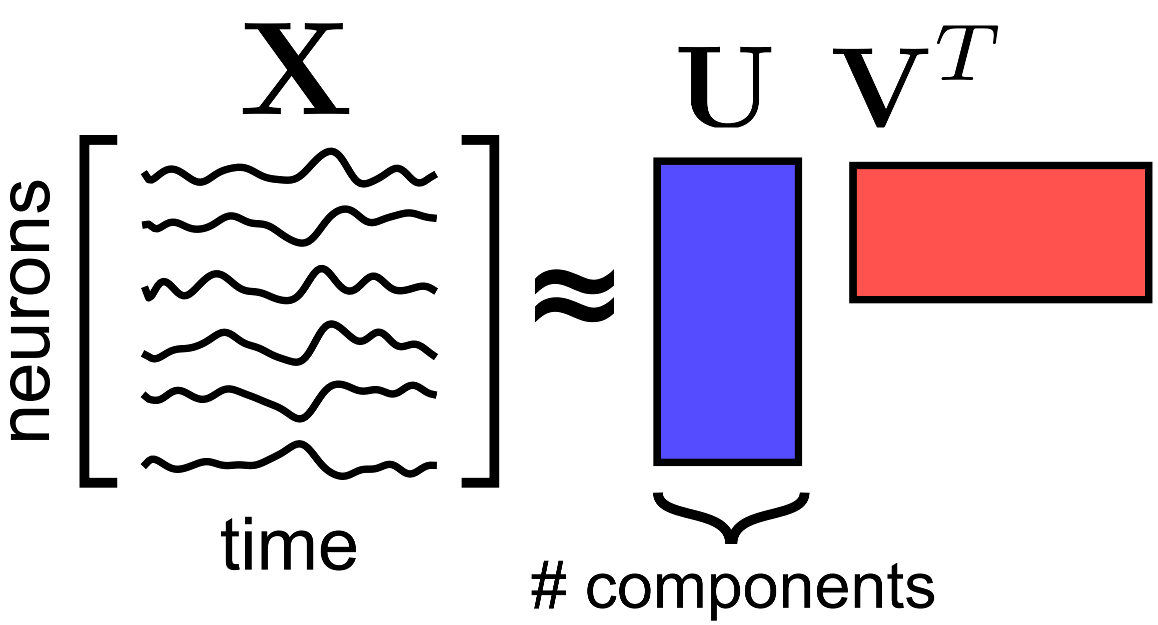
Matrix Decomposition

A simple & general framework for extracting correlations and low-dimensional structure from matrix-coded datasets



Matrix Decomposition

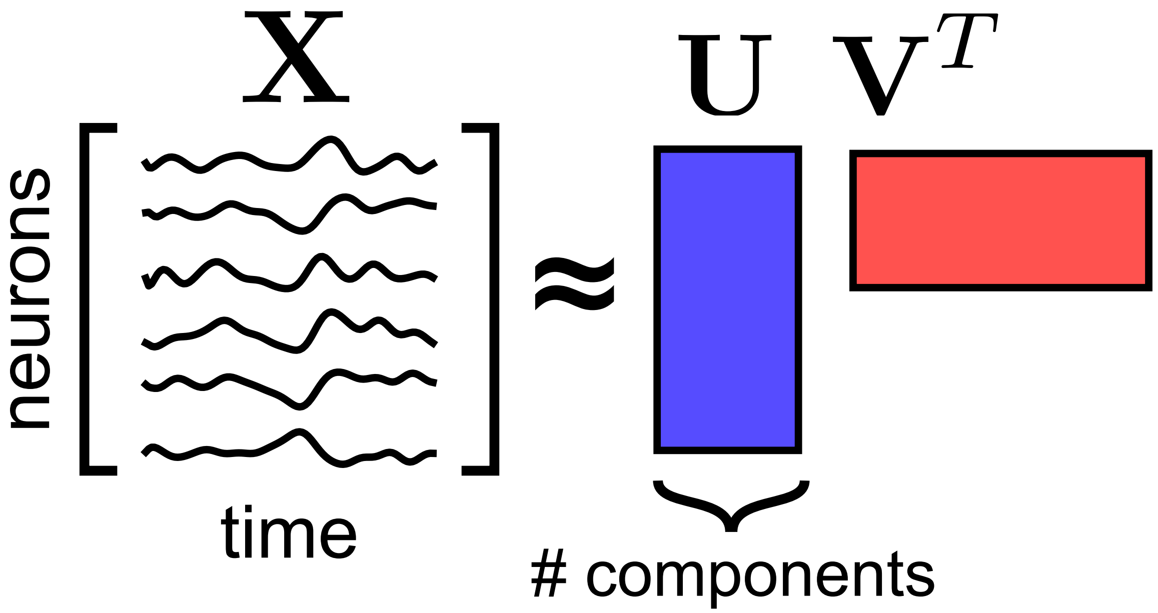
A simple & general framework for extracting correlations and low-dimensional structure from matrix-coded datasets



Matrix Decomposition

A simple & general framework for extracting correlations and low-dimensional structure from matrix-coded datasets

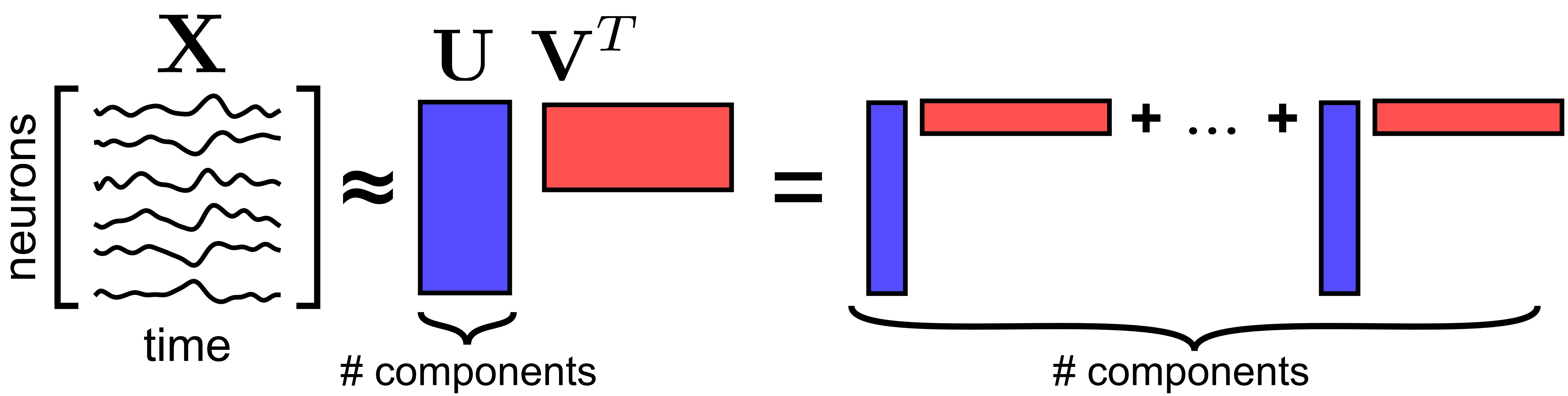
$$x_{ij} \approx \sum_{r=1}^{R} u_i^r v_j^r$$



Matrix Decomposition

A simple & general framework for extracting correlations and low-dimensional structure from matrix-coded datasets

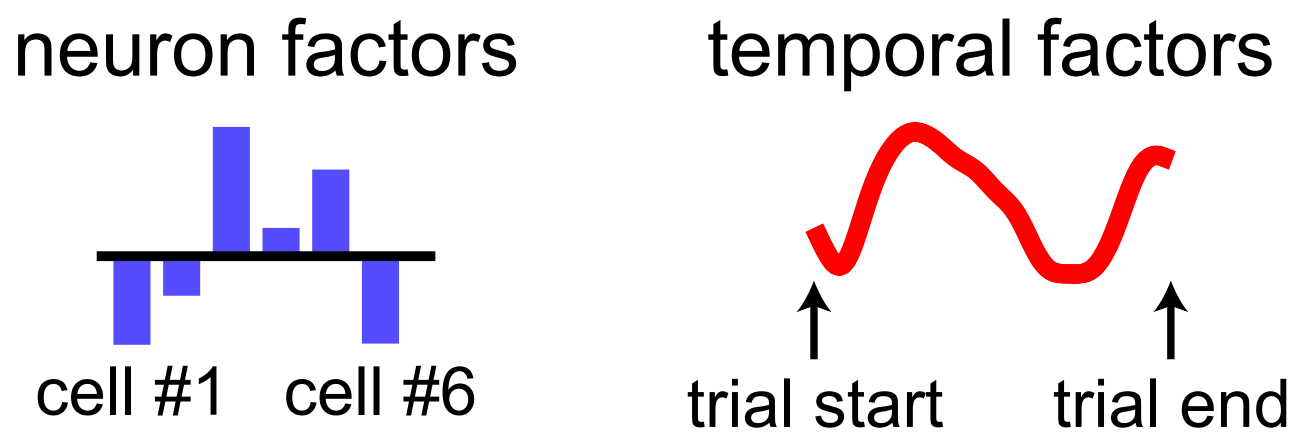
$$x_{ij} \approx \sum_{r=1}^{R} u_i^r v_j^r$$

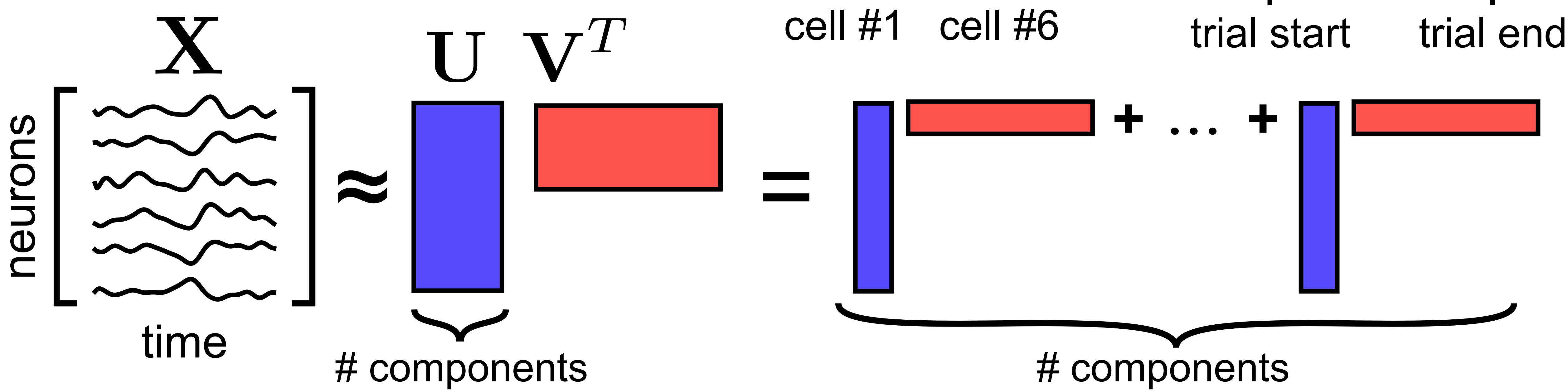


Matrix Decomposition

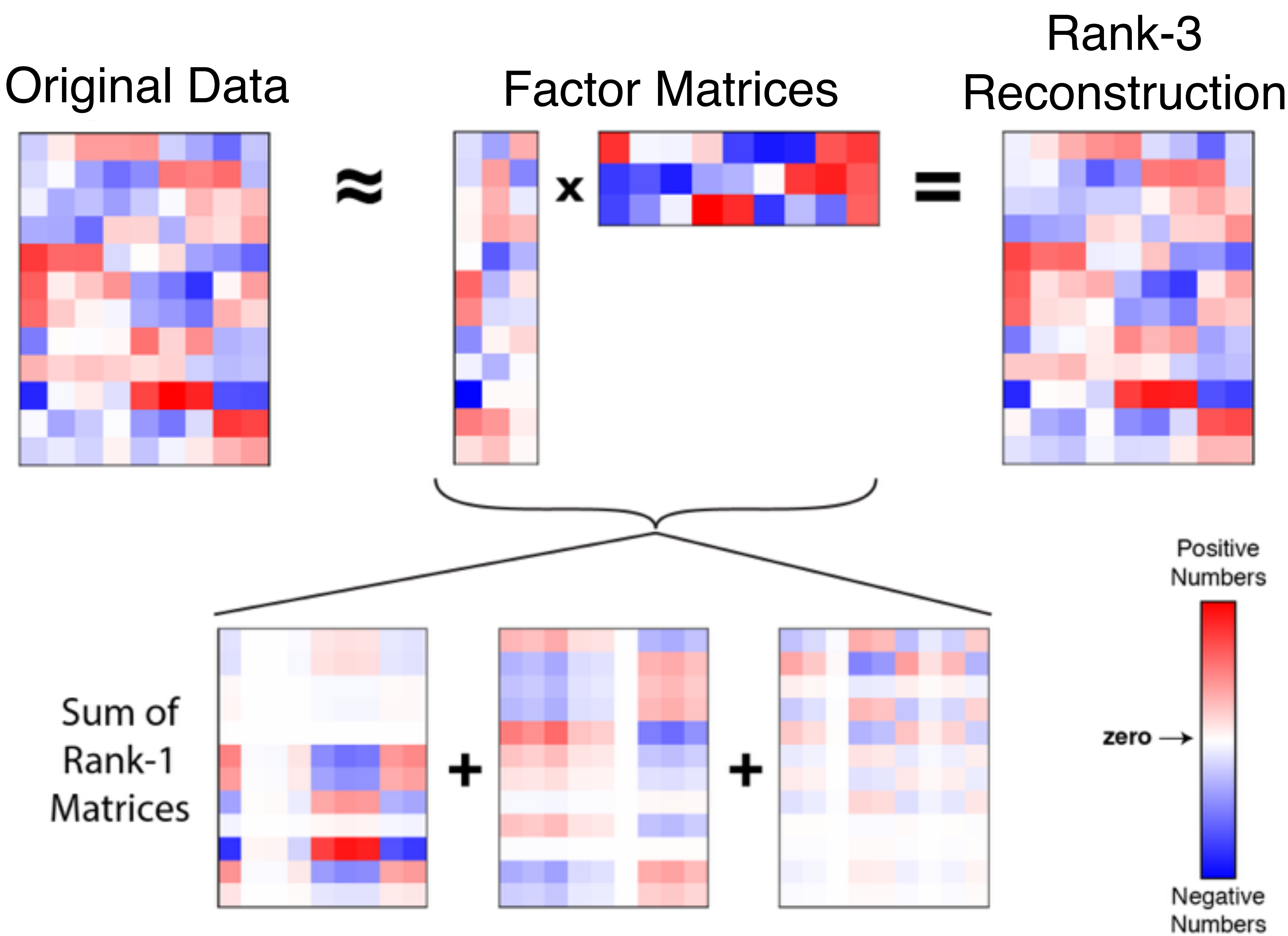
A simple & general framework for extracting correlations and low-dimensional structure from matrix-coded datasets

