Agenda

https://github.com/neurostatslab/factorization-tutorial

Lecture **(10-11am)**

Exercises on NMF **(11:15am - 12:15pm)**

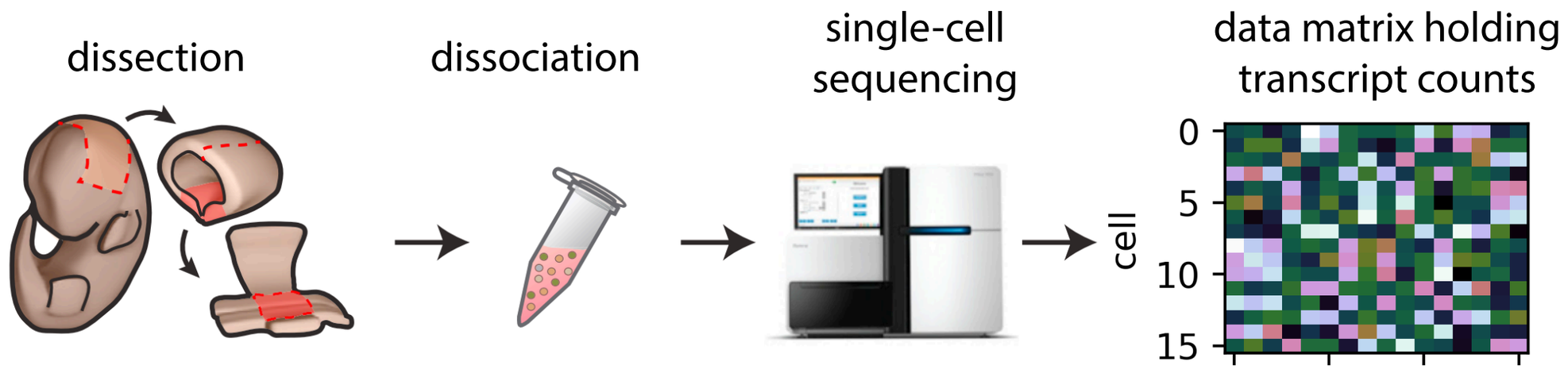
Dimensionality Reduction for Matrix- and Tensor-Coded Data

T32 Workshop @ NYU

08/17/2022

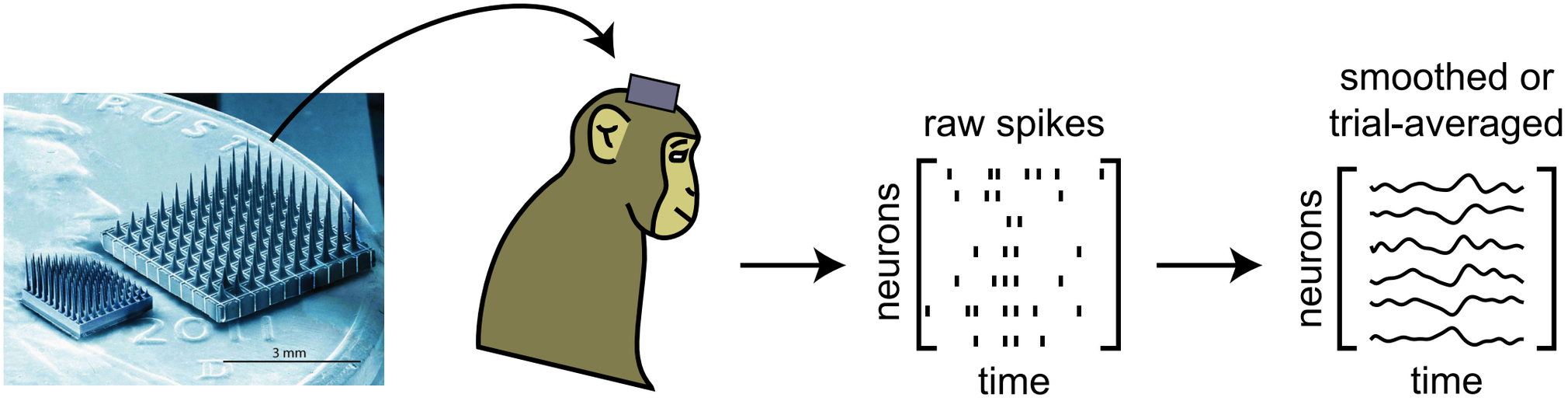
Examples of Matrix-Encoded Data

1. Gene Expression



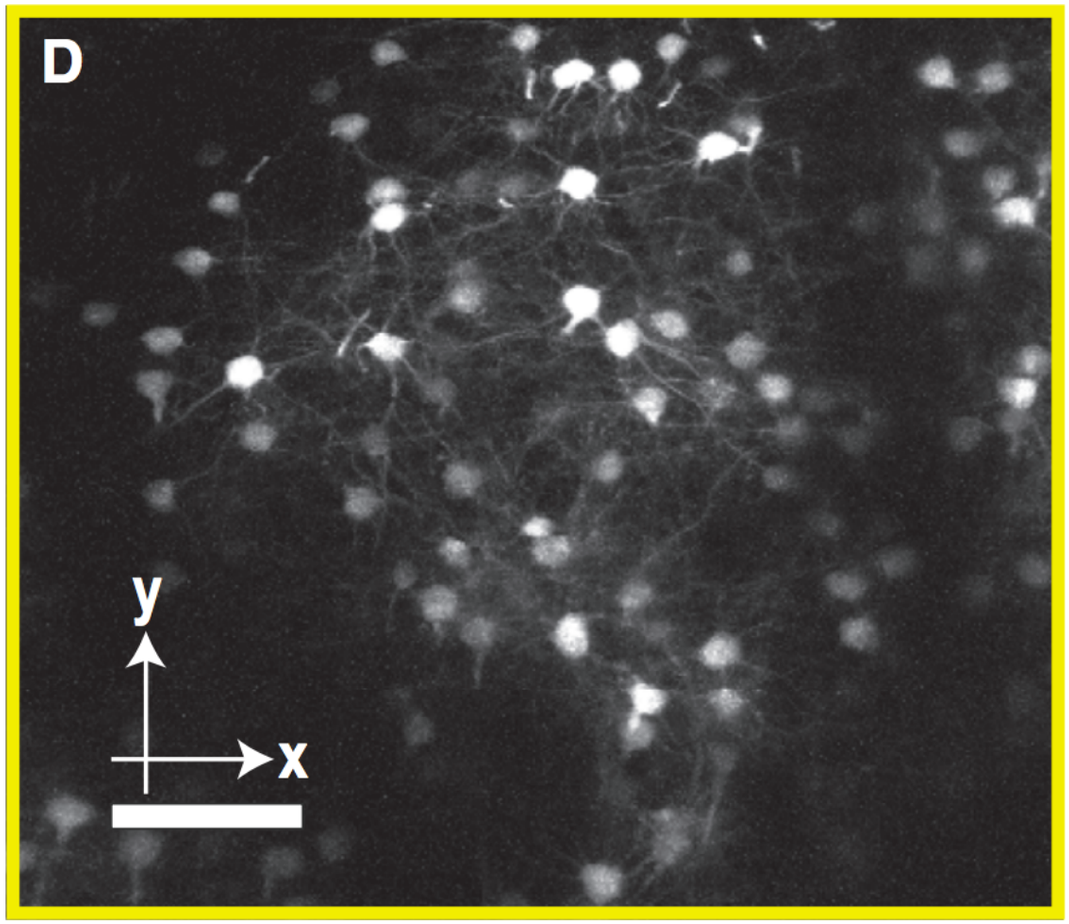
(schematic adapted from La Manno & Gyllborg et al., 2016)

2. Neural Activity



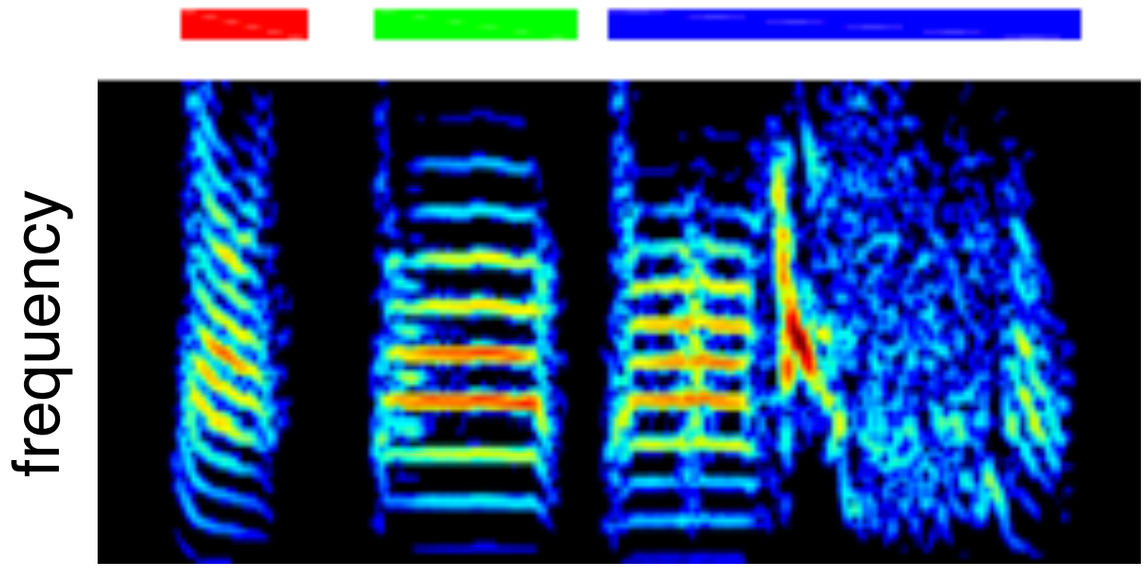
Examples of Matrix-Encoded Data

3. Fluorescence Images



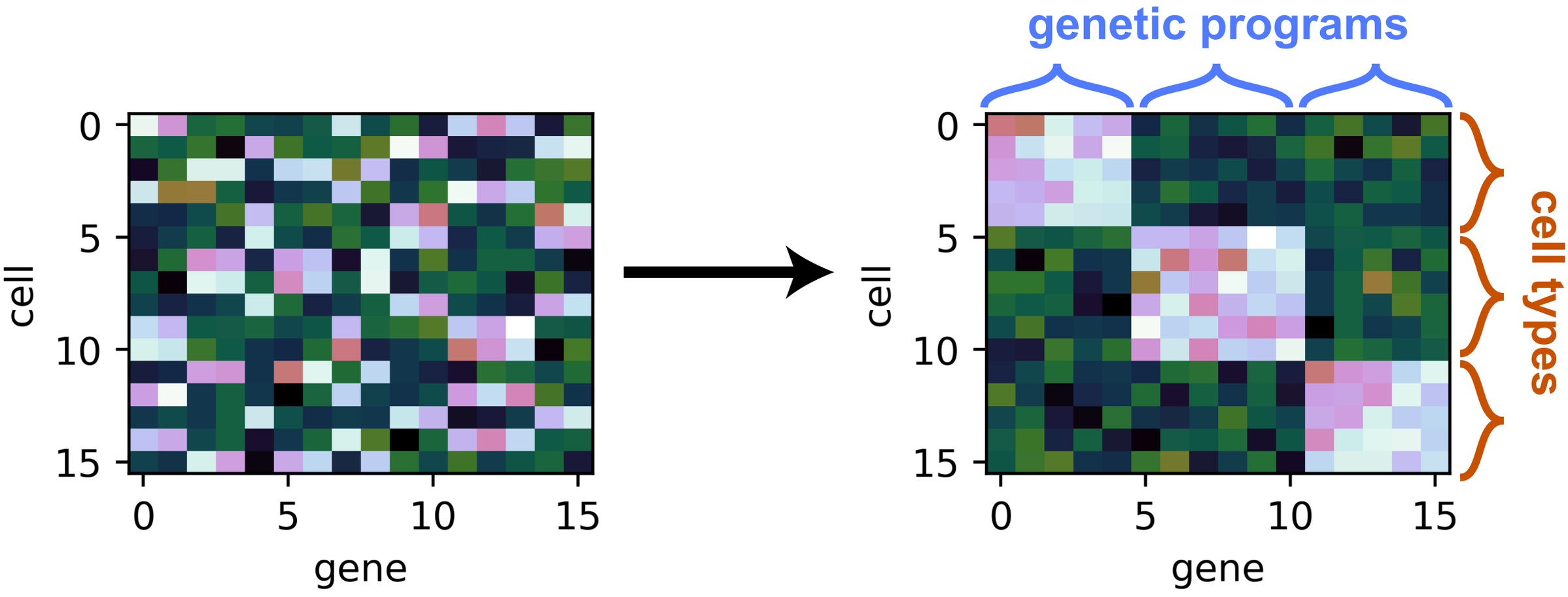
Cortical neurons expressing YFP
(Kim & Zhang et al., 2016)

4. Spectrograms



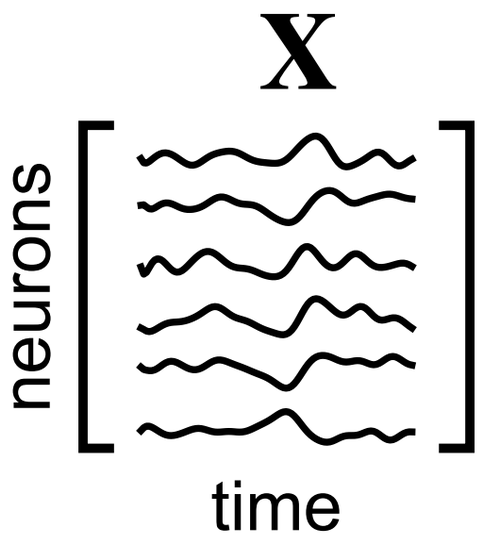
Zebra Finch courtship song
(Provided by Emily Mackevicius)