$$x_{ij} \approx \sum_{r=1}^{R} u_i^r v_j^r$$





Visualization of Matrix Decomposition



Talk Outline

1. Long list of matrix decomposition models

2. Optimization and model fitting

3. Visualization and model assessment

Talk Outline

1. **Long list of matrix decomposition models**

2. Optimization and model fitting

3. Visualization and model assessment

Matrix decomposition model, stated formally

loss *regularization*

$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \lambda_u f_u(\mathbf{U}) + \lambda_v f_v(\mathbf{V})$$

$$\text{subject to} \quad \mathbf{U} \in \Omega_u, \, \mathbf{V} \in \Omega_v$$

constraints

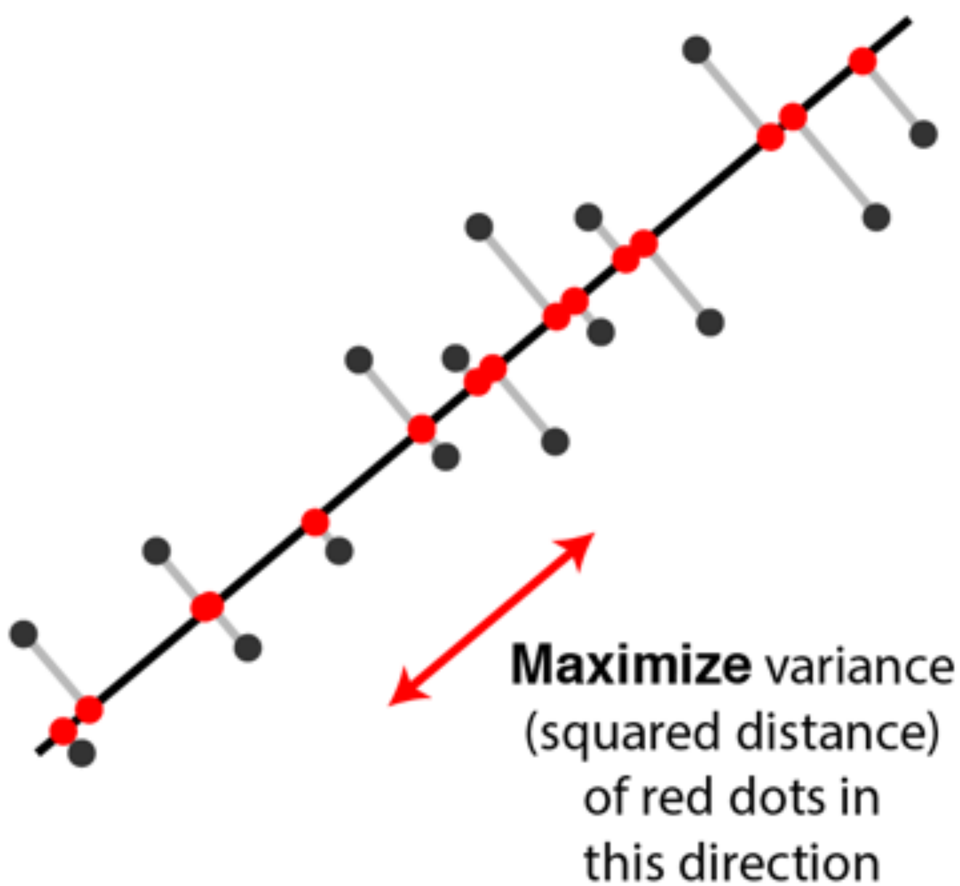
The simplest matrix decomposition is PCA

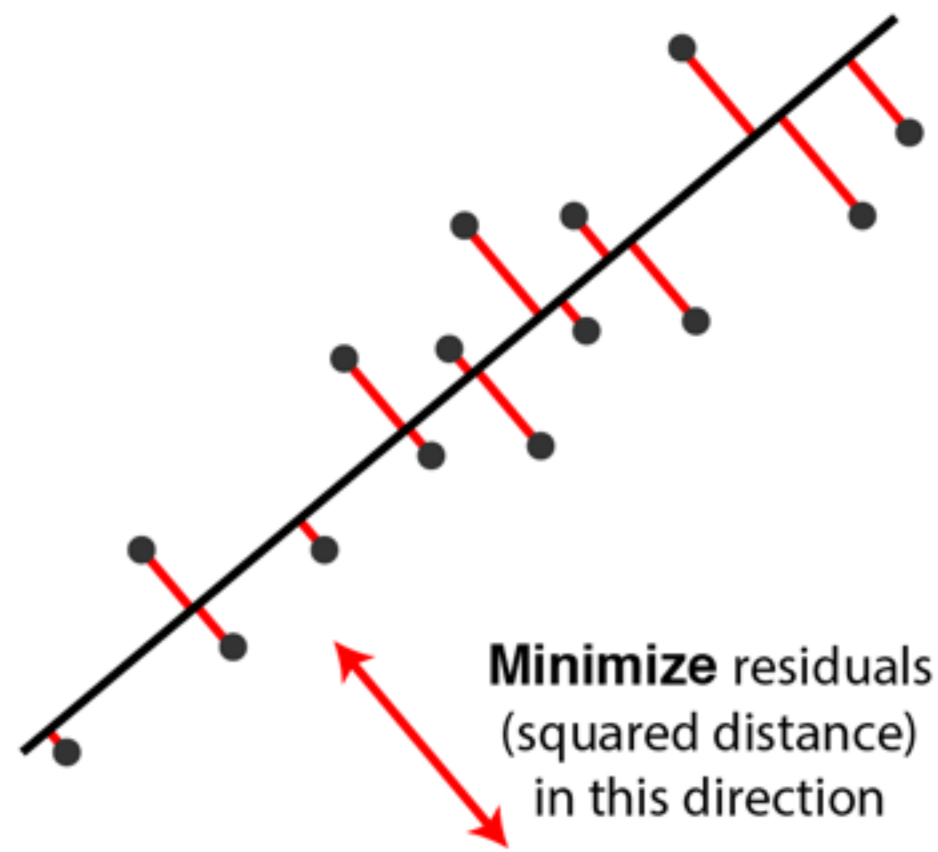
$$\underset{\mathbf{V}}{\text{maximize}} \quad \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2$$

(subject to ***V*** *orthonormal)*

$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2$$

(subject to ***U***, ***V*** *orthogonal)*





There are an infinite # of solutions to PCA

known as "the rotation problem"

$$\widehat{\mathbf{X}} = \mathbf{U}\mathbf{V}^T$$

There are an infinite # of solutions to PCA

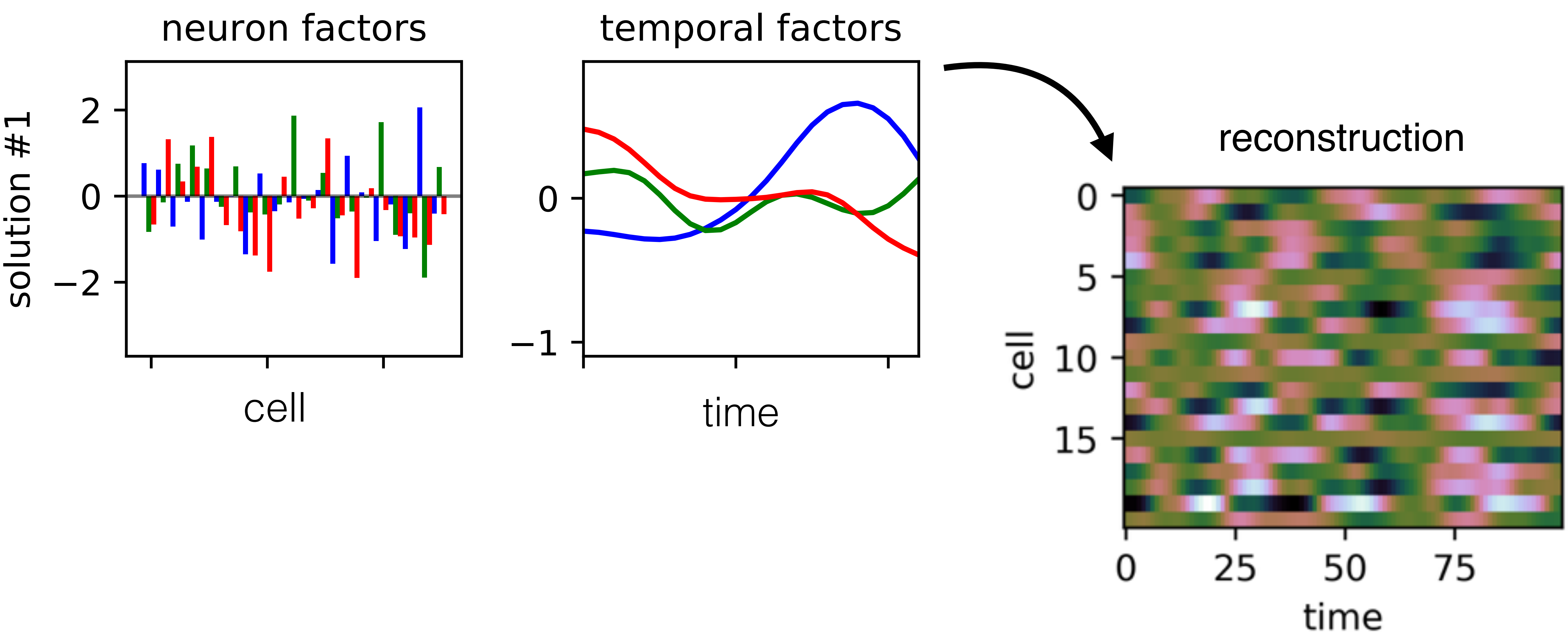
known as "the rotation problem"

$$\widehat{\mathbf{X}} = \mathbf{U}\mathbf{V}^T = \mathbf{U}\mathbf{F}^{-1}\mathbf{F}\mathbf{V}^T = \mathbf{U}'\mathbf{V}'^T$$

There are an infinite # of solutions to PCA

known as “the rotation problem”

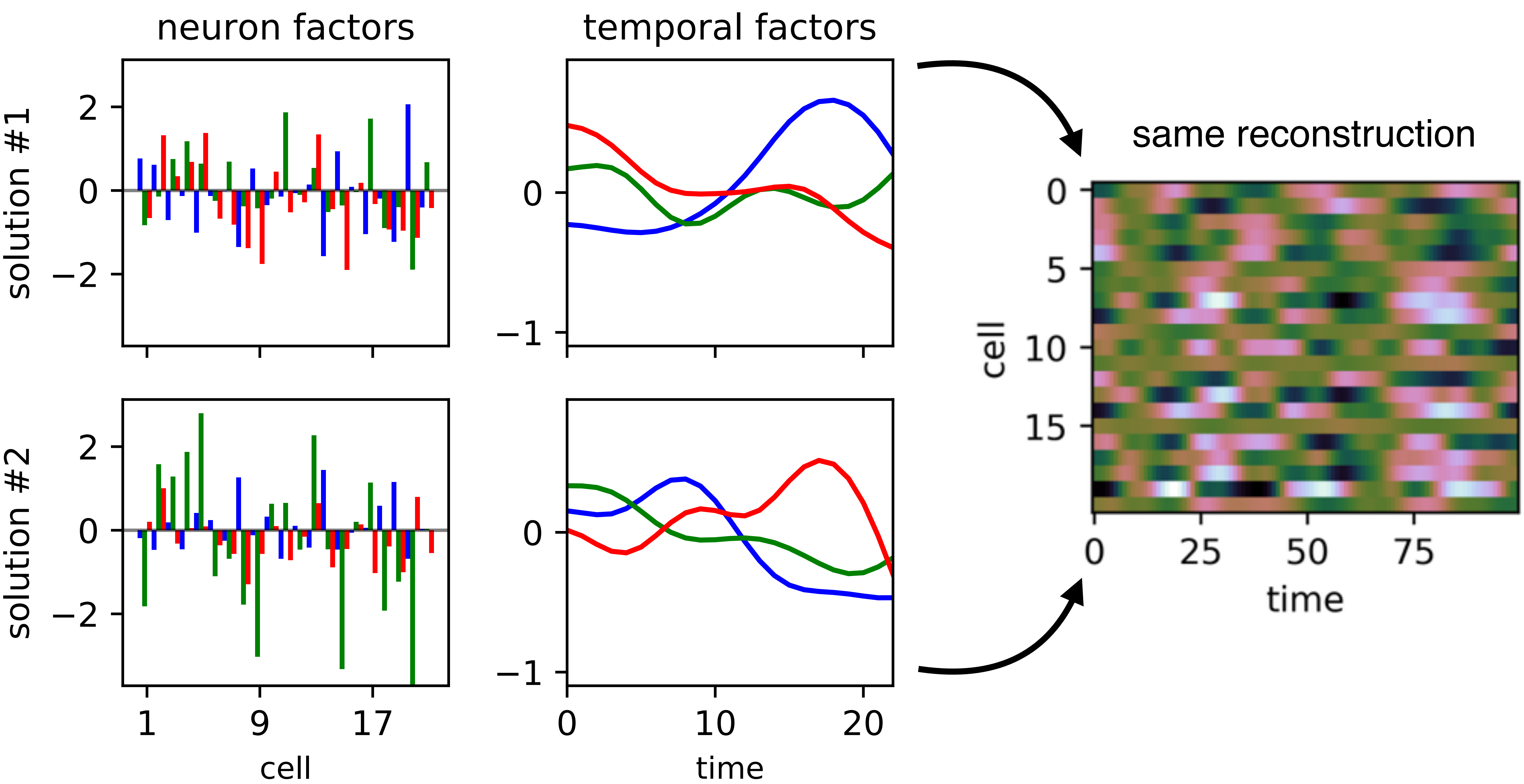
$$\widehat{\mathbf{X}} = \mathbf{U}\mathbf{V}^T = \mathbf{U}\mathbf{F}^{-1}\mathbf{F}\mathbf{V}^T = \mathbf{U}'\mathbf{V}'^T$$



There are an infinite # of solutions to PCA

known as "the rotation problem"

$$\widehat{\mathbf{X}} = \mathbf{U}\mathbf{V}^T = \mathbf{U}\mathbf{F}^{-1}\mathbf{F}\mathbf{V}^T = \mathbf{U}'\mathbf{V}'^T$$

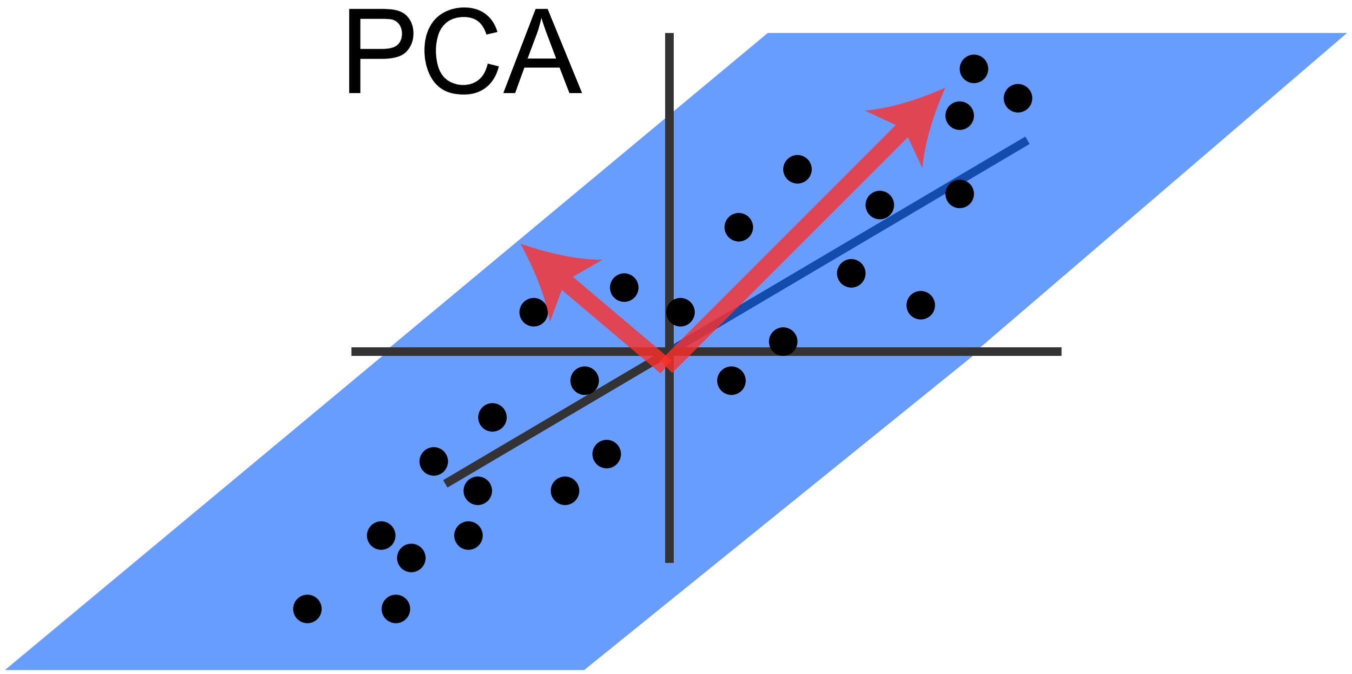


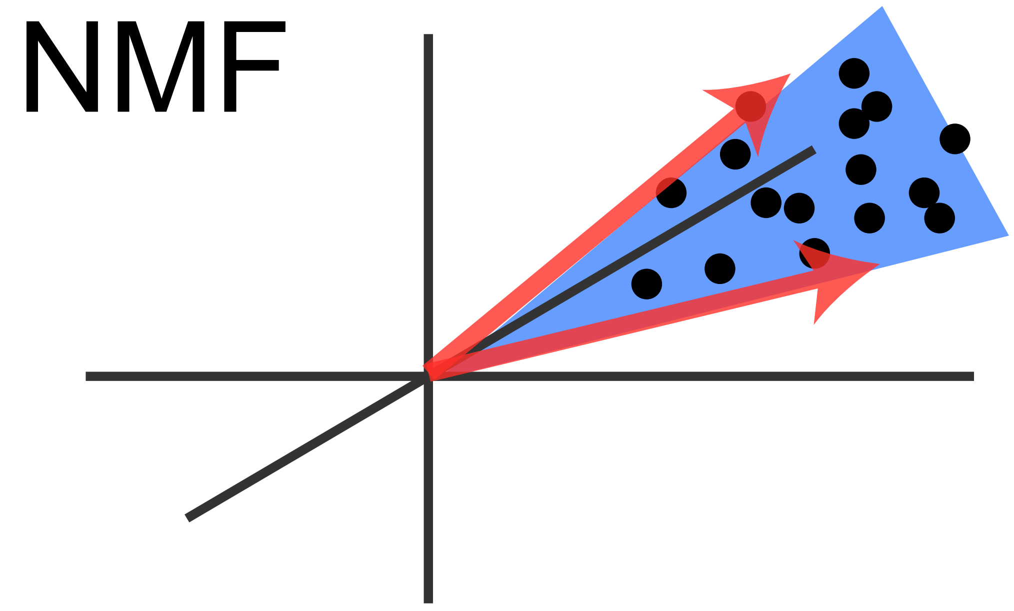
Nonnegative Matrix Factorization (NMF)

$$\begin{array}{ll} \underset{\mathbf{U},\mathbf{V}}{\text{minimize}} & \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 \\ \text{subject to} & \mathbf{U} \geq 0, \, \mathbf{V} \geq 0 \end{array}$$

Nonnegative Matrix Factorization (NMF)

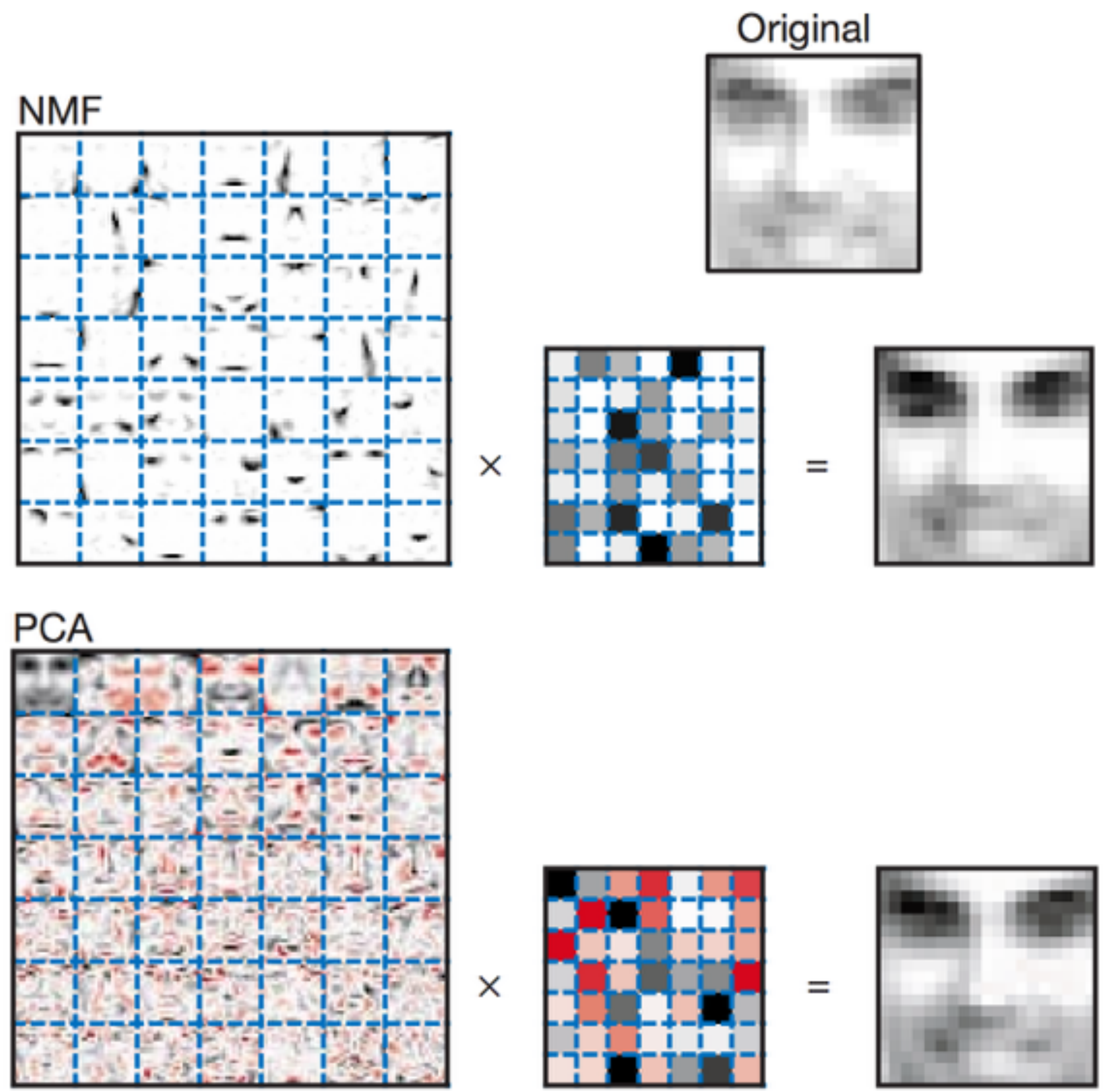
$$\begin{array}{ll} \underset{\mathbf{U},\mathbf{V}}{\text{minimize}} & \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 \\ \text{subject to} & \mathbf{U} \geq 0,\ \mathbf{V} \geq 0 \end{array}$$





Nonnegative Matrix Factorization

$$\begin{array}{ll} \underset{\mathbf{U},\mathbf{V}}{\text{minimize}} & \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 \\ \text{subject to} & \mathbf{U} \geq 0,\ \mathbf{V} \geq 0 \end{array}$$



(Lee & Seung, 1999)

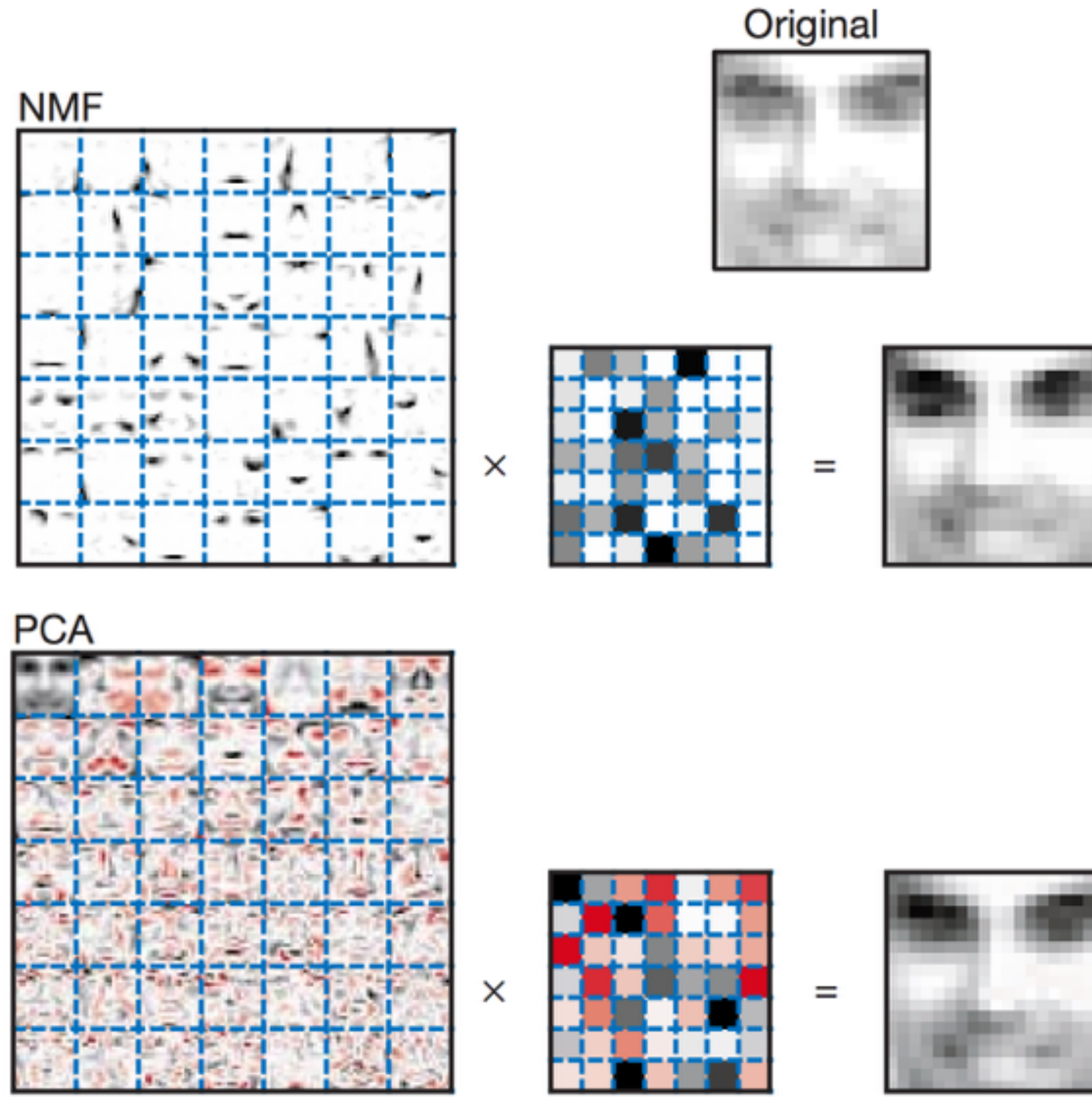
Nonnegative Matrix Factorization

$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2$$
$$\text{subject to} \quad \mathbf{U} \geq 0,\ \mathbf{V} \geq 0$$

NMF advantages:

- sparse factors
- additively combined
- can be “parts-based”
- can be unique (i.e. no rotation problem)

(Stodden & Donoho, 1999)



(Lee & Seung, 1999)

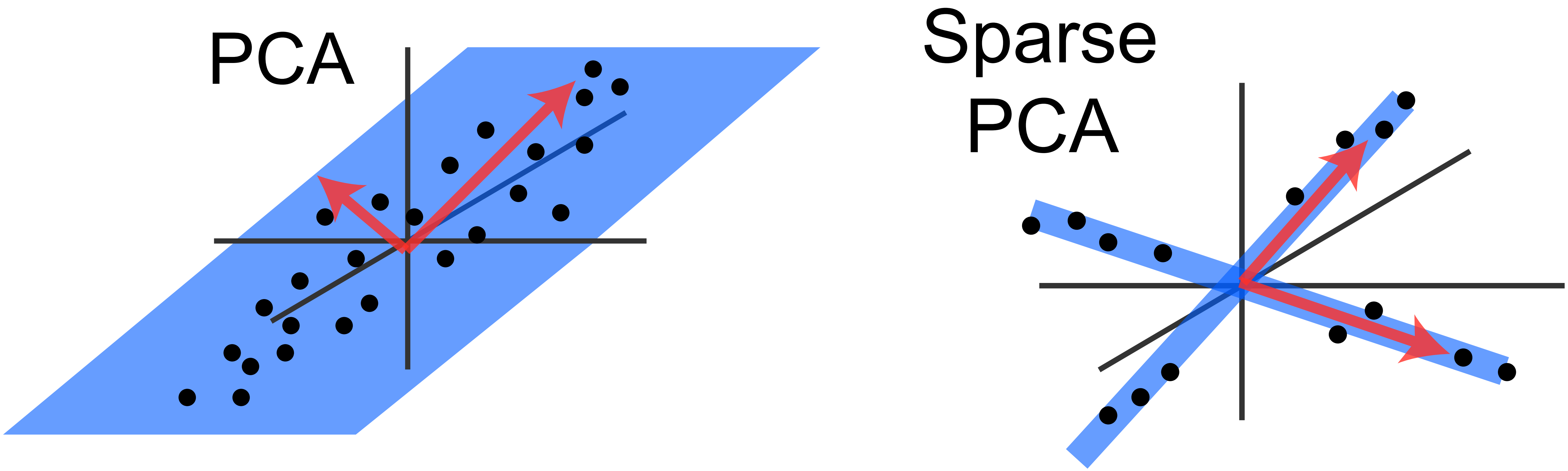
Sparse PCA*

$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \lambda_u \sum_i \|\mathbf{u}_{i:}\|_1 + \lambda_v \sum_j \|\mathbf{v}_{j:}\|_2^2$$

* Several variants of this model with different properties appear in the literature. Originally it was proposed by Zou et al. (2006).