Goal: extract simple structure from these large-scale datasets



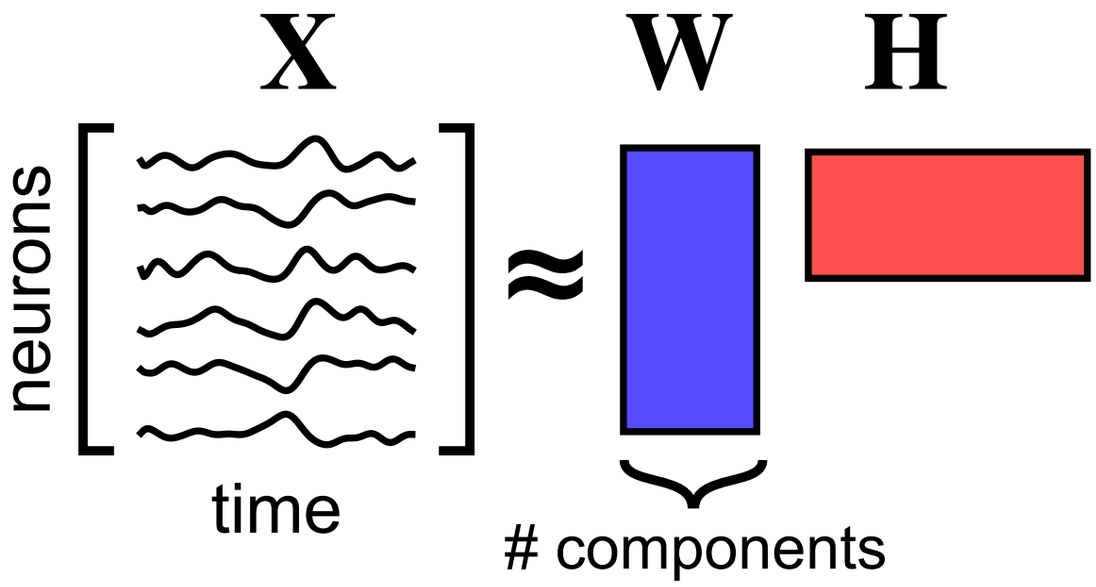
Matrix Factorization:

A general approach to compress matrix-coded data



Matrix Factorization:

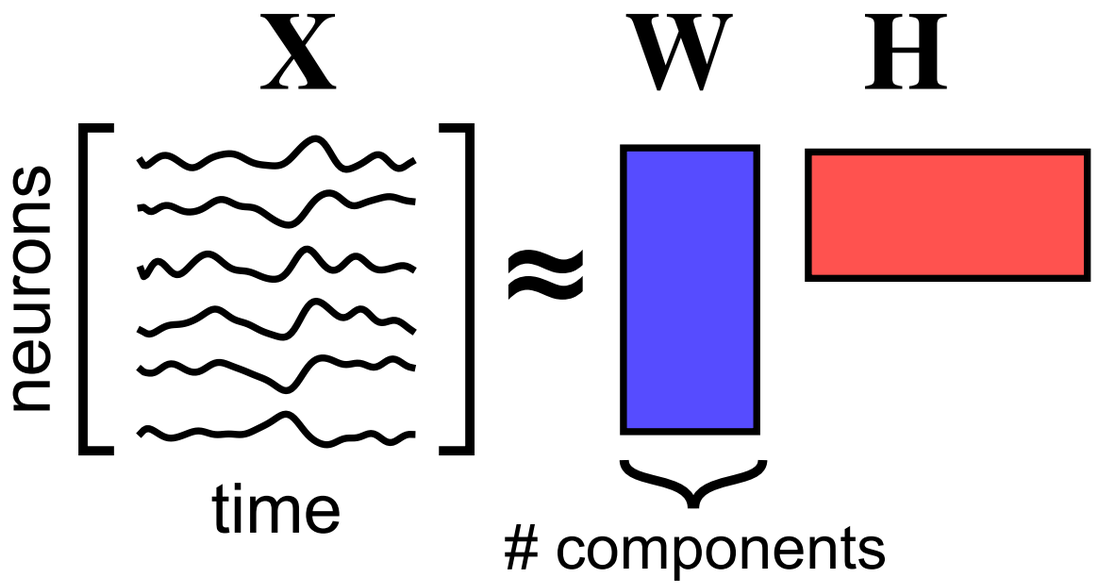
A general approach to compress matrix-coded data



Matrix Factorization:

A general approach to compress matrix-coded data

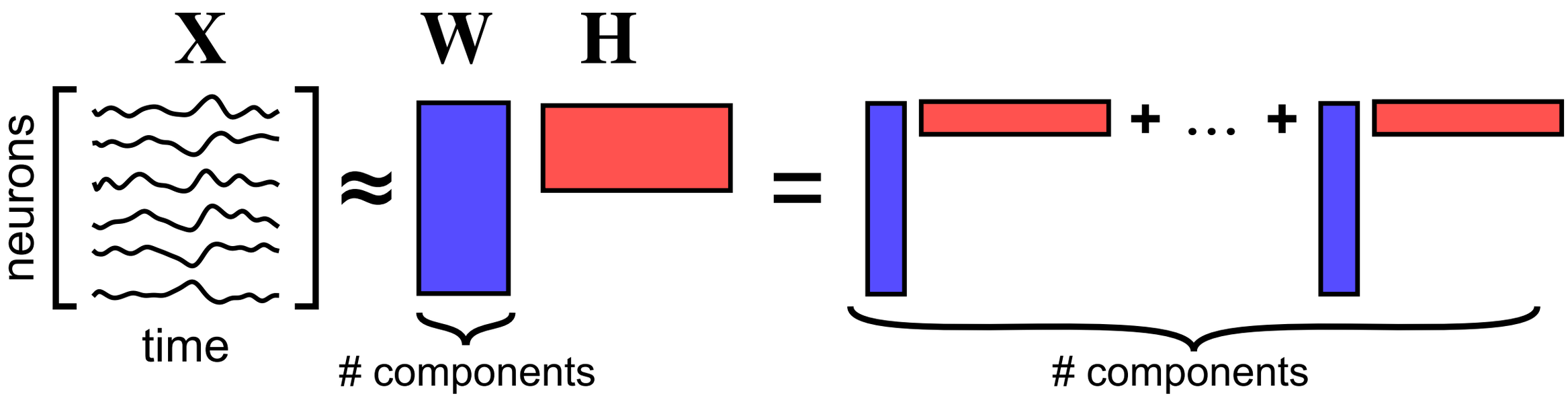
$$X_{ij} = \sum_{r=1}^{R} W_{ir} \cdot H_{rj}$$



Matrix Factorization:

A general approach to compress matrix-coded data

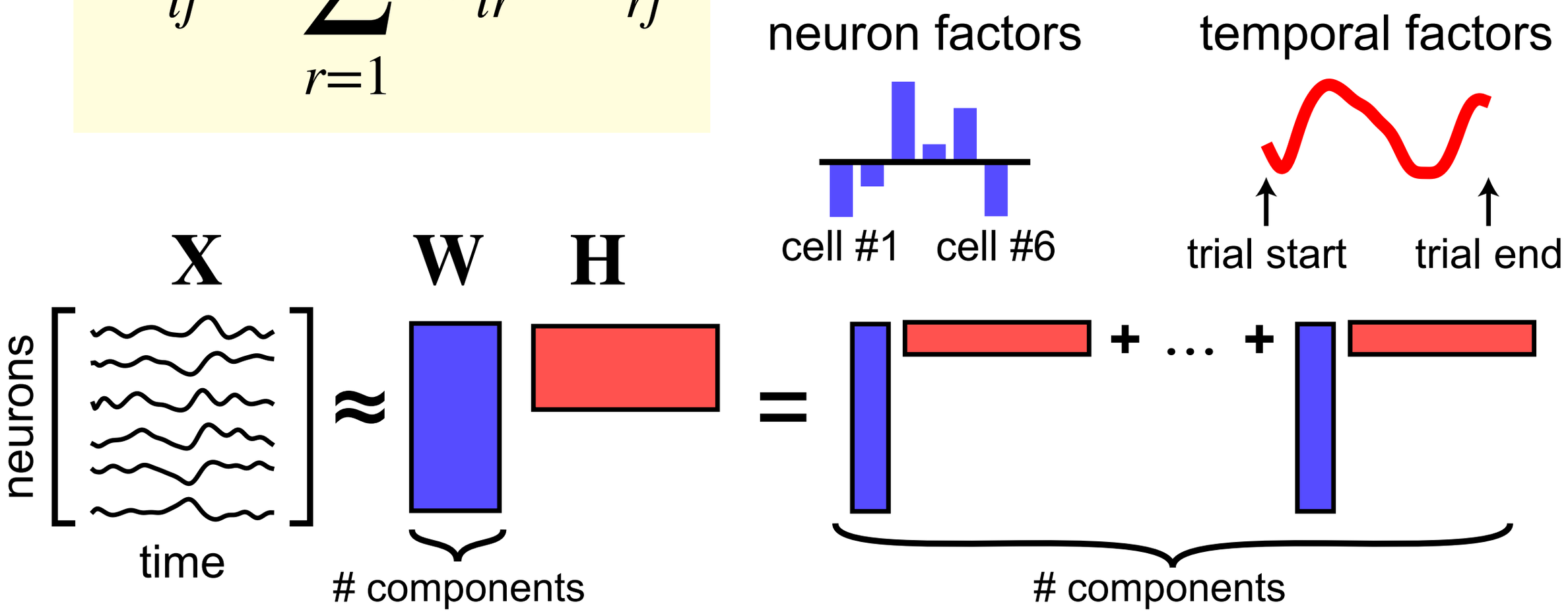
$$X_{ij} = \sum_{r=1}^{R} W_{ir} \cdot H_{rj}$$



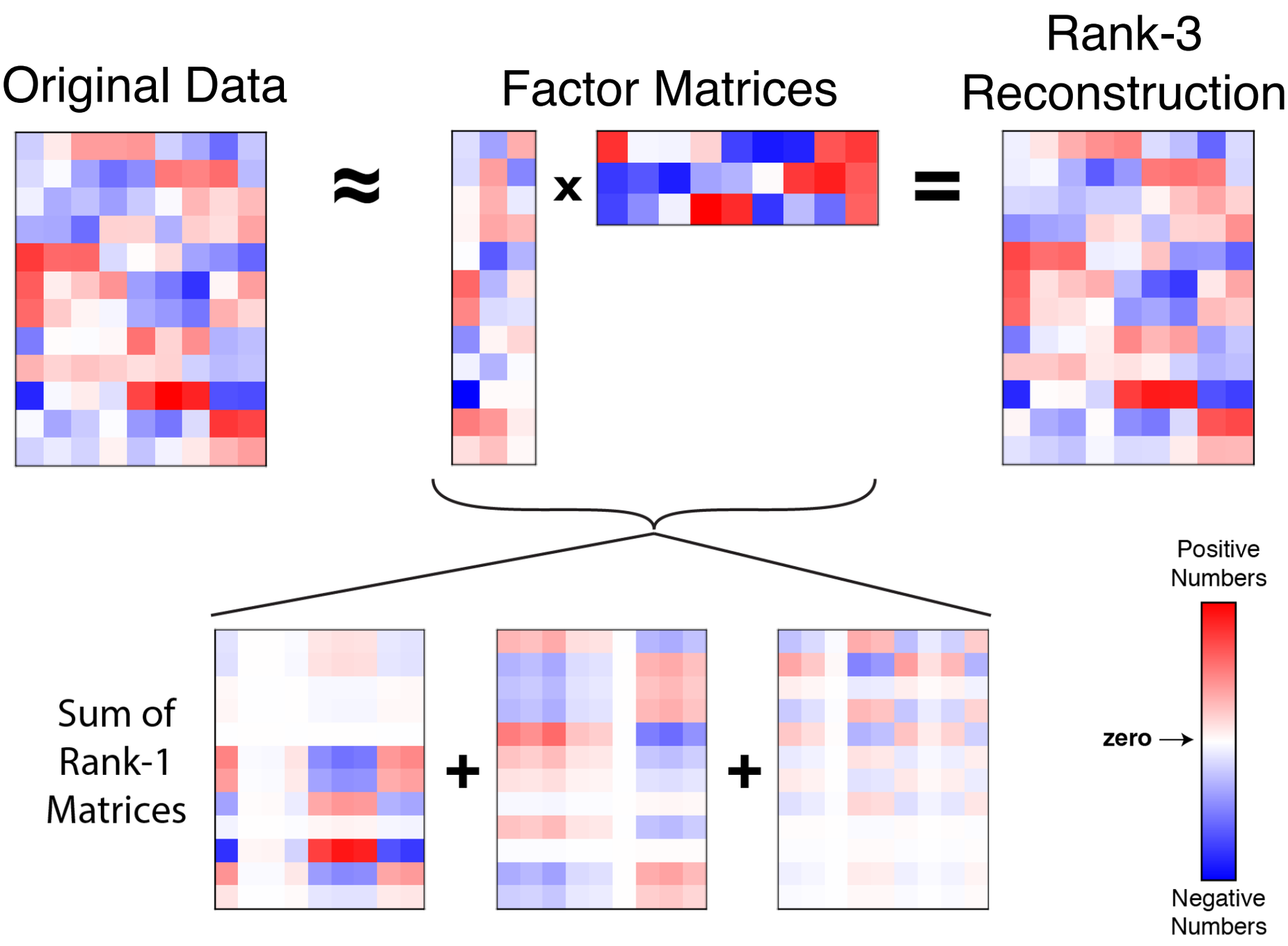
Matrix Factorization:

A general approach to compress matrix-coded data

$$X_{ij} = \sum_{r=1}^{R} W_{ir} \cdot H_{rj}$$



Visualization of Matrix Decomposition



Talk Outline

1. Long list of matrix decomposition models

2. ~~Optimization and model fitting~~

3. Visualization and model assessment

4. Tensor decomposition

Matrix decomposition model, stated formally

loss *regularization*

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

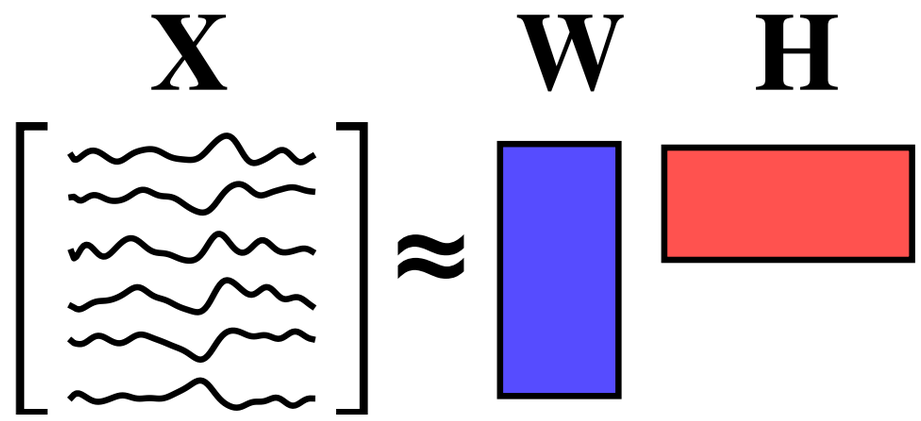
$$\text{subject to} \quad \mathbf{U} \in \Omega_u,\ \mathbf{V} \in \Omega_v$$

constraints

The simplest matrix decomposition is PCA*

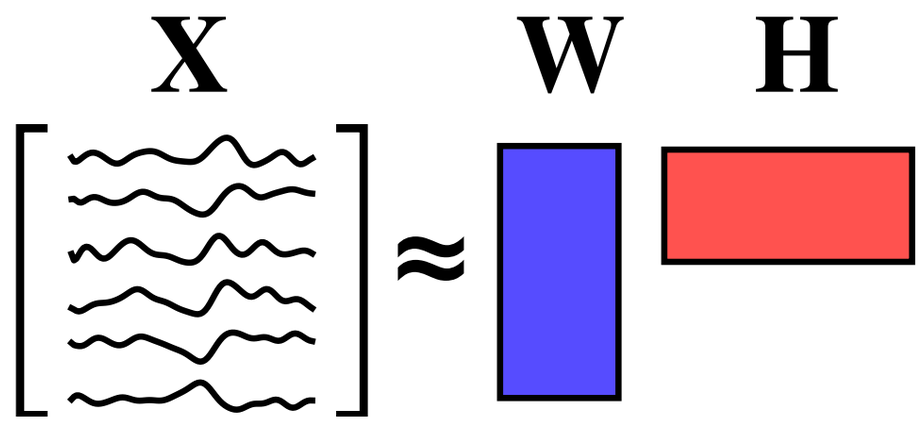
The simplest matrix decomposition is PCA*

$$\underset{\mathbf{W},\mathbf{H}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{WH}\|_F^2$$



The simplest matrix decomposition is PCA*

$$\underset{\mathbf{W},\mathbf{H}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{WH}\|_F^2$$



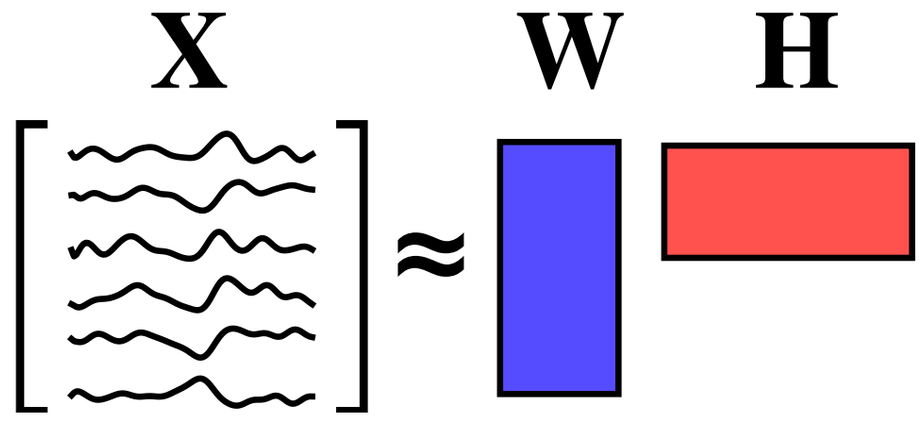
Asterisk: The principal components provide *one solution* to the above optimization problem.

PCA builds in additional constraints that factors are orthogonal. We will see this doesn't matter much.

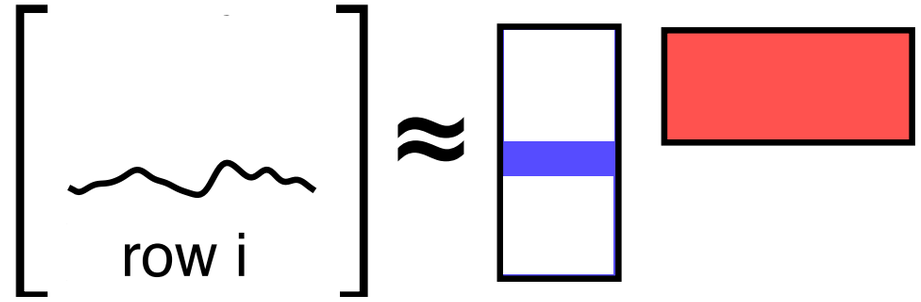
PCA also typically assumes the data have been mean-subtracted, which we will brush under the rug here.

The simplest matrix decomposition is PCA*

$$\underset{\mathbf{W},\mathbf{H}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{WH}\|_F^2$$

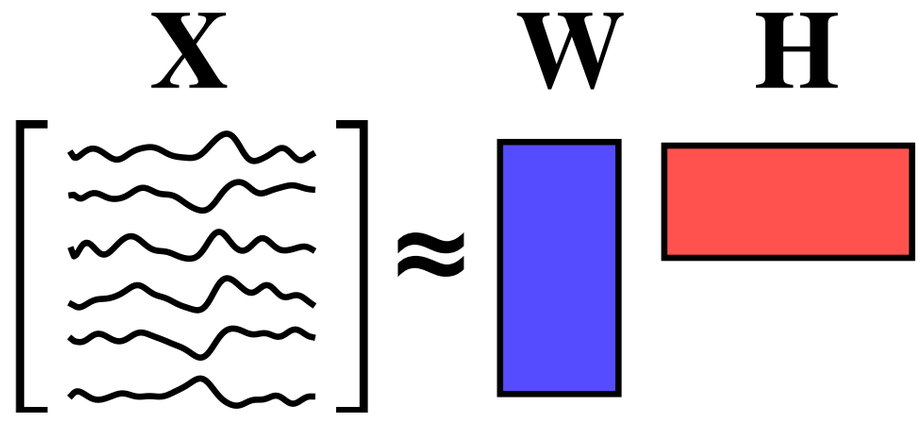


$$\underset{\mathbf{W},\mathbf{H}}{\text{minimize}} \quad \sum_{i=1}^{m} \left\| \mathbf{x}_{i\cdot} - \mathbf{H}\mathbf{w}_{i\cdot} \right\|_2^2$$

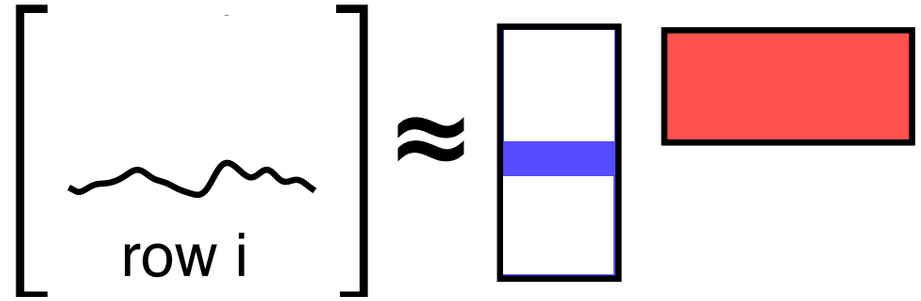


The simplest matrix decomposition is PCA*

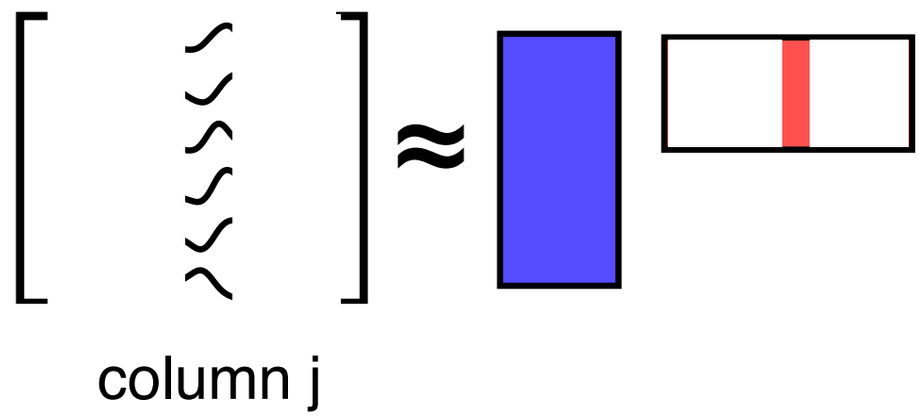
$$\underset{\mathbf{W},\mathbf{H}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{WH}\|_F^2$$



$$\underset{\mathbf{W},\mathbf{H}}{\text{minimize}} \quad \sum_{i=1}^{m} \left\|\mathbf{x}_{i\cdot} - \mathbf{H}\mathbf{w}_{i\cdot}\right\|_2^2$$

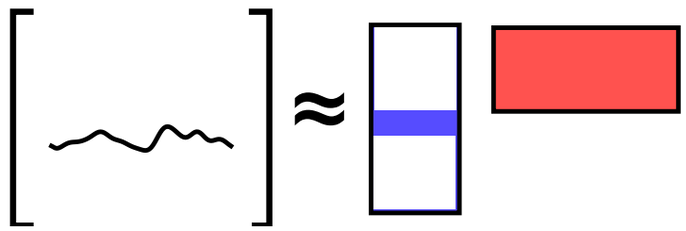


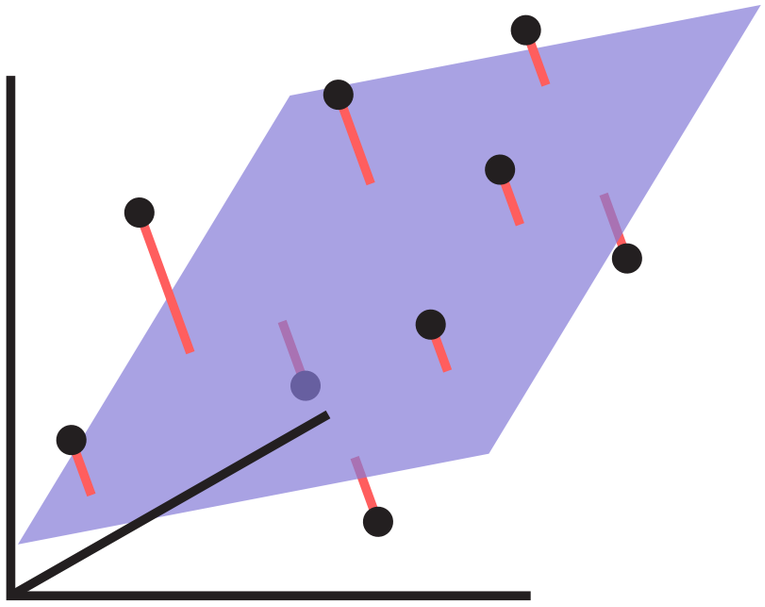
$$\underset{\mathbf{W},\mathbf{H}}{\text{minimize}} \quad \sum_{j=1}^{n} \left\|\mathbf{x}_{\cdot j} - \mathbf{W}\mathbf{h}_{\cdot j}\right\|_2^2$$



Two views of PCA* on an $m \times n$ matrix

$$\mathbf{x}_{i\cdot} \approx \mathbf{H}\mathbf{w}_{i\cdot}$$

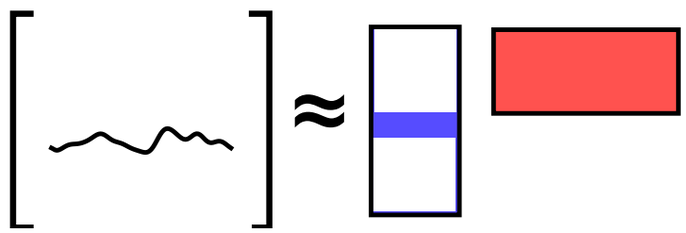




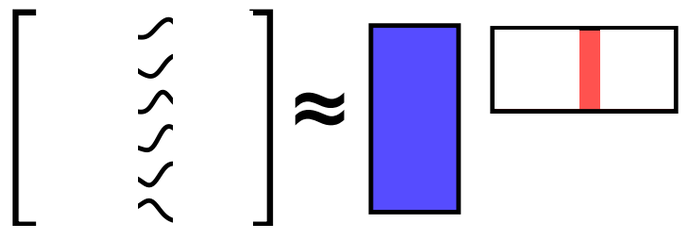
m points in n dimensions

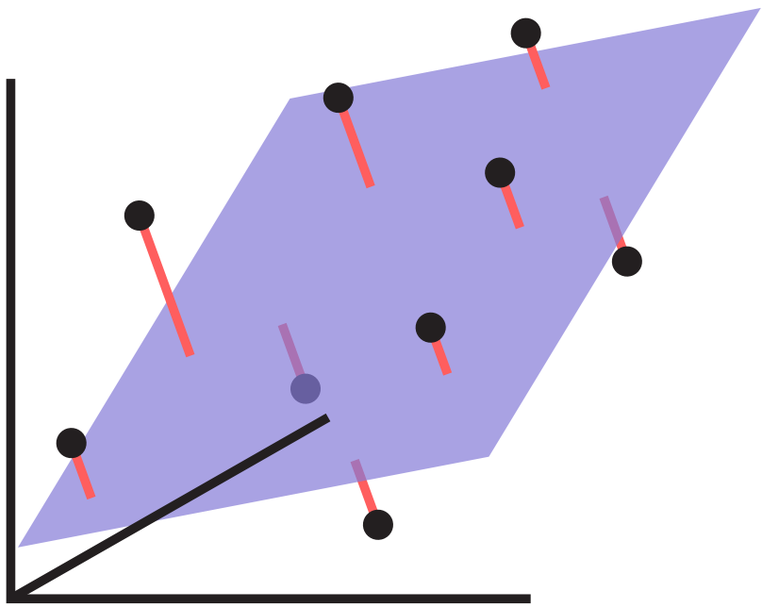
Two views of PCA* on an $m \times n$ matrix

$$\mathbf{x}_{i\cdot} \approx \mathbf{H}\mathbf{w}_{i\cdot}$$

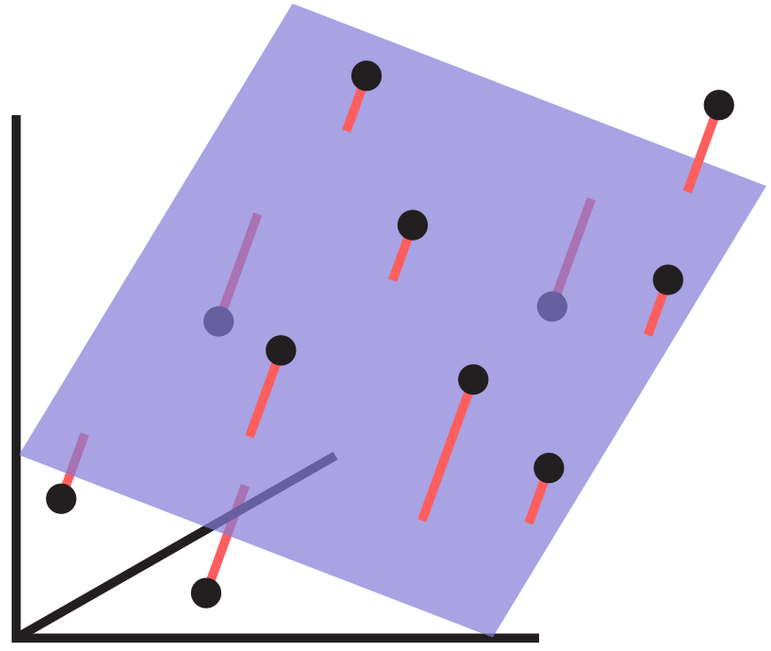


$$\mathbf{x}_{\cdot j} \approx \mathbf{W}\mathbf{h}_{\cdot j}$$





m points in n dimensions



n points in m dimensions

Rigorous connection between matrix factorization and PCA

Exercise 1: show that

$$\min_{\mathbf{W},\mathbf{H}} \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2 = \min_{\mathbf{W},\mathbf{H}} \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2 \text{ subject to } \mathbf{W}^\top\mathbf{W} = \mathbf{I}$$

Exercise 2: Show that

$$\min_{\mathbf{W},\mathbf{H}} \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2 \text{ subject to } \mathbf{W}^\top\mathbf{W} = \mathbf{I}$$

is equivalent to:

$$\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{W}\mathbf{W}^\top\mathbf{X}\|_F^2 \text{ subject to } \mathbf{W}^\top\mathbf{W} = \mathbf{I}$$

Rigorous connection between matrix factorization and PCA

Exercise 3: Show that

$$\min_{\mathbf{W}} \; \|\mathbf{X} - \mathbf{W}\mathbf{W}^\top\mathbf{X}\|_F^2 \;\text{ subject to }\; \mathbf{W}^\top\mathbf{W} = \mathbf{I}$$

is equivalent to:

$$\max_{\mathbf{W}} \; \text{Tr}\left[\mathbf{W}^\top\mathbf{X}^\top\mathbf{X}\mathbf{W}\right] \;\text{ subject to }\; \mathbf{W}^\top\mathbf{W} = \mathbf{I}$$

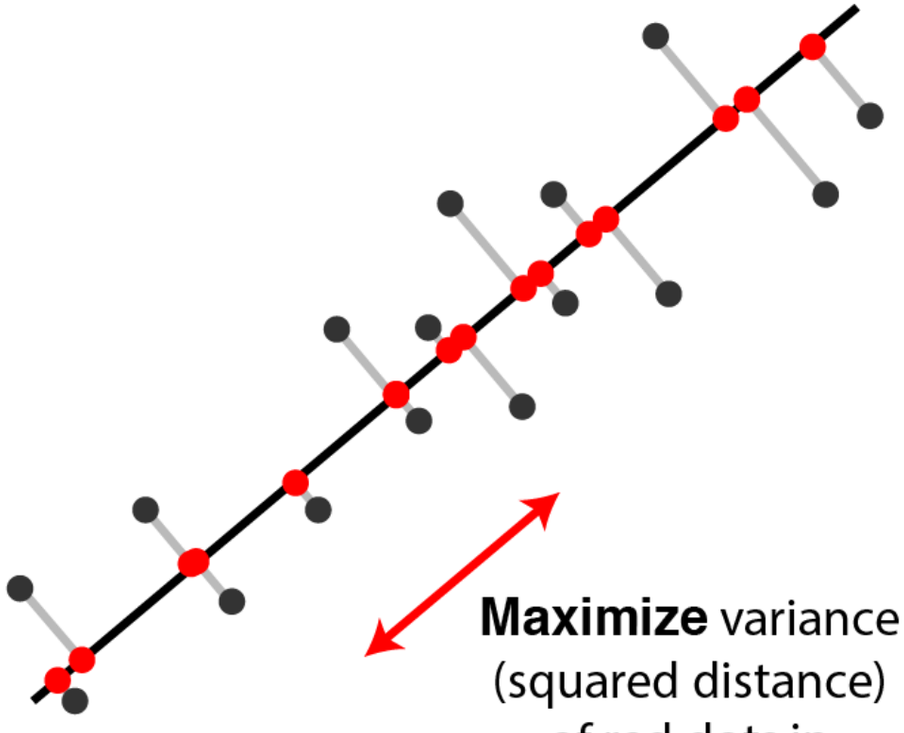
In summary: Two more views of PCA

maximize $\|\mathbf{W}\mathbf{W}^\top\mathbf{X}\|_F^2$ $\mathbf{W}$

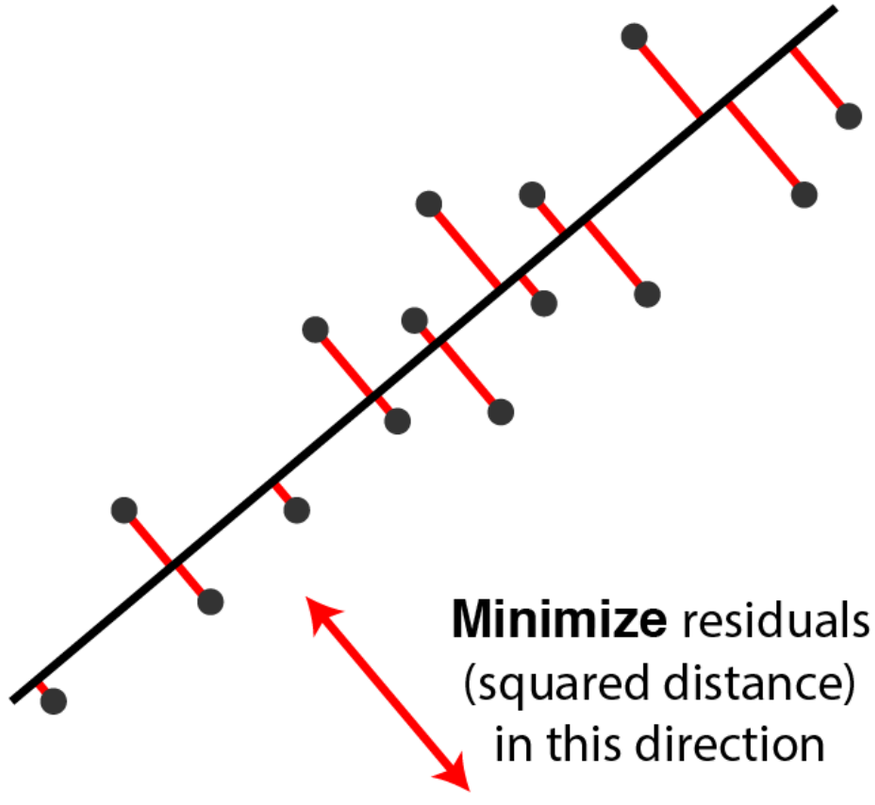
(subject to $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$*)*

minimize $\|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2$ $\mathbf{W},\mathbf{H}$

(subject to $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$*)*



Maximize variance (squared distance) of red dots in this direction



Minimize residuals (squared distance) in this direction

PCA is just one of an infinite set of solutions

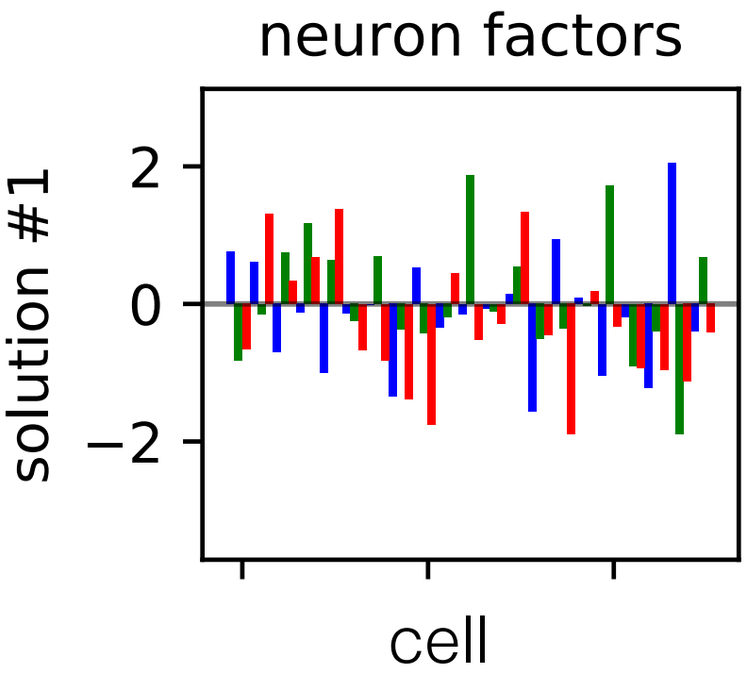
known as "the rotation problem"

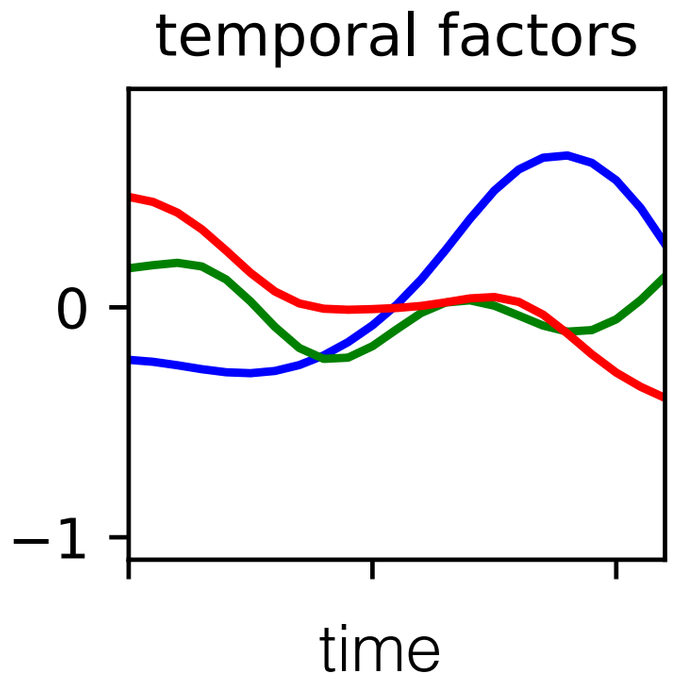
$$\widehat{\mathbf{X}} = \mathbf{WH} = \mathbf{WFF}^{-1}\mathbf{H} = \overline{\mathbf{W}}\,\overline{\mathbf{H}}$$