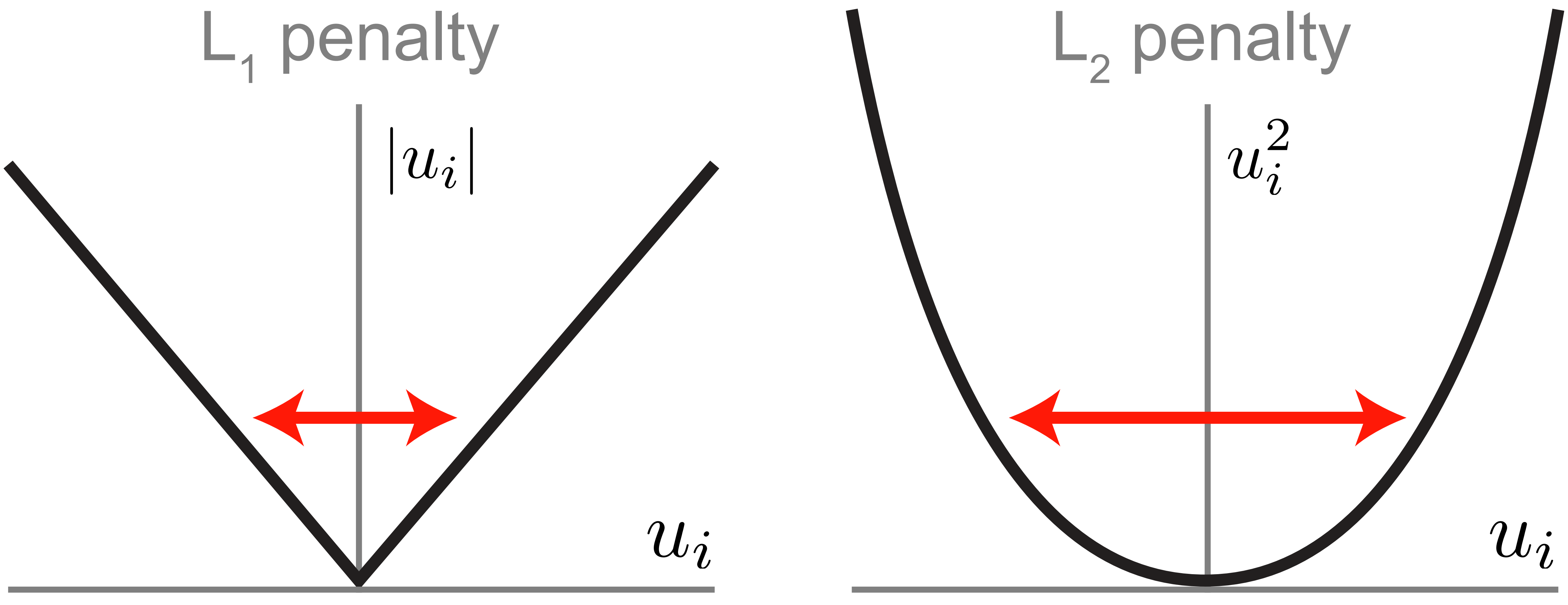
Sparse PCA*

$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \lambda_u \sum_i \|\mathbf{u}_{i:}\|_1 + \lambda_v \sum_j \|\mathbf{v}_{j:}\|_2^2$$



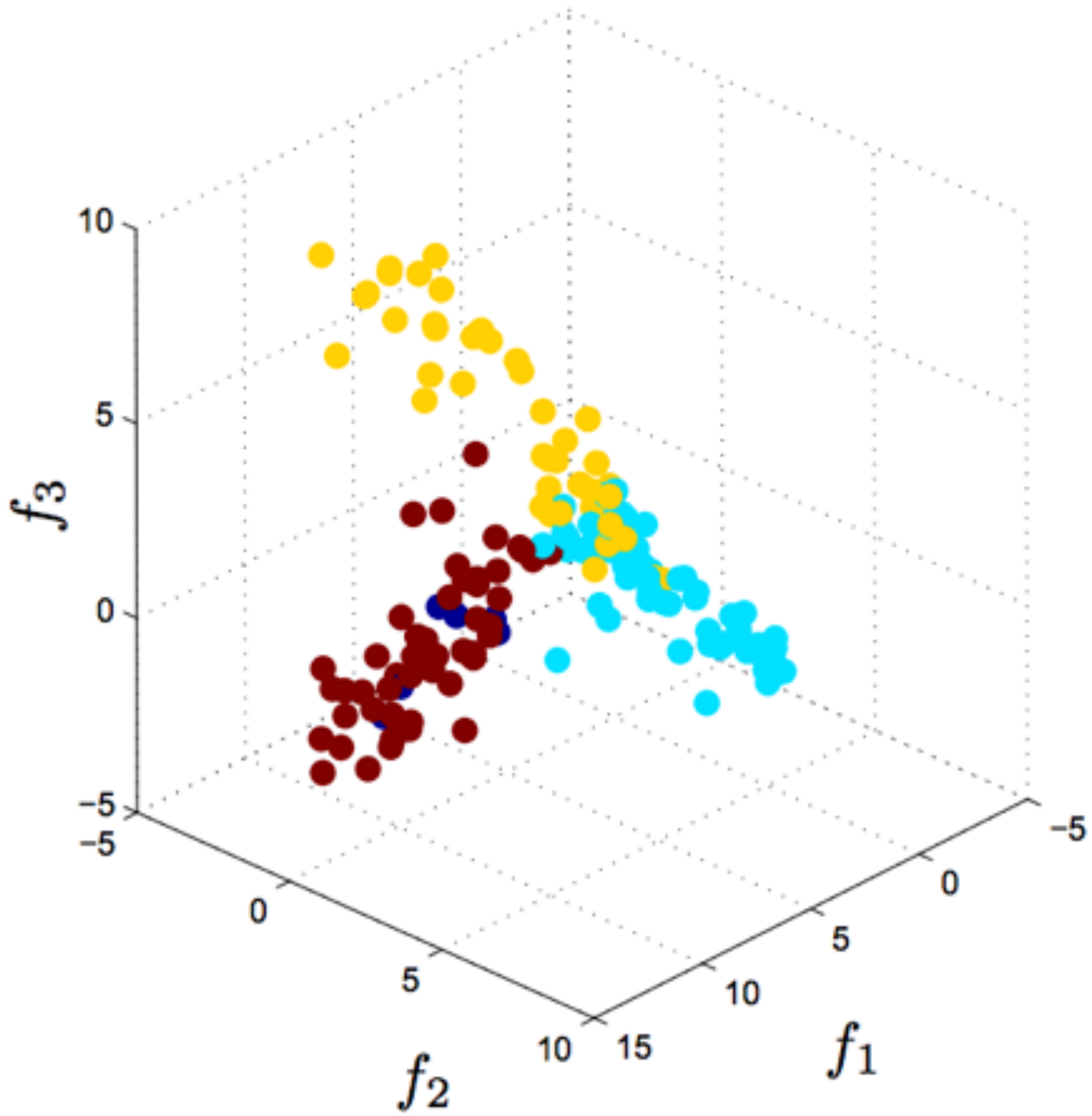
* Several variants of this model with different properties appear in the literature. Originally it was proposed by Zou et al. (2006).

Why L1 penalties result in sparse factors

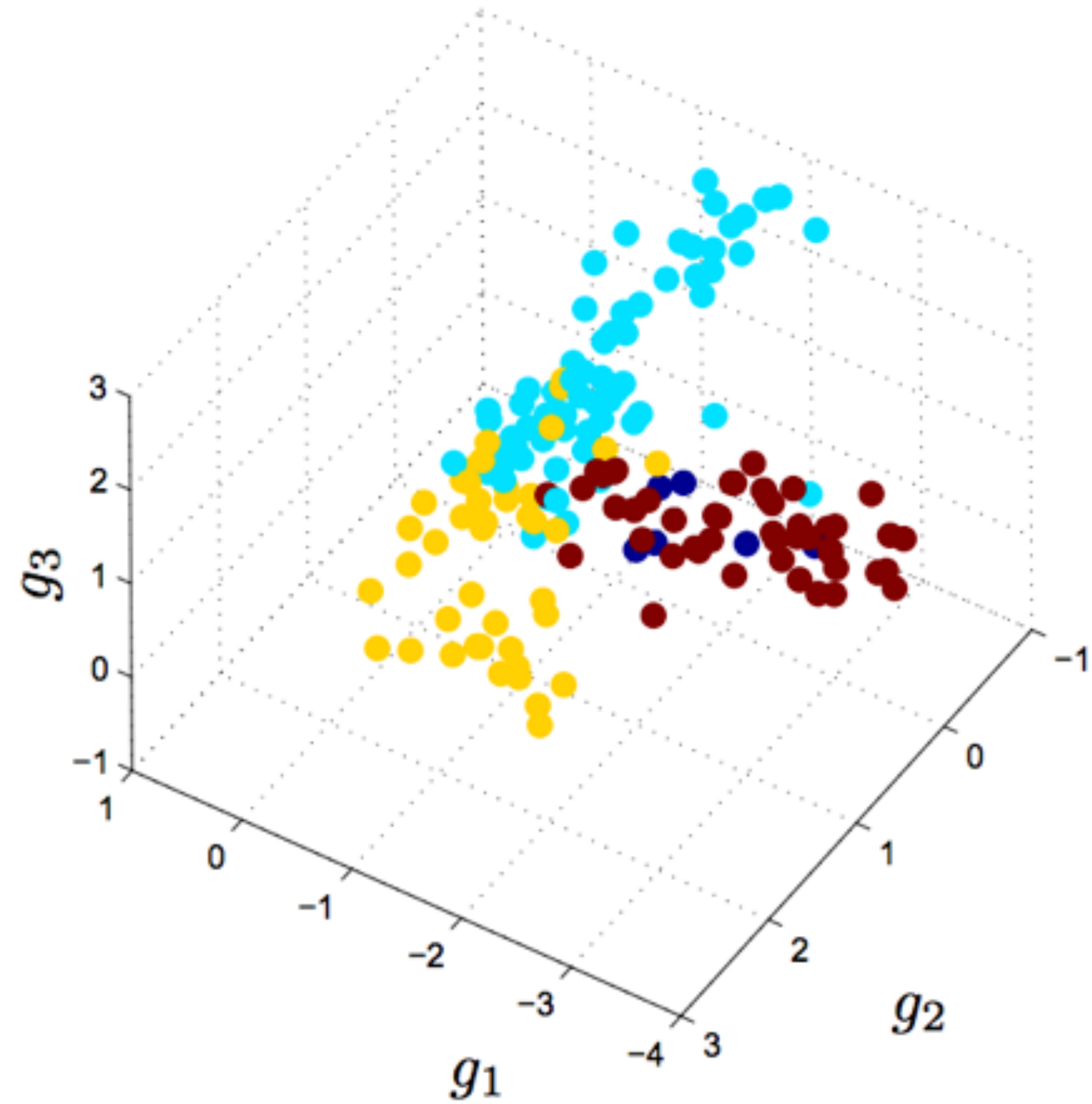


Sparse PCA

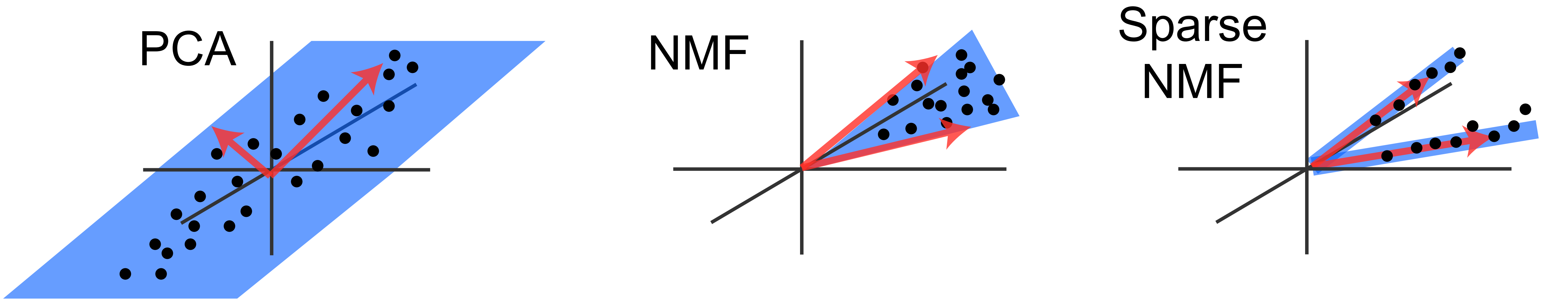
PCA



Sparse PCA



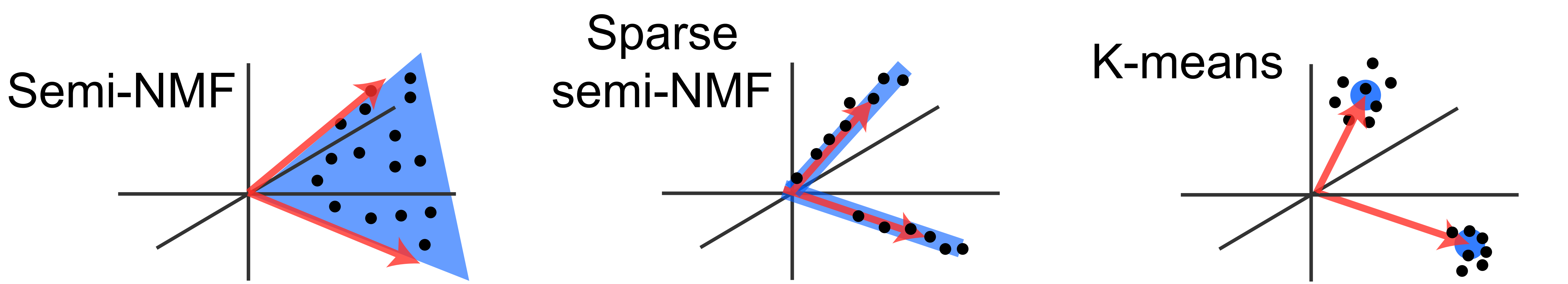
(D'Aspremont et al., 2007)



$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 \qquad \text{subject to} \quad \mathbf{U}^T\mathbf{U}=\mathbf{V}^T\mathbf{V}=\mathbf{I}$$

$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 \qquad \text{subject to} \quad \mathbf{U}\geq 0,\ \mathbf{V}\geq 0$$

$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2+\lambda_u\sum_i\|\mathbf{u}_{i:}\|_1 \qquad \text{subject to} \quad \mathbf{U}\geq 0,\ \mathbf{V}\geq 0$$



$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 \qquad \text{subject to} \quad \mathbf{U}\geq 0$$

$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2+\lambda_u\sum_i\|\mathbf{u}_{i:}\|_1 \qquad \text{subject to} \quad \mathbf{U}\geq 0$$

$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 \qquad \text{subject to} \quad \mathbf{u}_{i:}\in\{\mathbf{e}_k\},\forall i$$

Matrix decomposition can be interpreted probabilistically, via Bayes Rule:

$$p(\text{model} \mid \text{data}) = \frac{p(\text{data} \mid \text{model})\, p(\text{model})}{p(\text{data})}$$

Matrix decomposition can be interpreted probabilistically, via Bayes Rule:

$$\underbrace{p(\text{model} \mid \text{data})}_{\textit{posterior}} = \frac{\overbrace{p(\text{data} \mid \text{model})}^{\textit{likelihood}}\,\overbrace{p(\text{model})}^{\textit{prior}}}{p(\text{data})}$$

Matrix decomposition can be interpreted probabilistically, via Bayes Rule:

$$\underbrace{p(\text{model} \mid \text{data})}_{\textit{posterior}} = \frac{\overbrace{p(\text{data} \mid \text{model})}^{\textit{likelihood}}\,\overbrace{p(\text{model})}^{\textit{prior}}}{p(\text{data})}$$

$$-\ln p(\text{model} \mid \text{data}) \propto -\ln p(\text{data} \mid \text{model}) - \ln p(\text{model})$$

Matrix decomposition can be interpreted probabilistically, via Bayes Rule:

$$\underbrace{p(\text{model} \mid \text{data})}_{\textit{posterior}} = \frac{\overbrace{p(\text{data} \mid \text{model})}^{\textit{likelihood}}\;\overbrace{p(\text{model})}^{\textit{prior}}}{p(\text{data})}$$

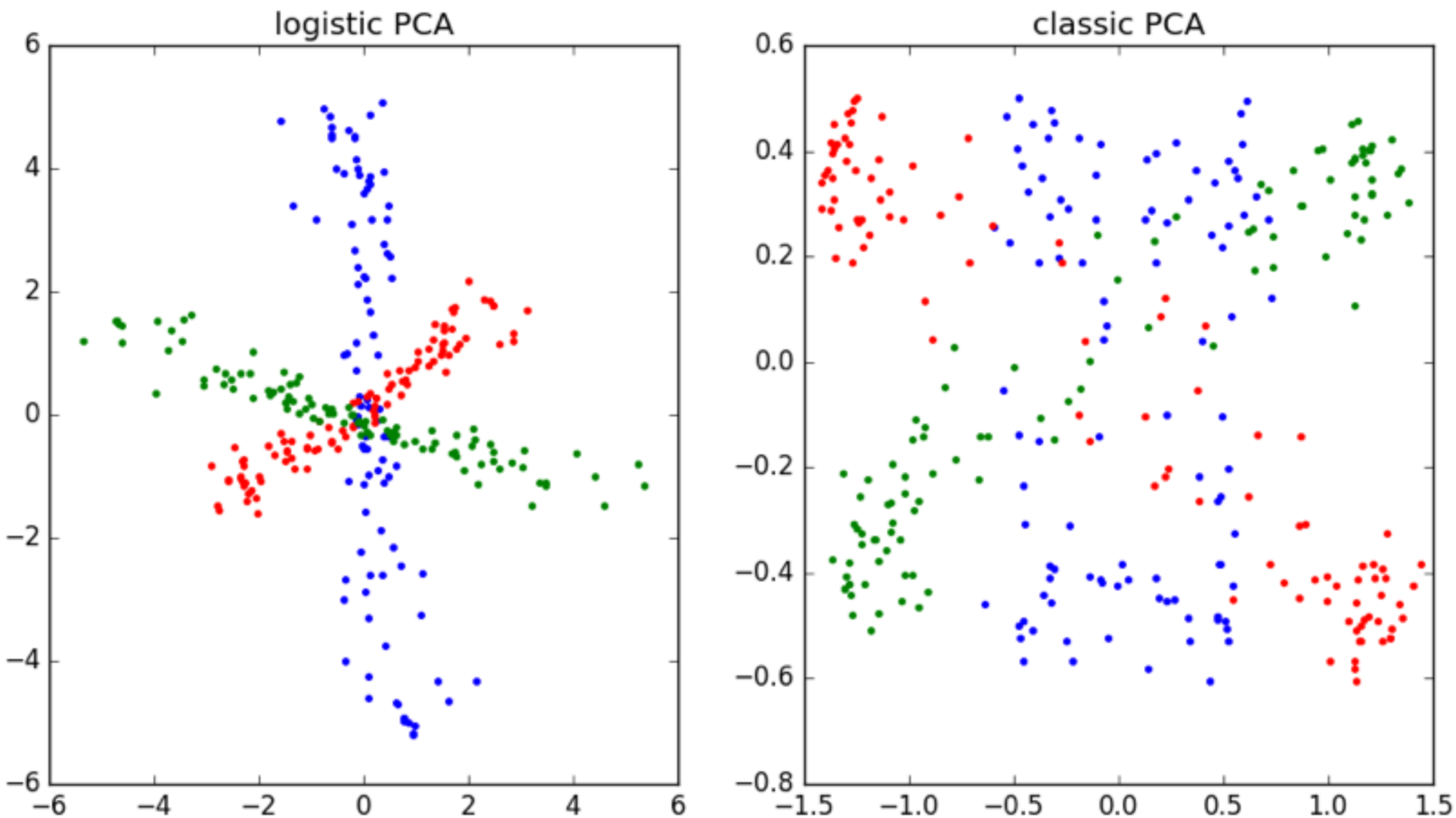
$$-\ln p(\text{model} \mid \text{data}) \propto -\ln p(\text{data} \mid \text{model}) - \ln p(\text{model})$$

Bottom Line: *Standard matrix decomposition can be viewed as maximum a posteriori estimation*

Loss functions often map onto the negative log-likelihood

Regularizers often map onto the prior distributions

Using the appropriate loss function can make a difference



http://alexhwilliams.info/itsneuronalblog/2016/03/27/pca/

Combinatorial menu of models

loss functions

quadratic
(real data)

absolute
(robust to outliers)

logistic
(binary data)

Poisson
(integer data)

circular
(angular data)

regularizers/constraints

L2 norm
(small factors)

L1 norm (sparsity)
(sparse factors)

Nonnegative
(additive factors)

Derivative penalties
(smooth factors)

Further Reading

Udell et al. (2016). **"Generalized Low Rank Models."** *Foundations and Trends in Machine Learning.*

Presents one of the most general matrix factorization frameworks that includes PCA, NMF, Sparse PCA, K-means, and many others as special cases.

Essid & Ozerov (2014). **Tutorial on NMF.** *ICME 2014.*

http://perso.telecom-paristech.fr/~essid/teach/NMF_tutorial_ICME-2014.pdf

A comprehensive overview of applications and extensions of NMF

Gillis (2011). **Nonnegative Matrix Factorization: Complexity, Algorithms, and Applications.** *PhD thesis, Université Catholique de Louvain.*

A very comprehensive thesis placing greater focus on the algorithmic aspects of NMF. Also see more recent work from Gillis.

Talk Outline

1. Long list of matrix decomposition models

2. Optimization and model fitting

3. Visualization and model assessment

Properties of PCA

Rotation problem limits interpretability. However, it also allows us to organize factors to have convenient properties.

Canonically, choose factors to be orthogonal and order them by variance explained.

Properties of PCA

Rotation problem limits interpretability. However, it also allows us to organize factors to have convenient properties.

Canonically, choose factors to be orthogonal and order them by variance explained.

Eckart-Young Theorem: solution given by truncated singular value decomposition (SVD)

Consequence: the solution with R components is contained in the solution with R+1 components.

Properties of PCA

PCA is one of the few examples of a nonconvex problem* that can be provably solved in polynomial time

* with a bit of work you can formulate a convex optimization problem whose solution also solves the PCA problem: http://www.stat.cmu.edu/~ryantibs/convexopt/lectures/nonconvex.pdf

Properties of PCA

PCA is one of the few examples of a nonconvex problem* that can be provably solved in polynomial time

Can prove that all local minima are solutions.

All non-optimal critical points are saddle points or maxima.

* with a bit of work you can formulate a convex optimization problem whose solution also solves the PCA problem:
http://www.stat.cmu.edu/~ryantibs/convexopt/lectures/nonconvex.pdf

Properties of PCA

PCA is one of the few examples of a nonconvex problem* that can be provably solved in polynomial time

Can prove that all local minima are solutions.

All non-optimal critical points are saddle points or maxima.

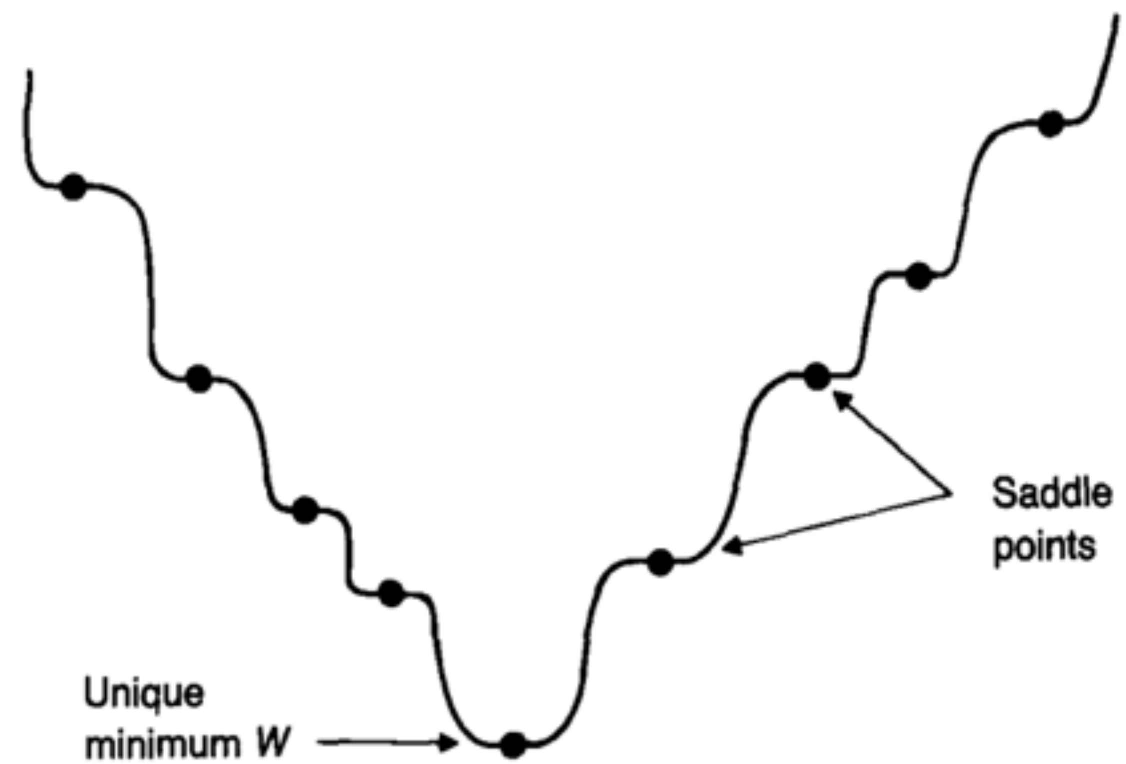
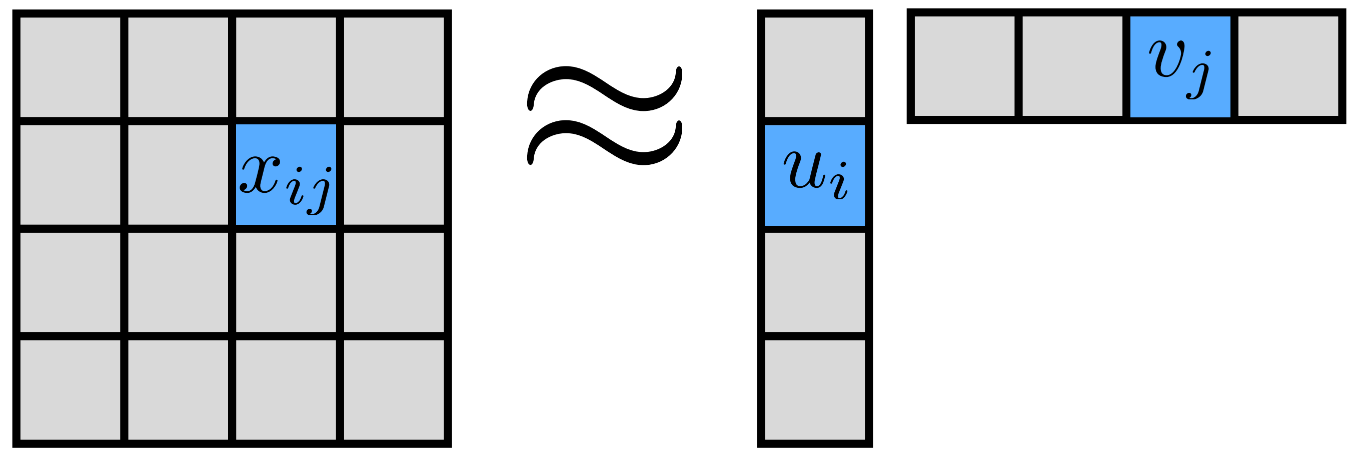


FIGURE 2. The landscape of *E*.

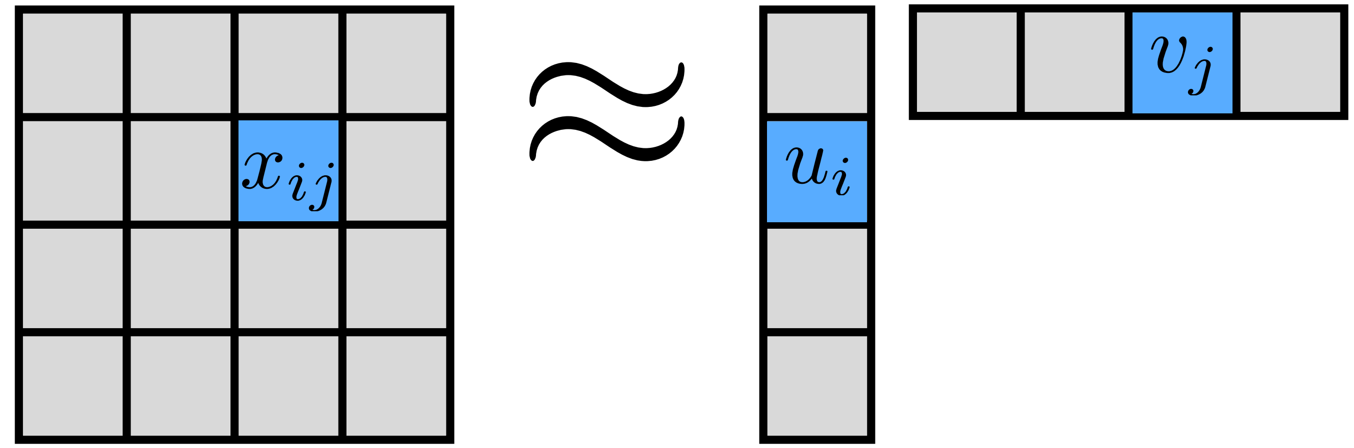
(Baldi & Hornik, 1989).