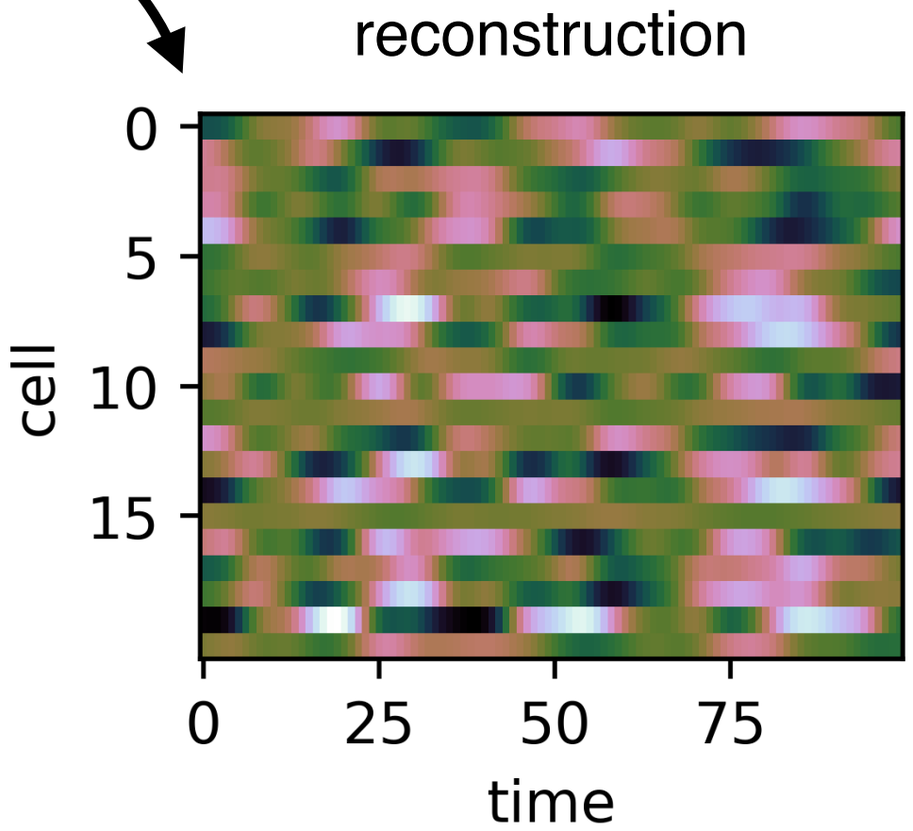
PCA is just one of an infinite set of solutions

known as "the rotation problem"

$$\widehat{\mathbf{X}} = \mathbf{WH} = \mathbf{WFF}^{-1}\mathbf{H} = \overline{\mathbf{W}}\,\overline{\mathbf{H}}$$



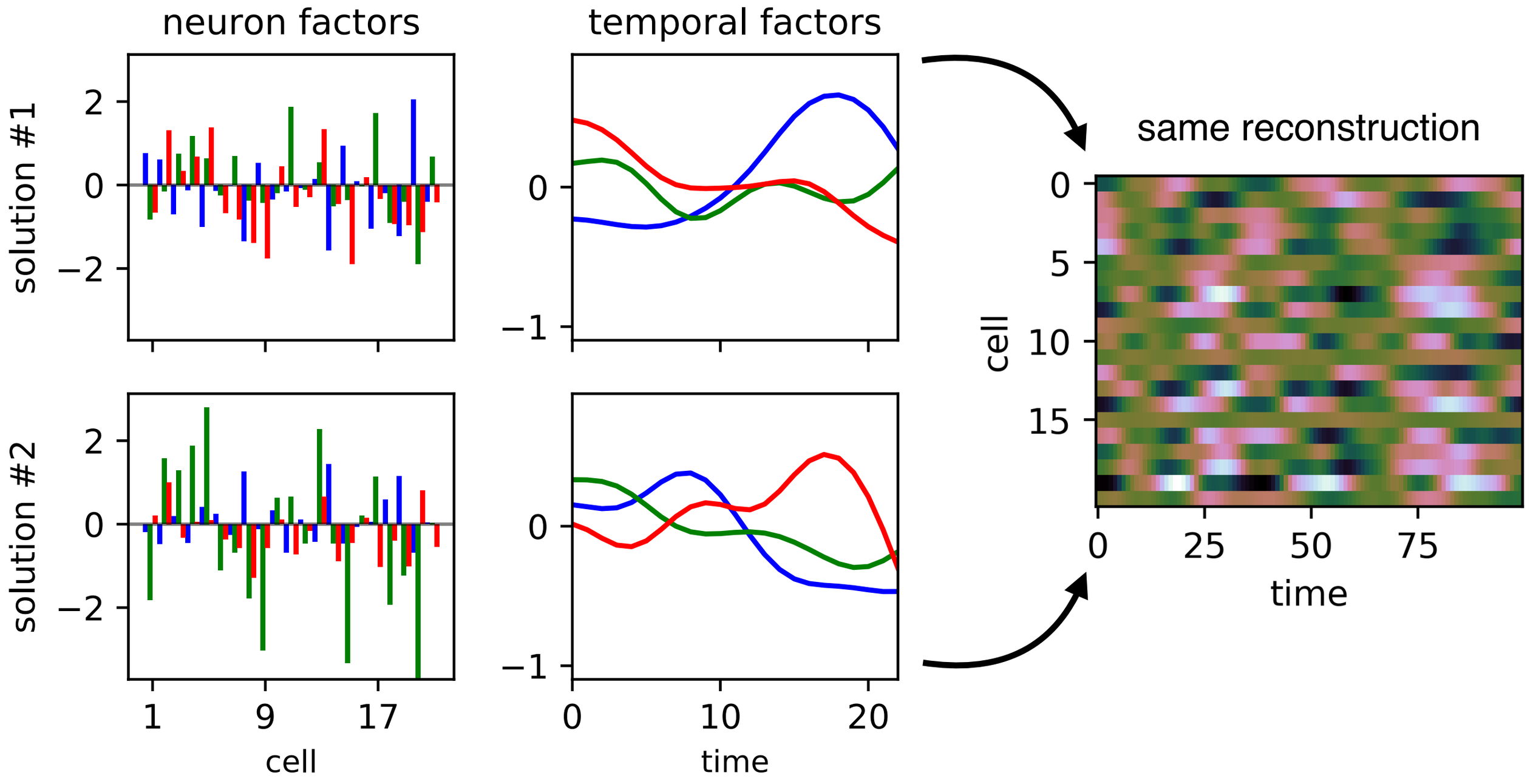




PCA is just one of an infinite set of solutions

known as "the rotation problem"

$$\widehat{\mathbf{X}} = \mathbf{WH} = \mathbf{WFF}^{-1}\mathbf{H} = \overline{\mathbf{W}}\,\overline{\mathbf{H}}$$



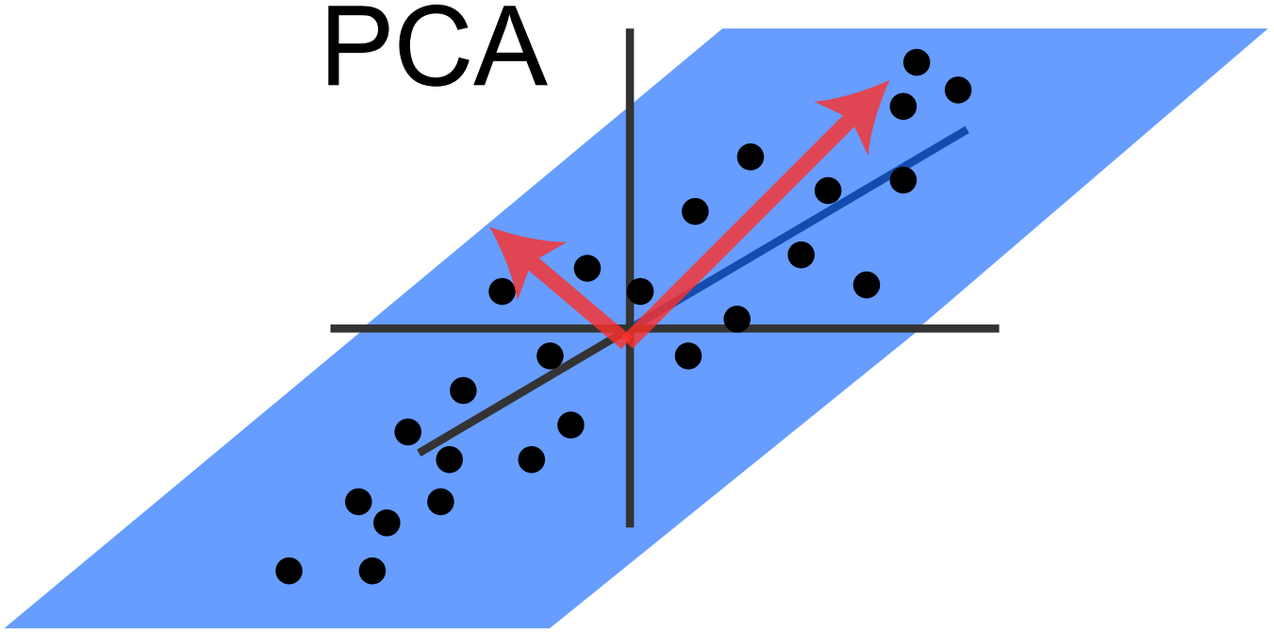
Nonnegative Matrix Factorization (NMF)

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

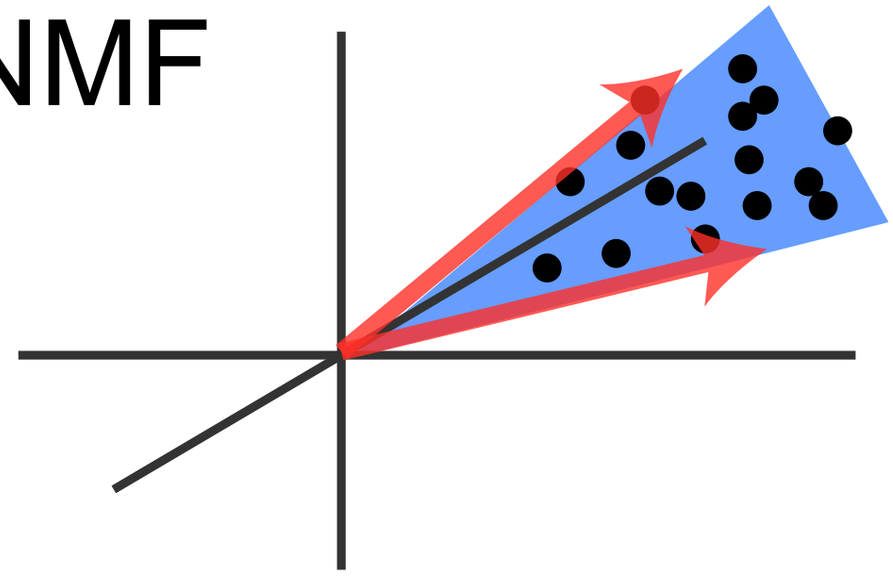
Nonnegative Matrix Factorization (NMF)

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

PCA



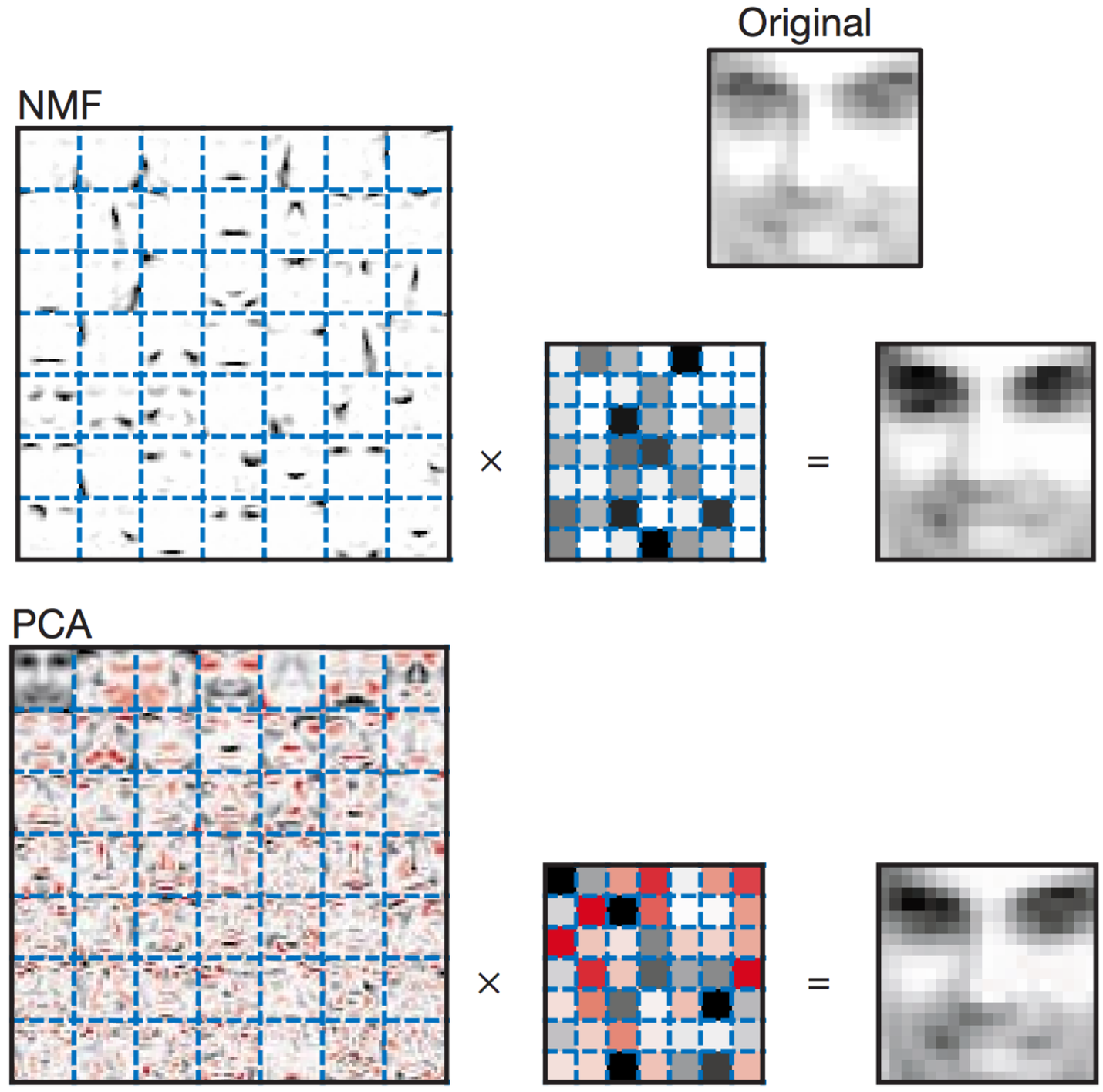
NMF



Nonnegative Matrix Factorization

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

$$\text{subject to} \quad \mathbf{U} \geq 0,\ \mathbf{V} \geq 0$$



(Lee & Seung, 1999)

Nonnegative Matrix Factorization

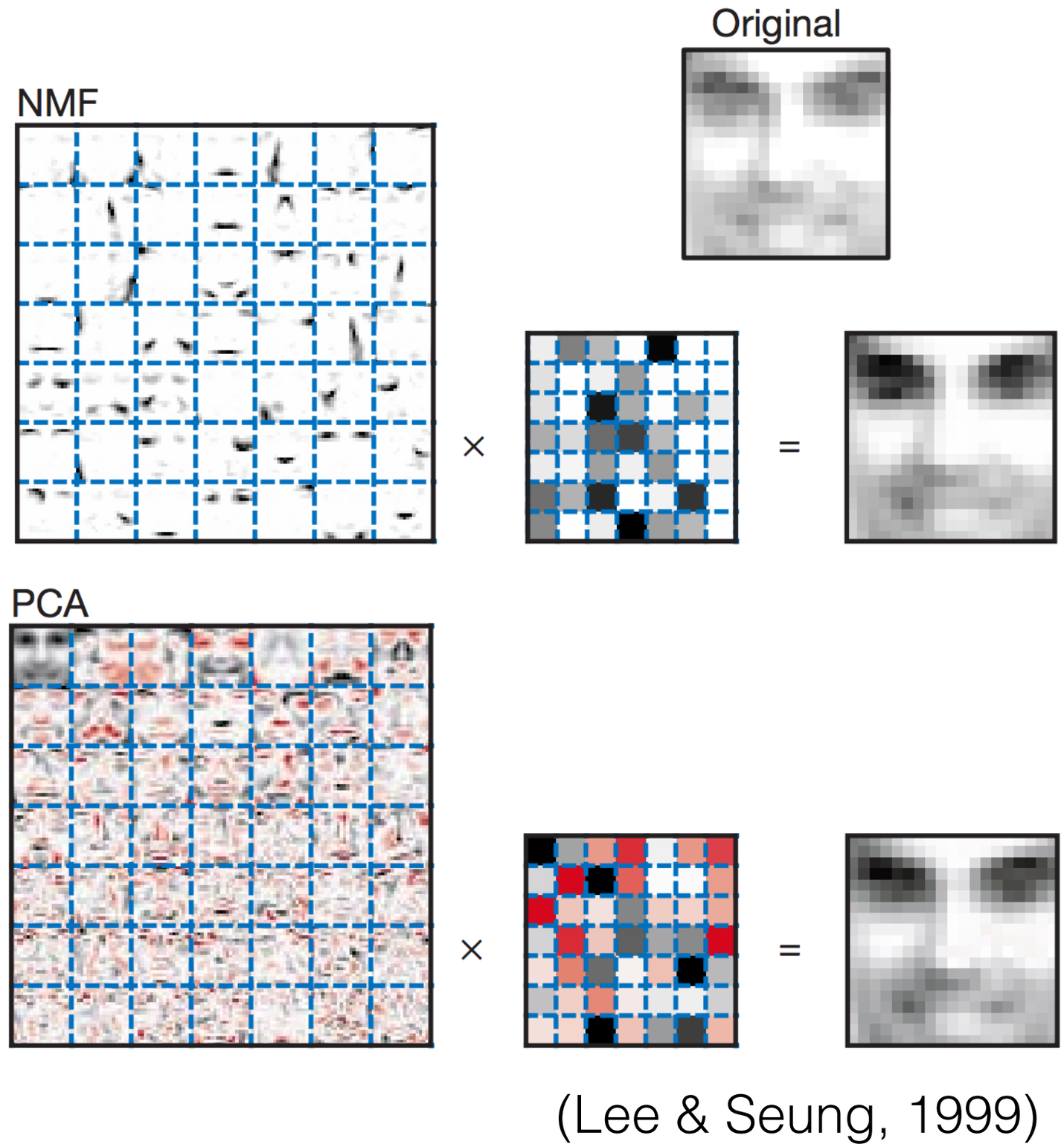
$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2$$

$$\text{subject to} \quad \mathbf{U} \geq 0,\ \mathbf{V} \geq 0$$

NMF advantages:

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

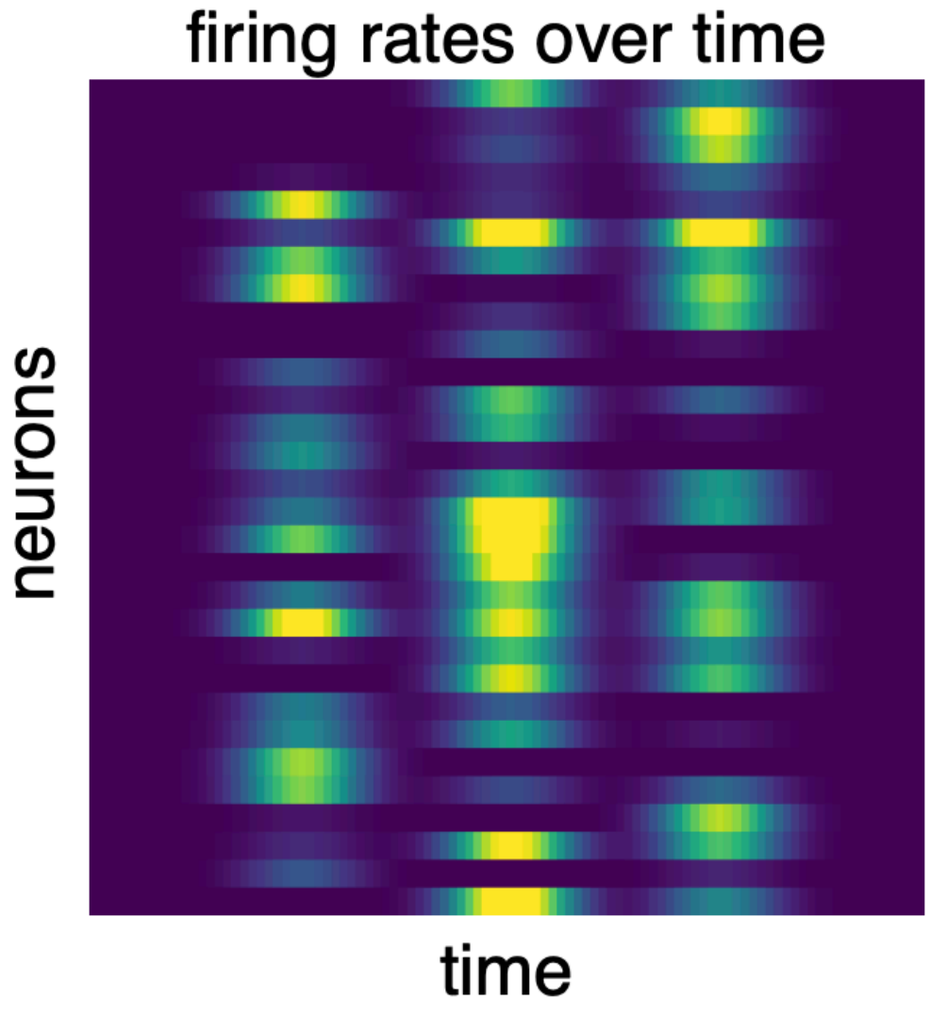
(Stodden & Donoho, 1999)

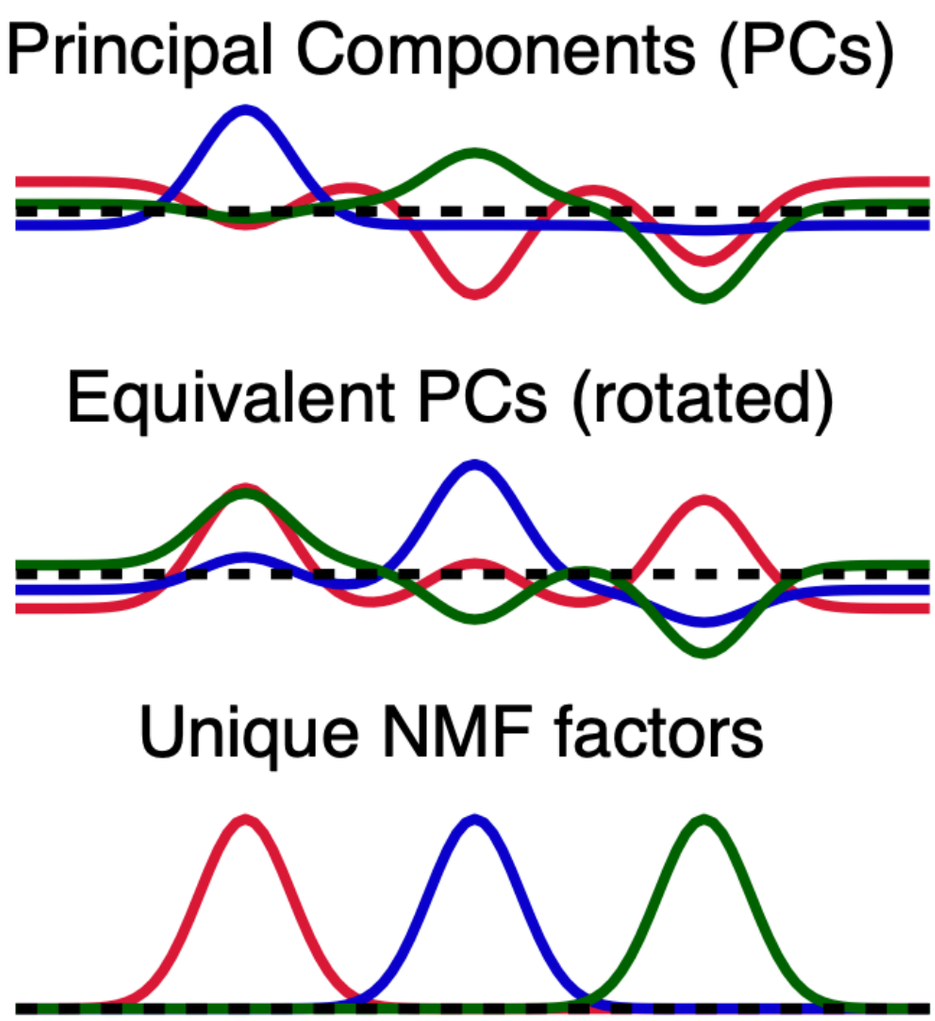


(Lee & Seung, 1999)

NMF produces unique factorizations under certain conditions!