* with a bit of work you can formulate a convex optimization problem whose solution also solves the PCA problem: http://www.stat.cmu.edu/~ryantibs/convexopt/lectures/nonconvex.pdf

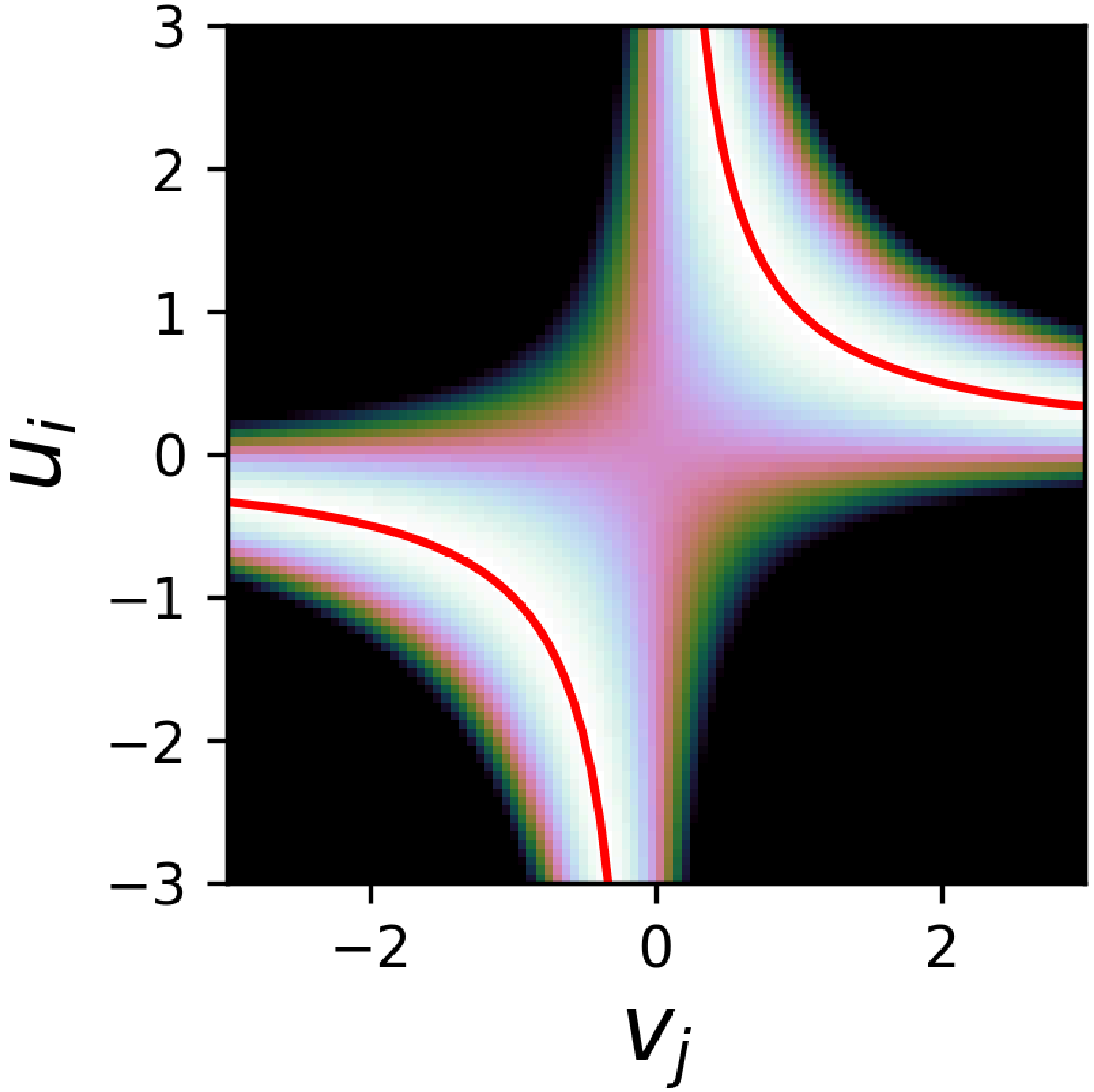
Consider the PCA loss for a single matrix element



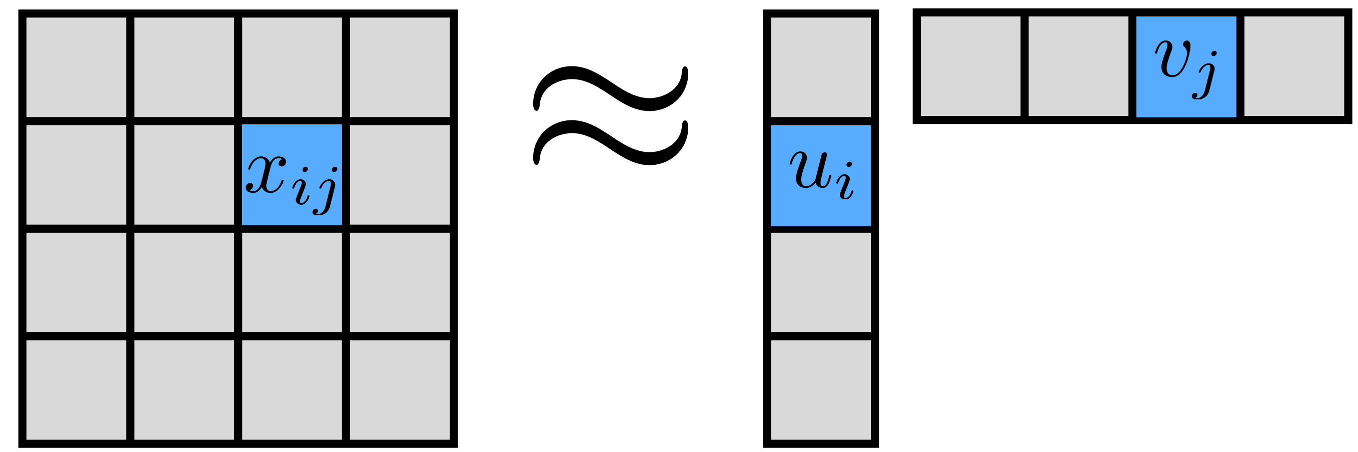
Consider the PCA loss for a single matrix element



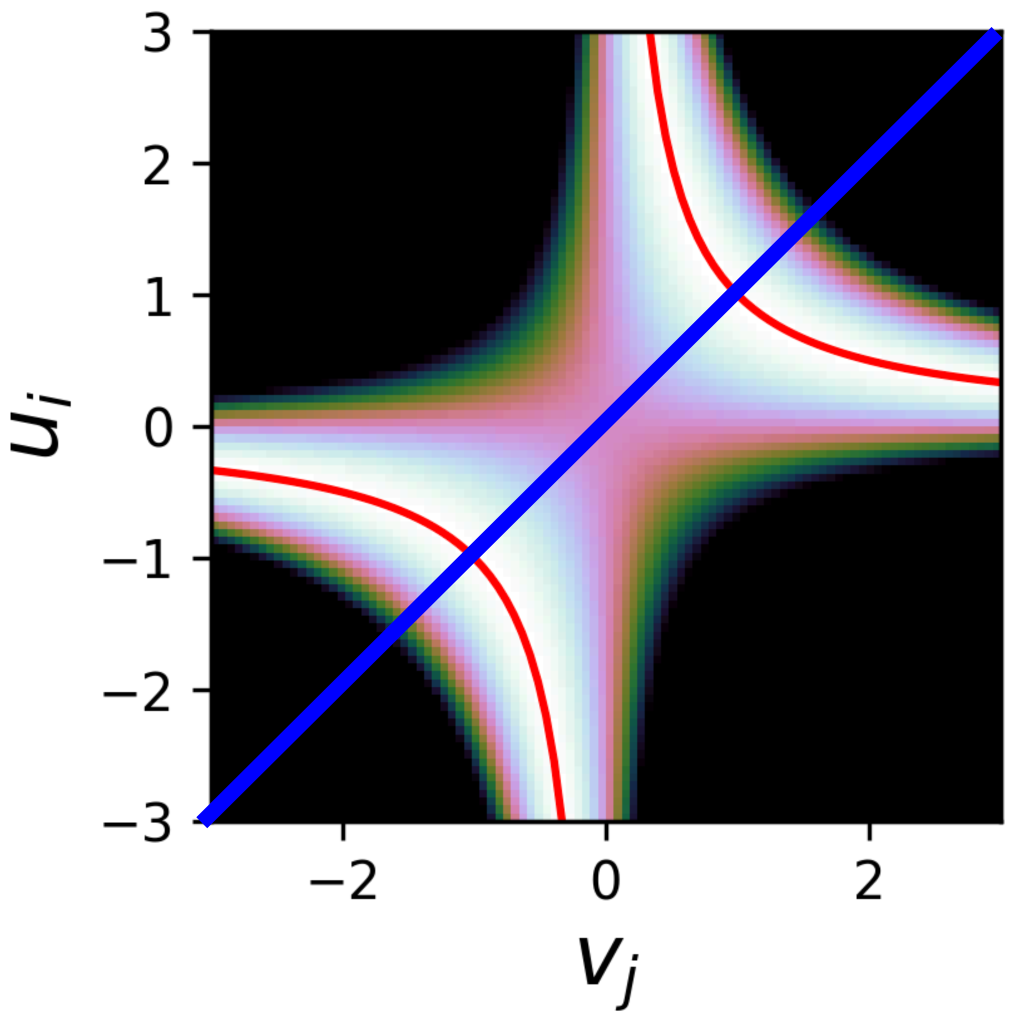
$$\ell_{ij}(u_i, v_j) = (x_{ij} - u_i v_j)^2$$