(Stodden & Donoho, 1999; Arora et al., 2012)

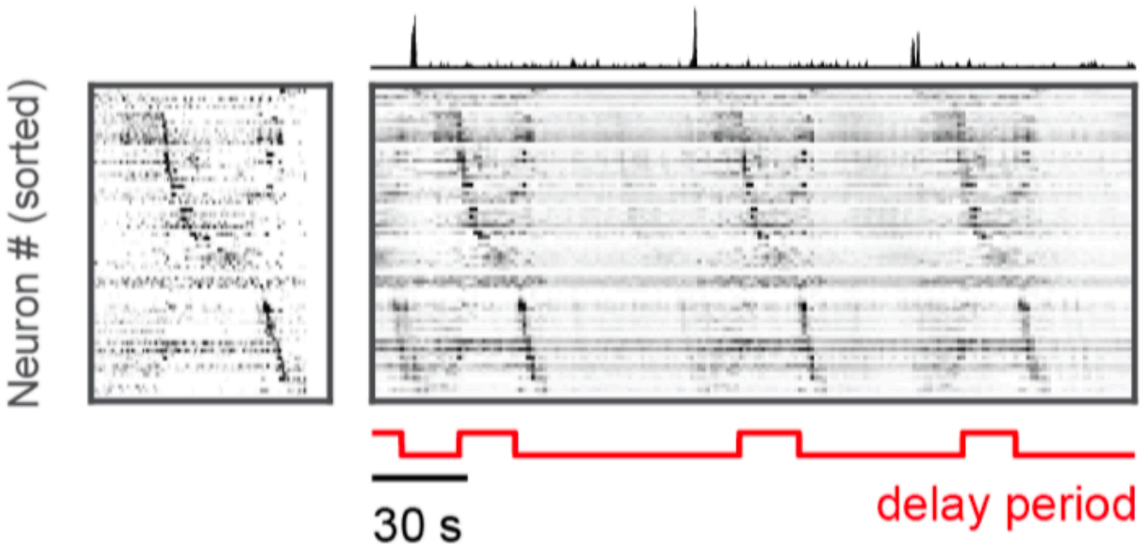




Examples of NMF in Neuroscience

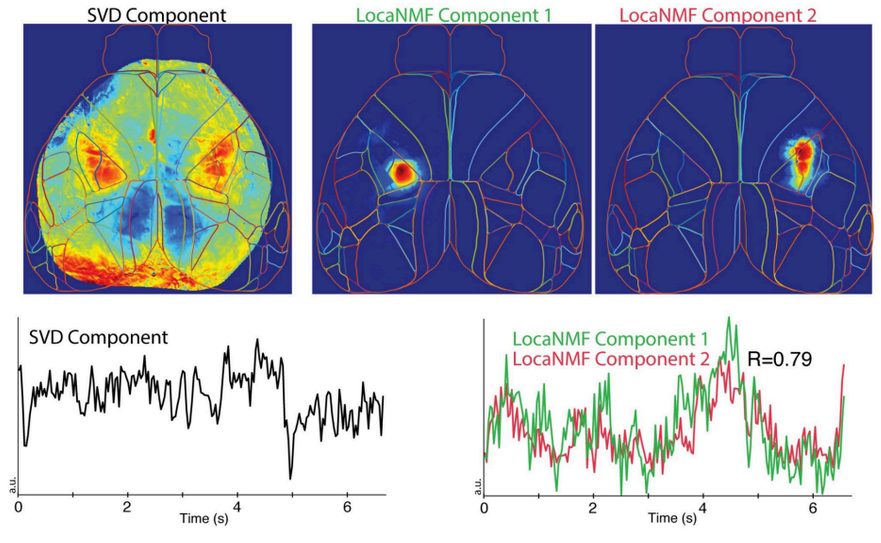
Sequence detection

Convolutional NMF / seqNMF — Mackevicius et al. (2019)



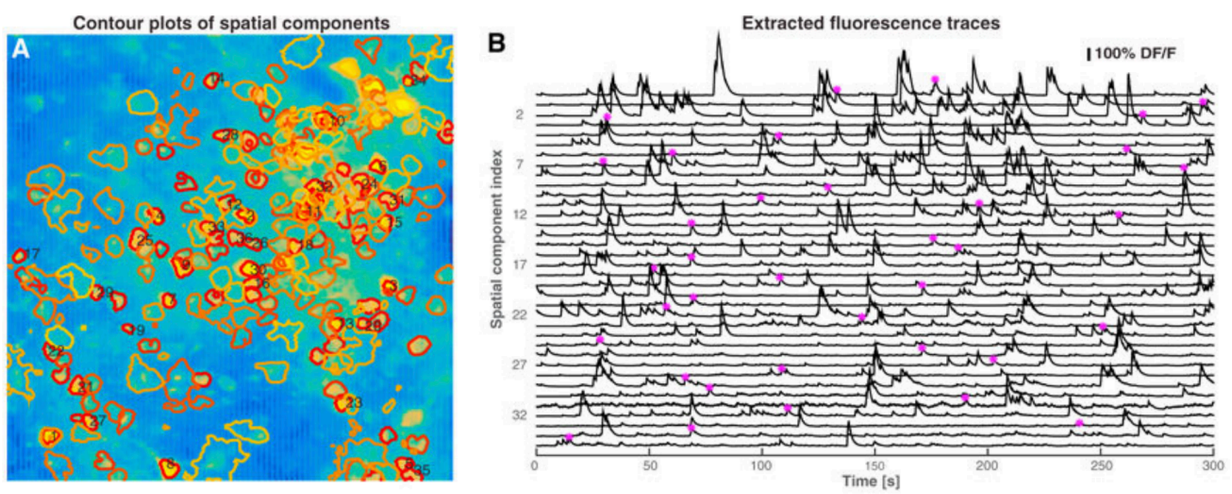
Parcellating brain regions

LocaNMF — Saxena et al. (2021)



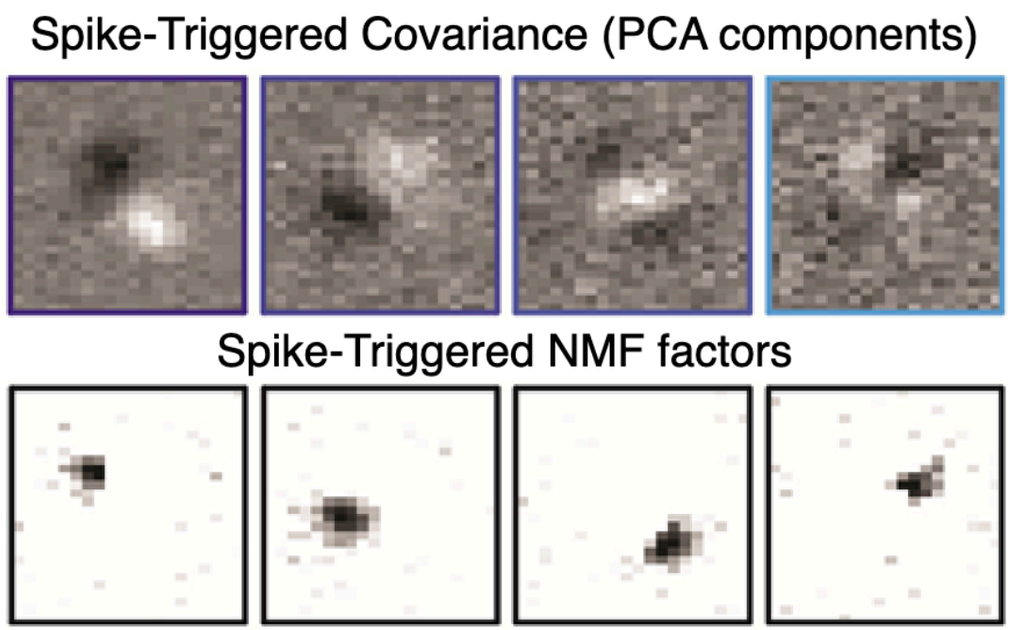
Cell extraction from Ca^{2+} imaging

C-NMF ("constrained NMF") — Pnevmatikakis et al. (2016)



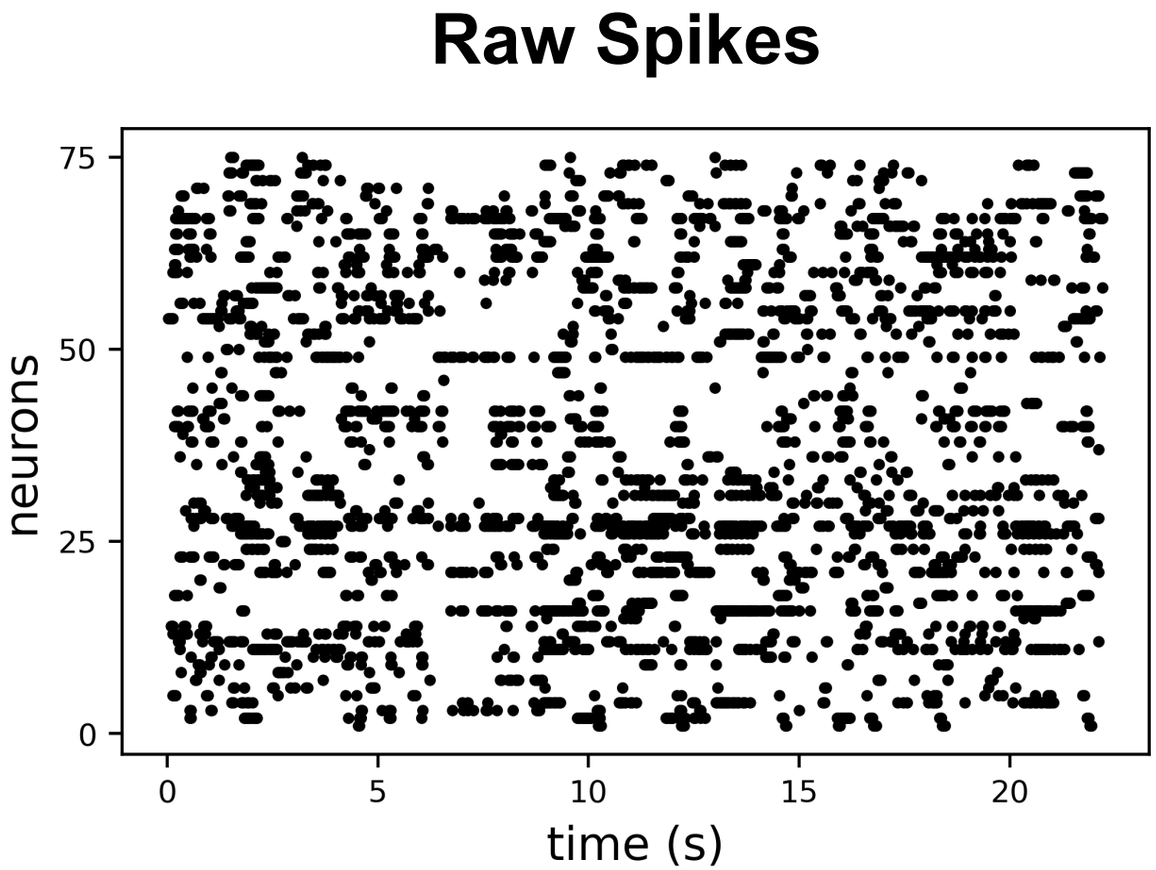
Extracting pre-synaptic subunits

Liu et al. (2017)



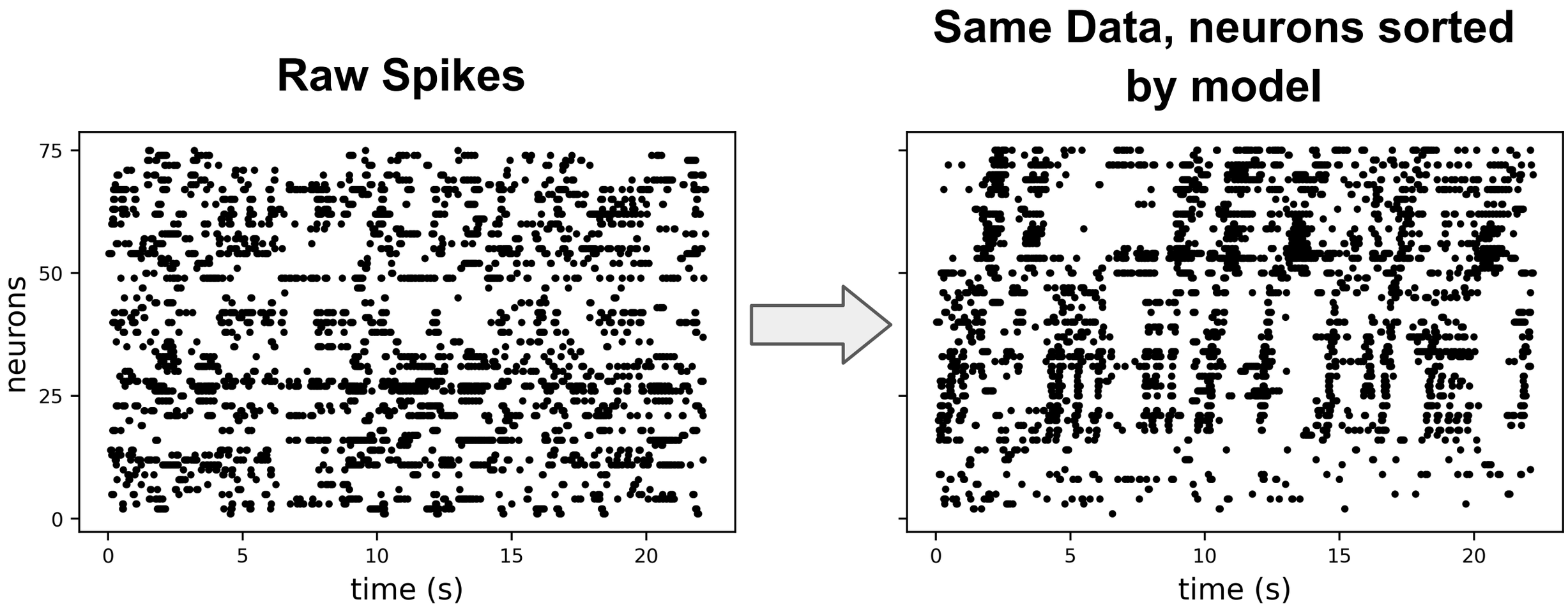
Sequence Detection by (a fancy version) of NMF