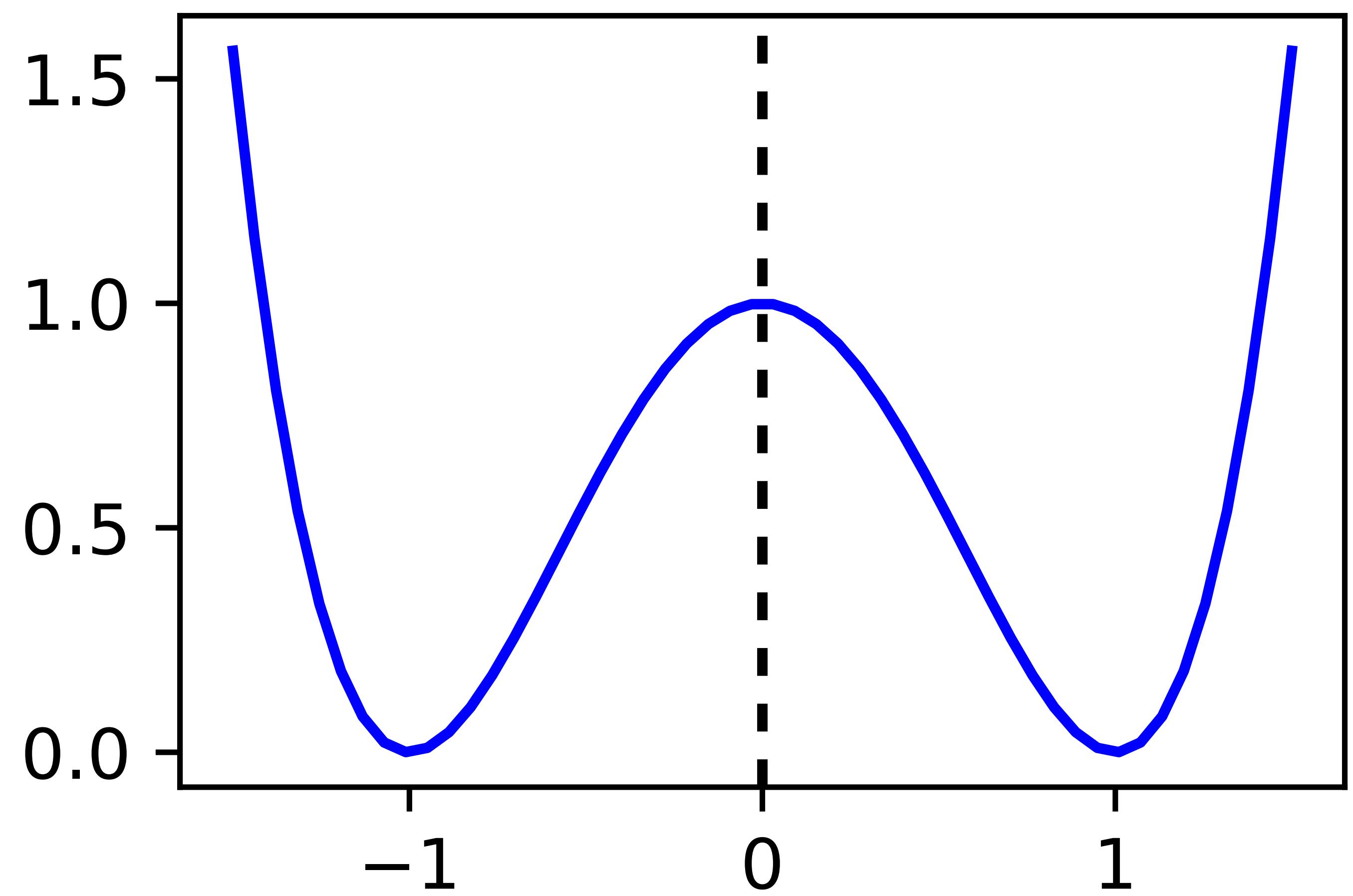


Consider the PCA loss for a single matrix element

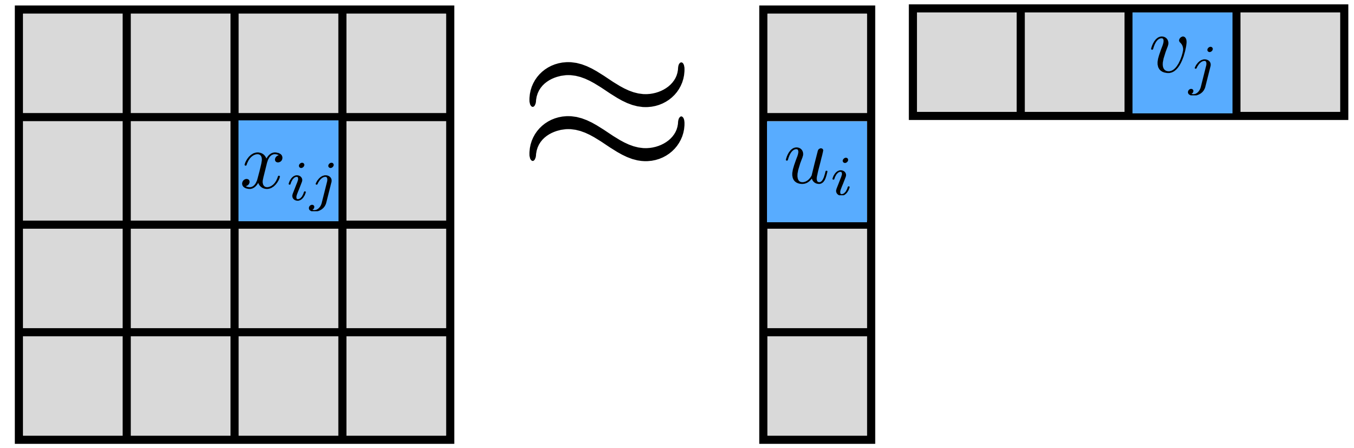


$$\ell_{ij}(u_i, v_j) = (x_{ij} - u_i v_j)^2$$

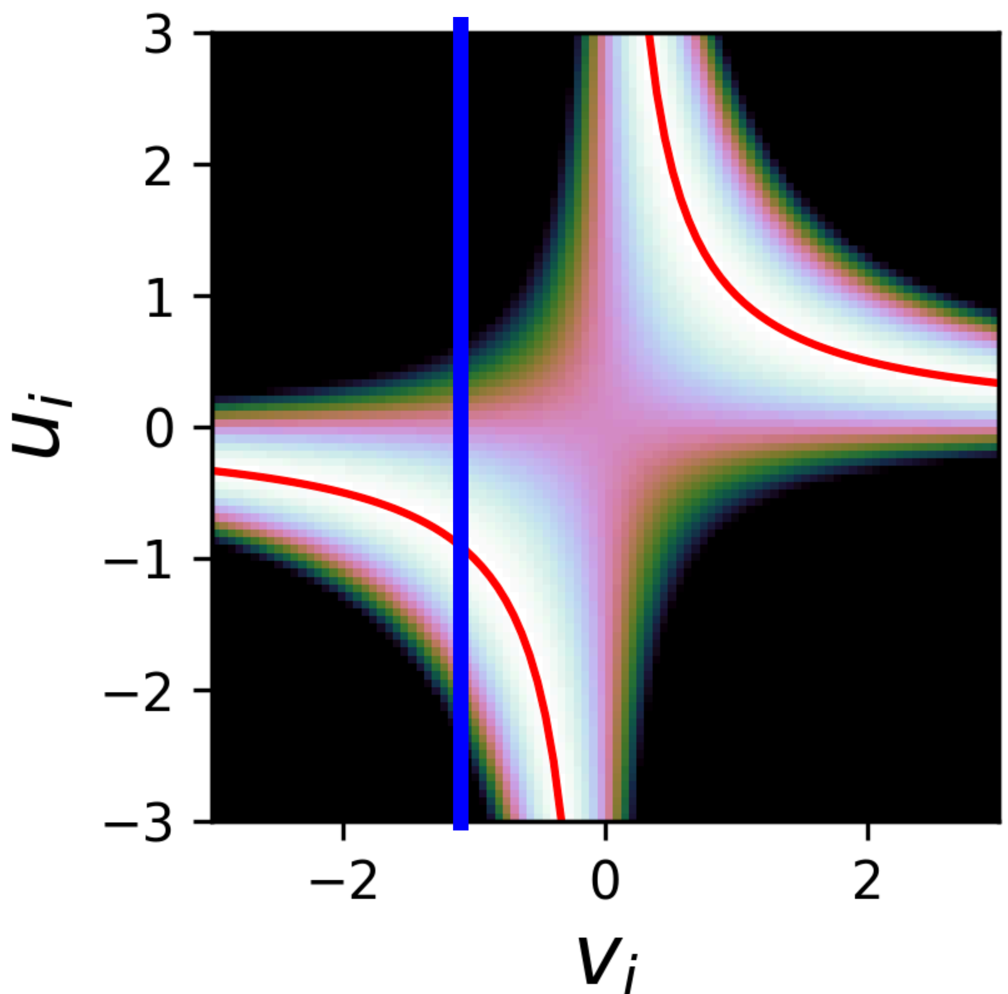




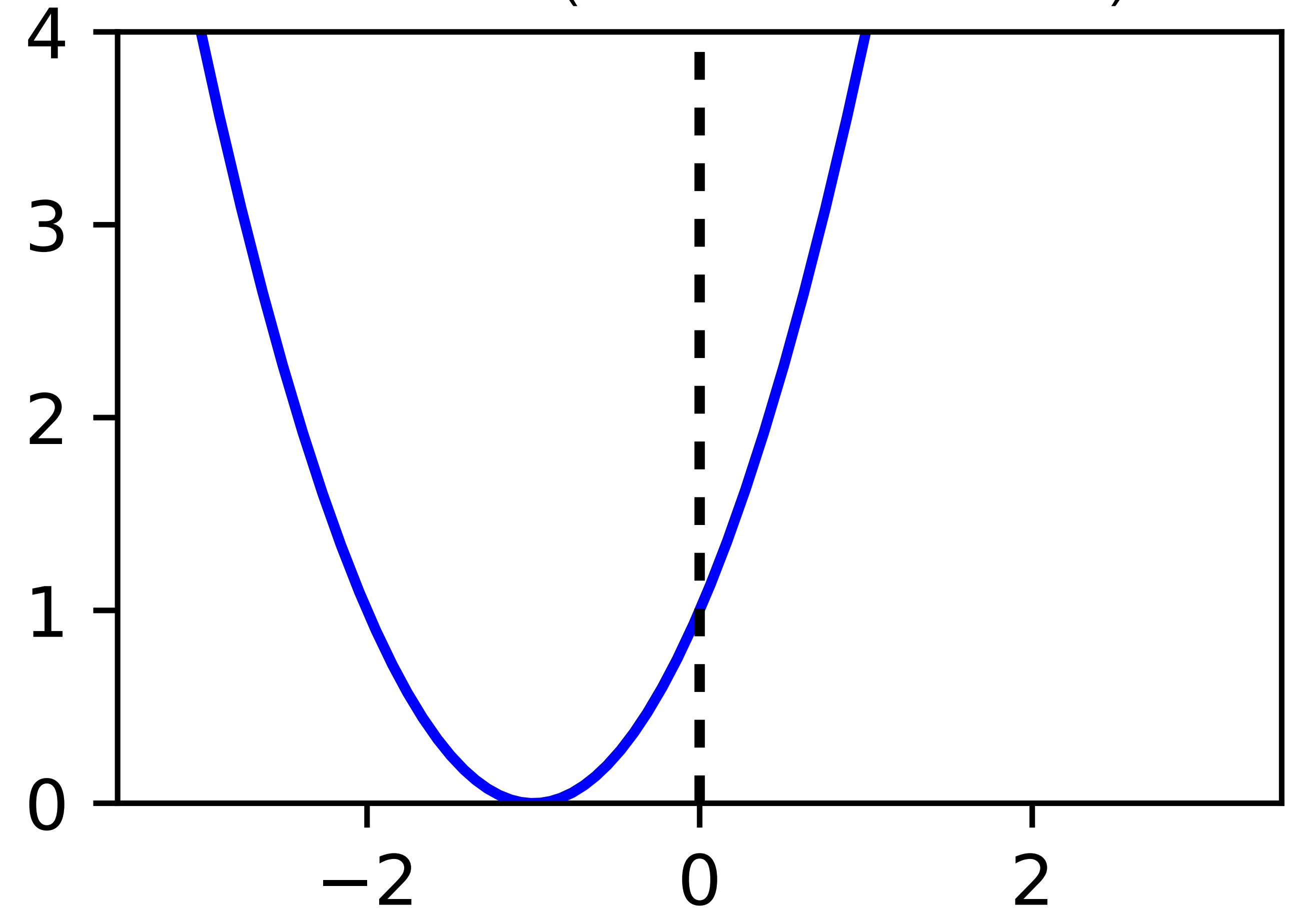
Consider the PCA loss for a single matrix element



$$\ell_{ij}(u_i, v_j) = (x_{ij} - u_i v_j)^2$$



Convex in **u** when **v** is fixed as constant (and vice versa)



Alternating Minimization

$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2$$

Decompose the loss function into two, easy to solve subproblems:

step 1: $\mathbf{U} \leftarrow \underset{\tilde{\mathbf{U}}}{\operatorname{argmin}} \|\mathbf{X} - \tilde{\mathbf{U}}\,\mathbf{V}^T\|_F^2$

step 2: $\mathbf{V} \leftarrow \underset{\tilde{\mathbf{V}}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{U}\,\tilde{\mathbf{V}}^T\|_F^2$

Repeat until loss function converges.

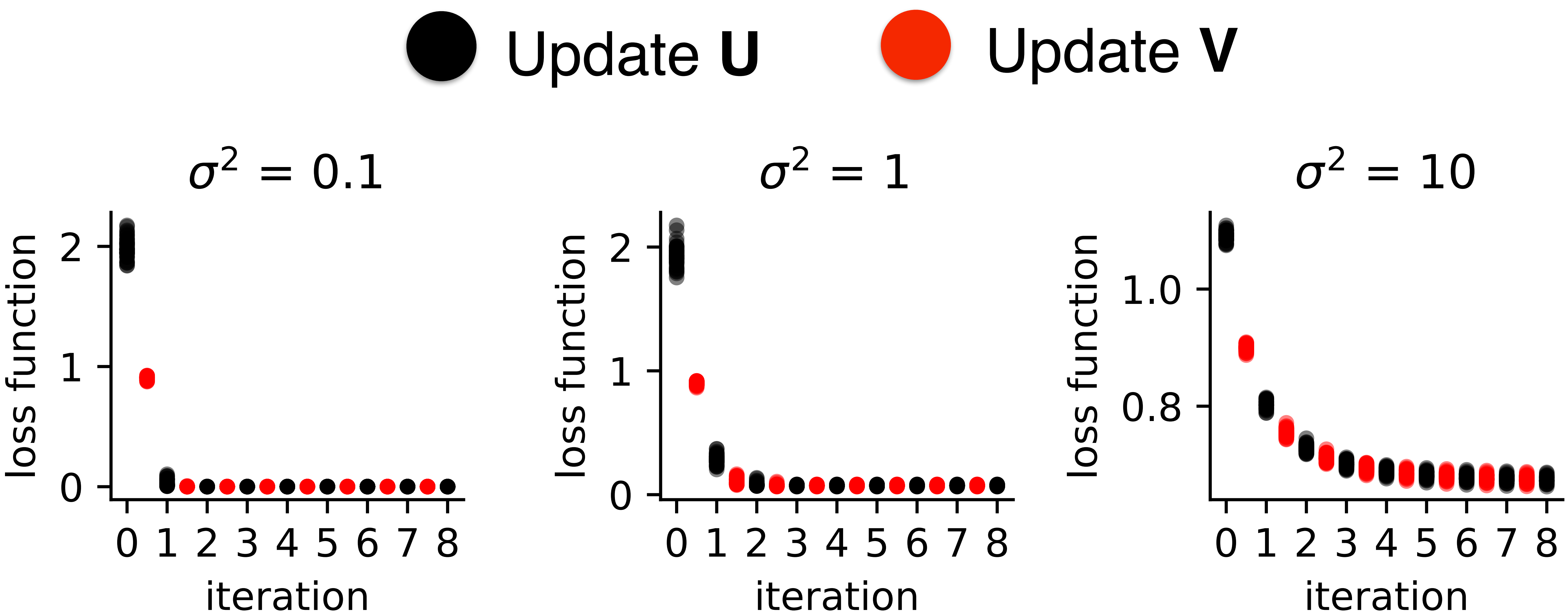
Fitting PCA in 10 lines of MATLAB

```
K = 3; % number of components
data = randn(100,K) * randn(K, 101);
[M, N] = size(data);
U = randn(M, K); % initial guess for U

for iteration = 1:10
    Vt = U \ data; % Update V (fixed U)
    U = data / Vt; % Update U (fixed V)
    loss(iteration) = norm(data - U*Vt, 'fro');
end
```

Alternating minimization is super effective in practice

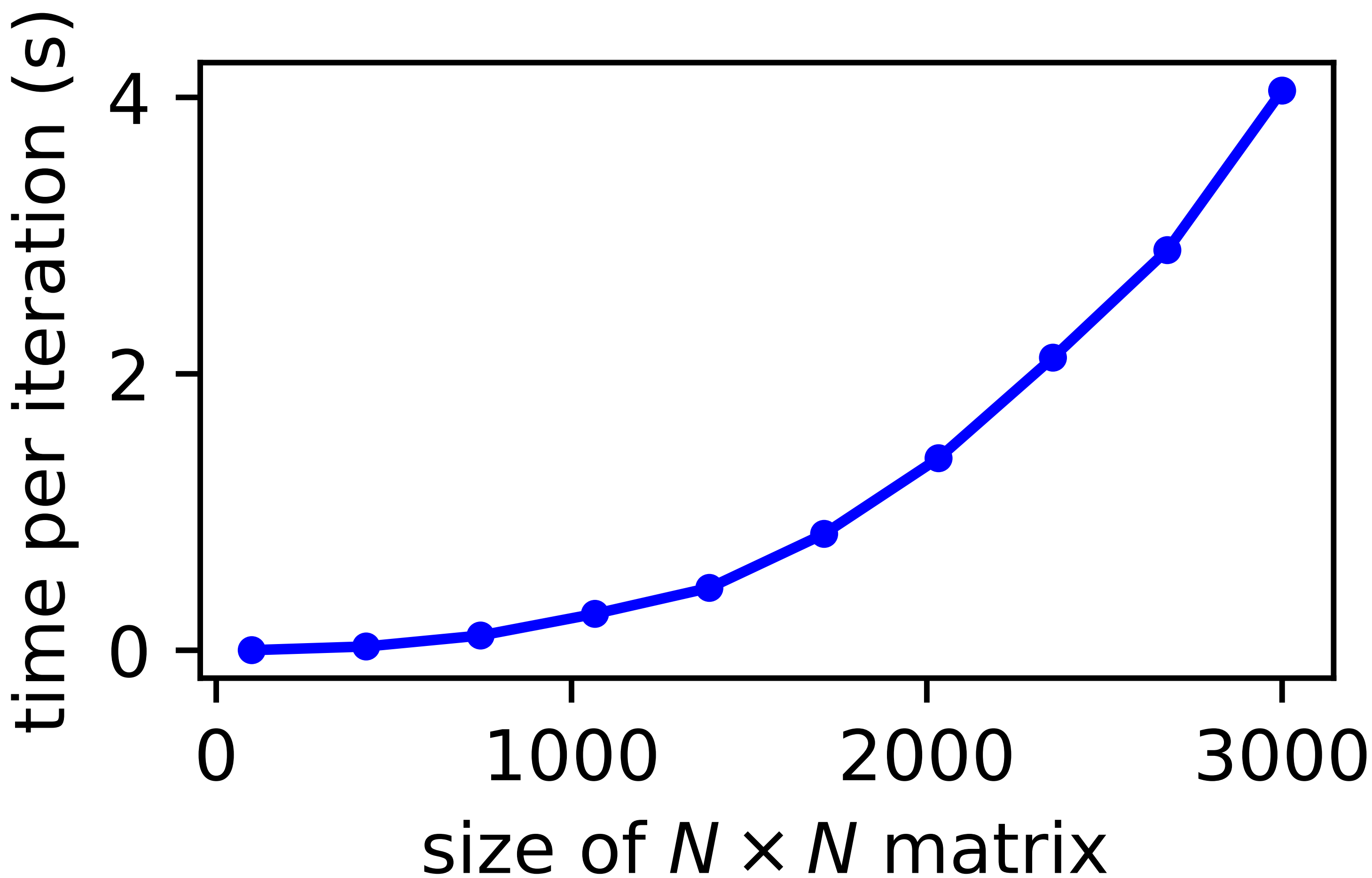
Generally, not that many iterations are needed.



Simulated 100x100 data matrix, with 10 components

Alternating minimization is super effective in practice

For moderate data sizes, iterations are fast.



*Time to perform 1 update of **U** and **V** on my MacBook Pro*

NMF can also be solved by alternating minimization

Each step is *nonnegative least squares* problem

$$\mathbf{U} \leftarrow \underset{\boxed{\tilde{\mathbf{U}}\geq 0}}{\operatorname{argmin}} \ \|\mathbf{X}- \ \tilde{\mathbf{U}} \ \mathbf{V}^T\|_F^2$$

$$\mathbf{V} \leftarrow \underset{\boxed{\tilde{\mathbf{V}}\geq 0}}{\operatorname{argmin}} \ \|\mathbf{X} - \mathbf{U} \ \tilde{\mathbf{V}}^T\|_F^2$$

NMF can also be solved by alternating minimization

Each step is *nonnegative least squares* problem

Convex problem

Specialized, fast optimization methods

(e.g. Kim & Park, 2008)

$$\mathbf{U} \leftarrow \underset{\boxed{\tilde{\mathbf{U}} \geq 0}}{\operatorname{argmin}} \|\mathbf{X} - \tilde{\mathbf{U}}\,\mathbf{V}^T\|_F^2$$

$$\mathbf{V} \leftarrow \underset{\boxed{\tilde{\mathbf{V}} \geq 0}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{U}\,\tilde{\mathbf{V}}^T\|_F^2$$

NMF can also be solved by alternating minimization

Each step is *nonnegative least squares* problem

Convex problem

Specialized, fast optimization methods

(e.g. Kim & Park, 2008)

$$\mathbf{U} \leftarrow \underset{\tilde{\mathbf{U}} \geq 0}{\operatorname{argmin}} \|\mathbf{X} - \tilde{\mathbf{U}}\, \mathbf{V}^T\|_F^2$$

$$\mathbf{V} \leftarrow \underset{\tilde{\mathbf{V}} \geq 0}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{U}\, \tilde{\mathbf{V}}^T\|_F^2$$

In MATLAB:

```
x = lsqnonneg(A, b);
```

In Python:

```
import scipy.optimize
x = scipy.optimize.nnls(A, b)
```

Further reading on optimization

Kim et al. (2014). **"Algorithms for nonnegative matrix and tensor factorizations."** *Journal of Global Optimization.*

> A unified review that covers alternating minimization along with other specialized methods for fitting NMF.

Parikh & Boyd. (2016). **"Proximal Methods."** *Foundations and Trends in Machine Learning.*

> An overview of a very simple, but powerful class of optimization methods for matrix optimization. Udell et al. (2016), cited earlier, make use of these methods.

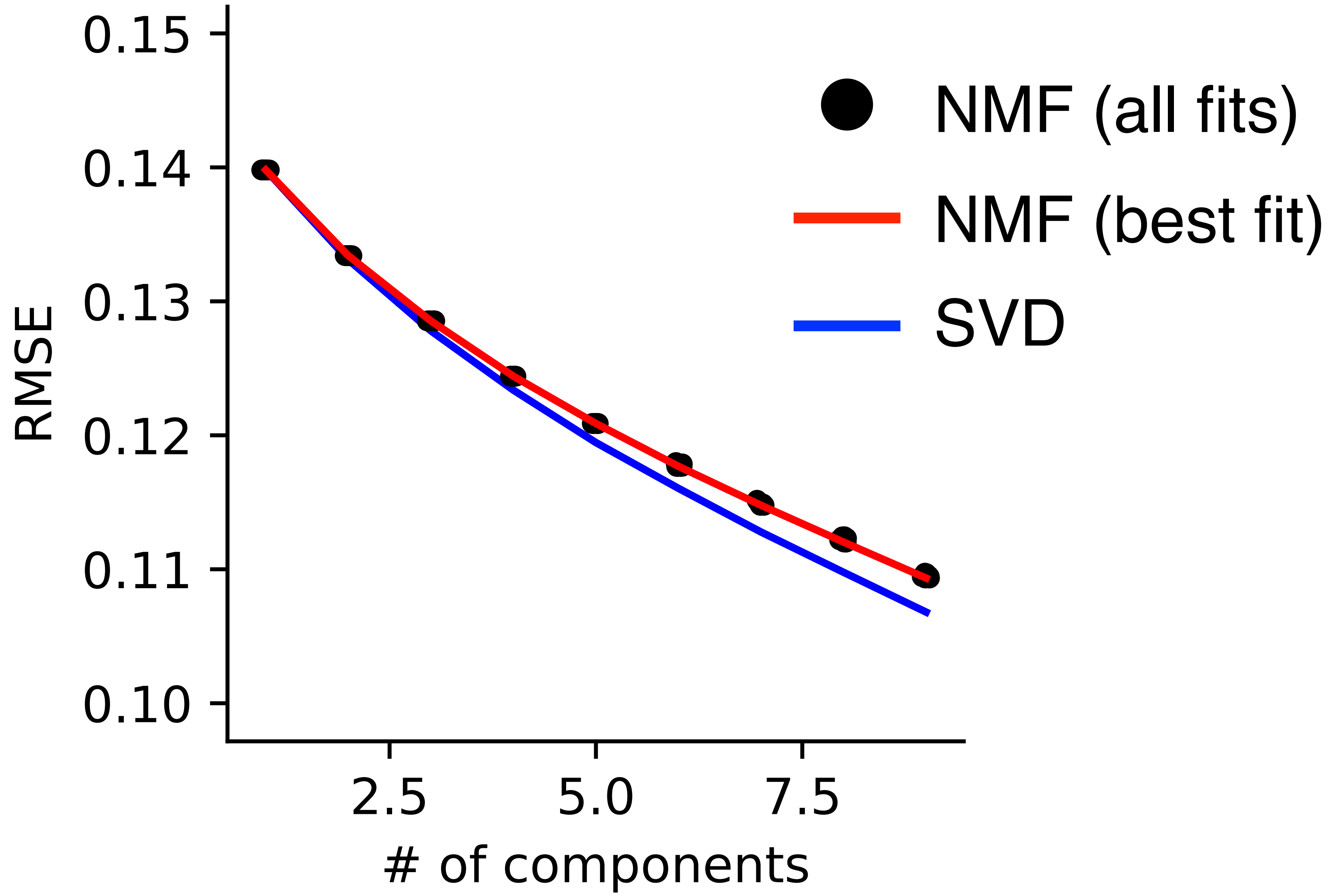
Talk Outline

1. Long list of matrix decomposition models

2. Optimization and model fitting

3. **Visualization and model assessment**

Scree Plot — How well am I fitting the data?



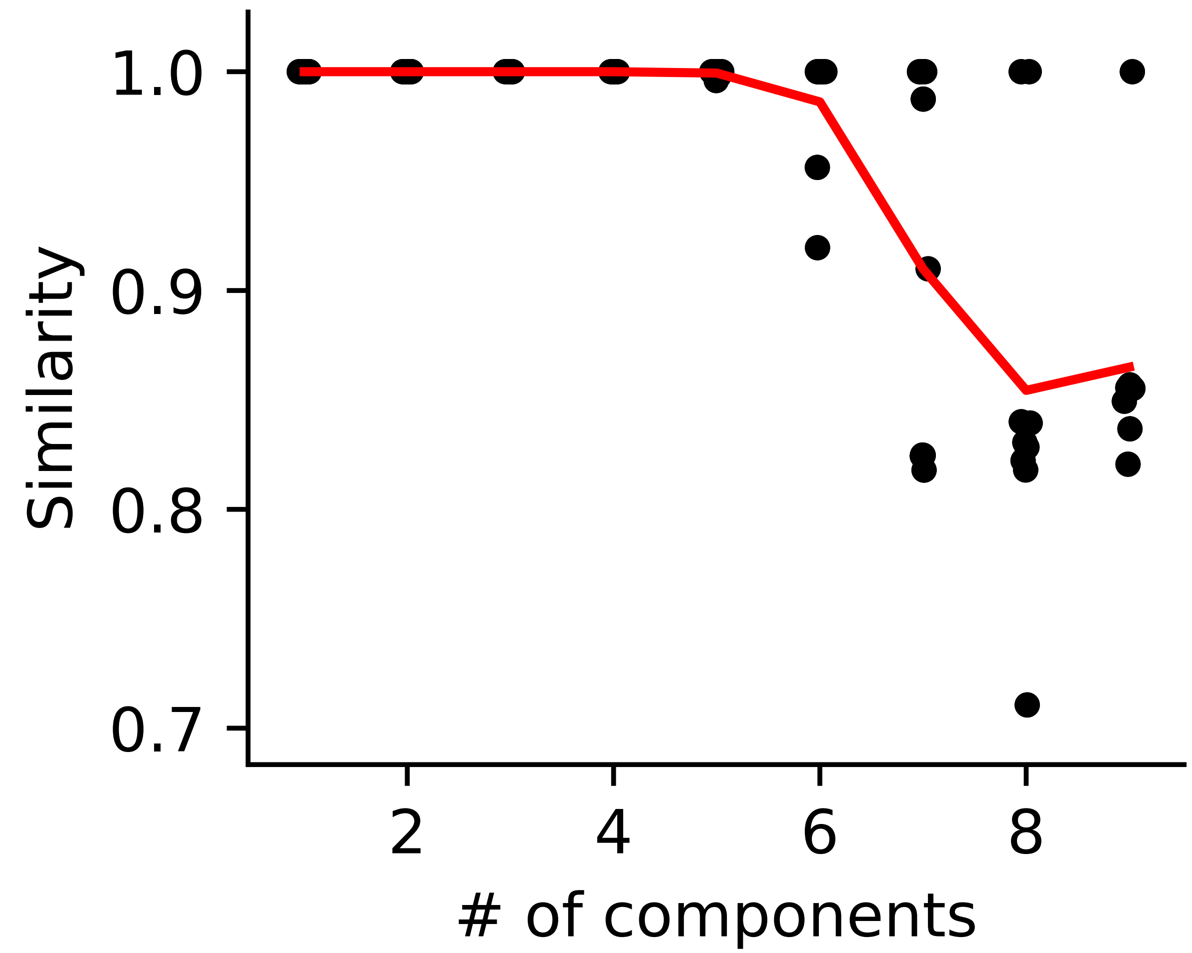
Interpretation: NMF converges to similar error from different initializations, and nearly achieves the optimal lower bound on performance set by SVD.

Similarity Plot — Are there multiple solutions that fit the data equally well?

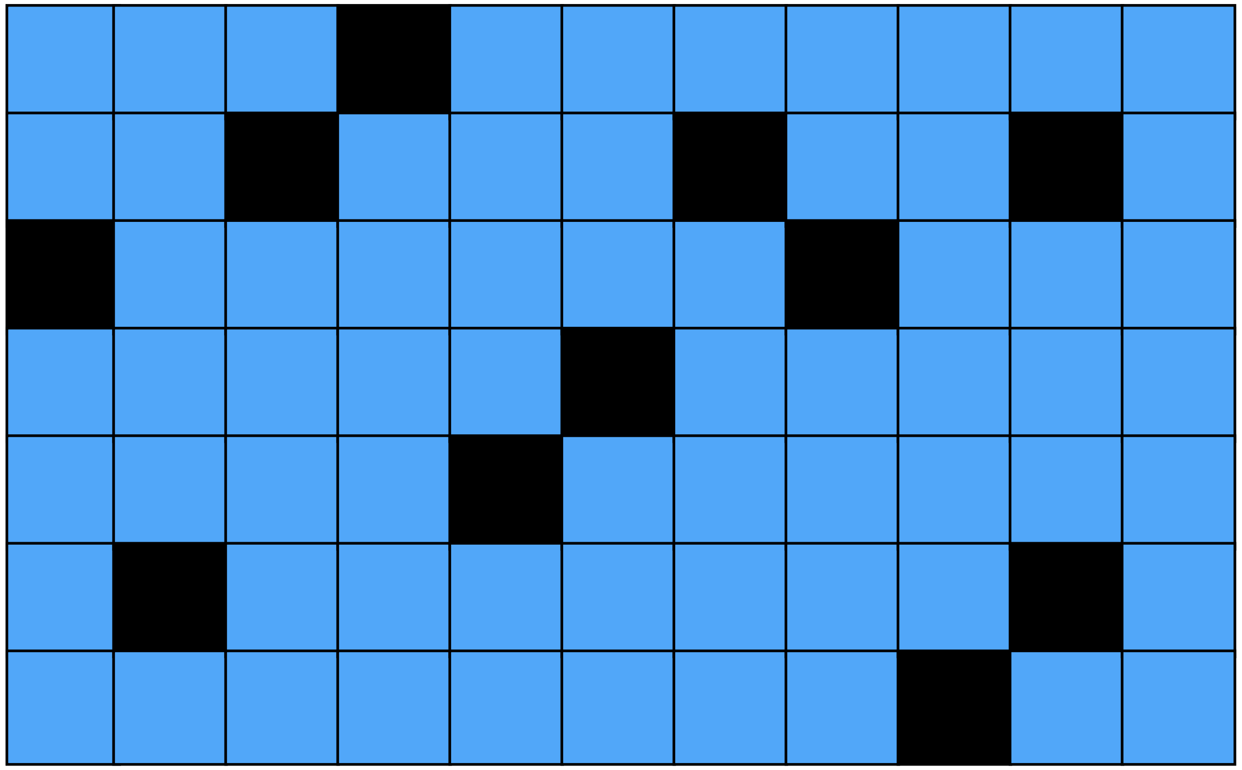
Define the similarity of two factor matrices as:

$$S(\mathbf{U}, \mathbf{U}') = \max_{\Pi} \quad \frac{1}{r} \mathrm{Tr}\left[\mathbf{U}^T \mathbf{U}' \Pi\right]$$

where Π, is an r x r permutation matrix.



Cross-Validation



training data

held-out data

Holding out data at random for cross-validation draws a connection to the well-studied matrix completion problem (see e.g. Candès & Recht, 2009)

Further reading on model assessment

Luxburg. (2010). **"Clustering Stability: An Overview."** *Foundations and Trends in Machine Learning.*

The subtle concepts behind the similarity plot are much better studied for clustering algorithms (rather than NMF). This review covers that literature.

Bro et al. (2008). **"Cross-validation of component models: a critical look at current methods."** *Analytical and Bioanalytical Chemistry*

An in-depth look at cross-validation procedures for PCA and other matrix factorization approaches.