Sequence Detection by (a fancy version) of NMF



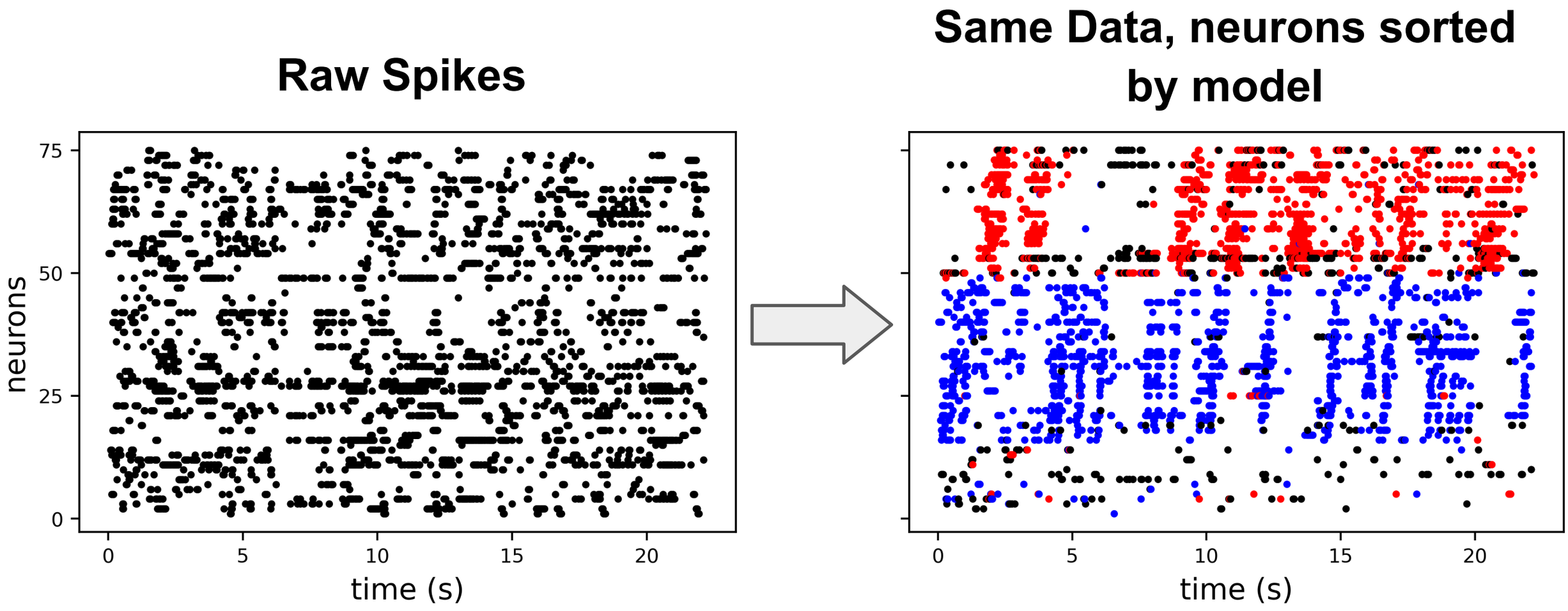
(Data collected by Emily Mackevicius)

Sequence Detection by (a fancy version) of NMF



(Data collected by Emily Mackevicius)

Sequence Detection by (a fancy version) of NMF



(Data collected by Emily Mackevicius)

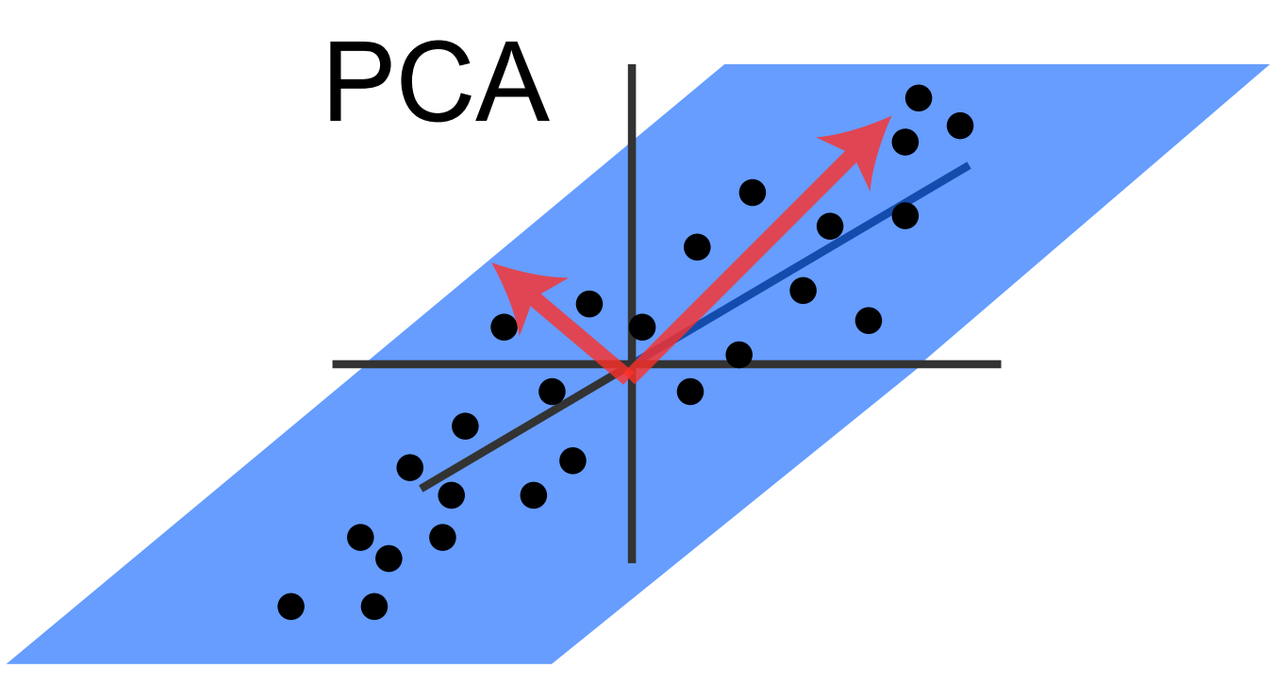
Sparse PCA*

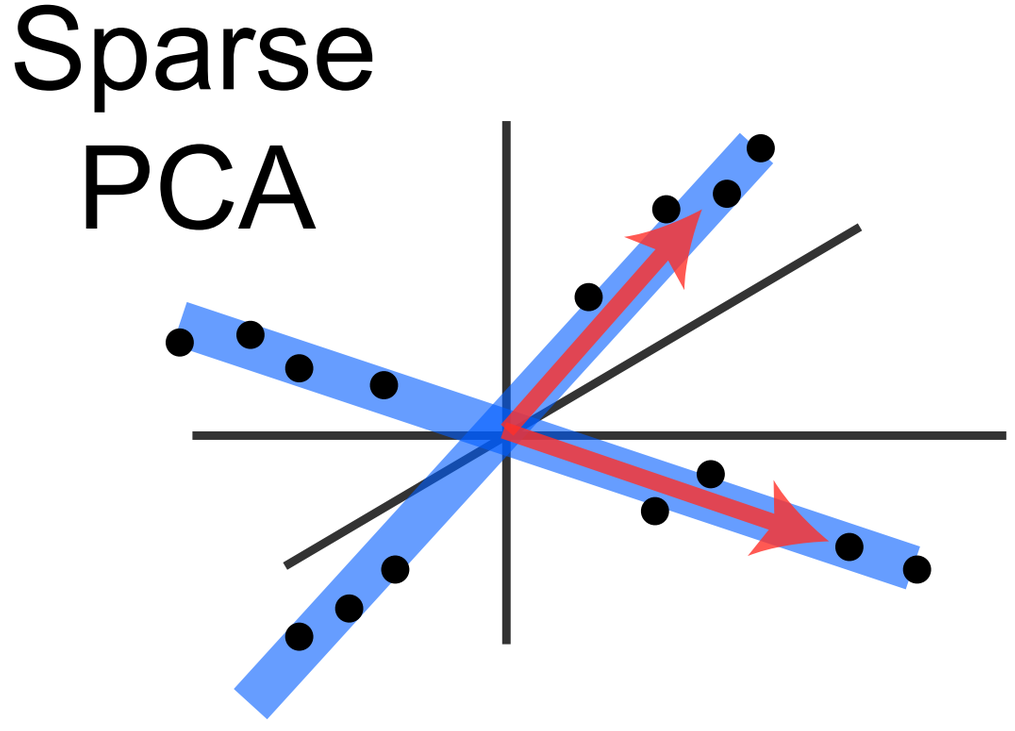
$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \lambda_u \sum_i \|\mathbf{u}_{i:}\|_1 + \lambda_v \sum_j \|\mathbf{v}_{j:}\|_2^2$$

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

Sparse PCA*

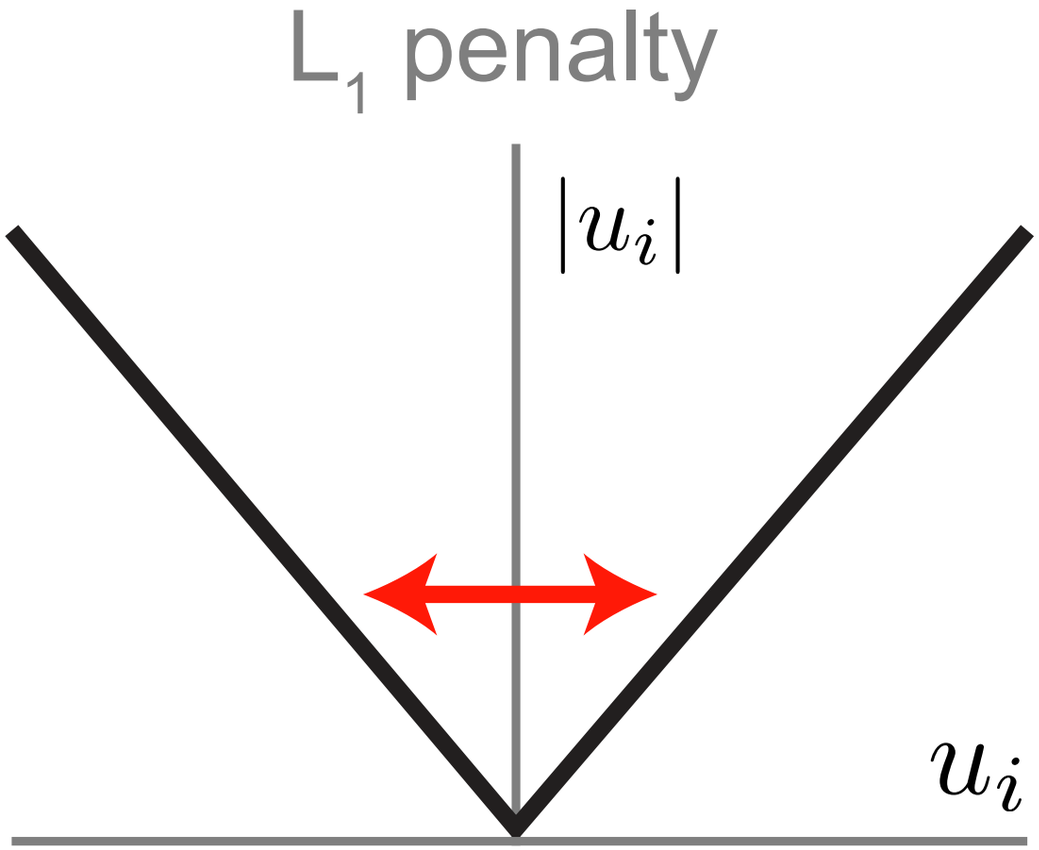
$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \lambda_u \sum_i \|\mathbf{u}_{i:}\|_1 + \lambda_v \sum_j \|\mathbf{v}_{j:}\|_2^2$$

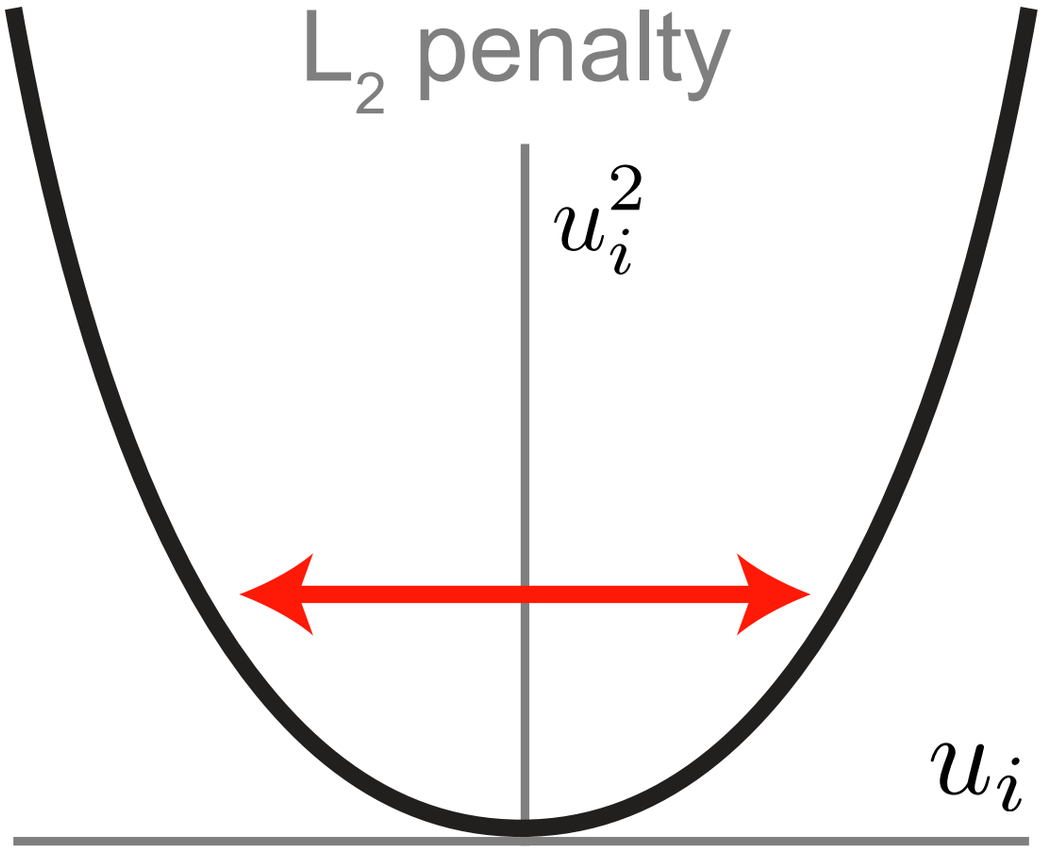




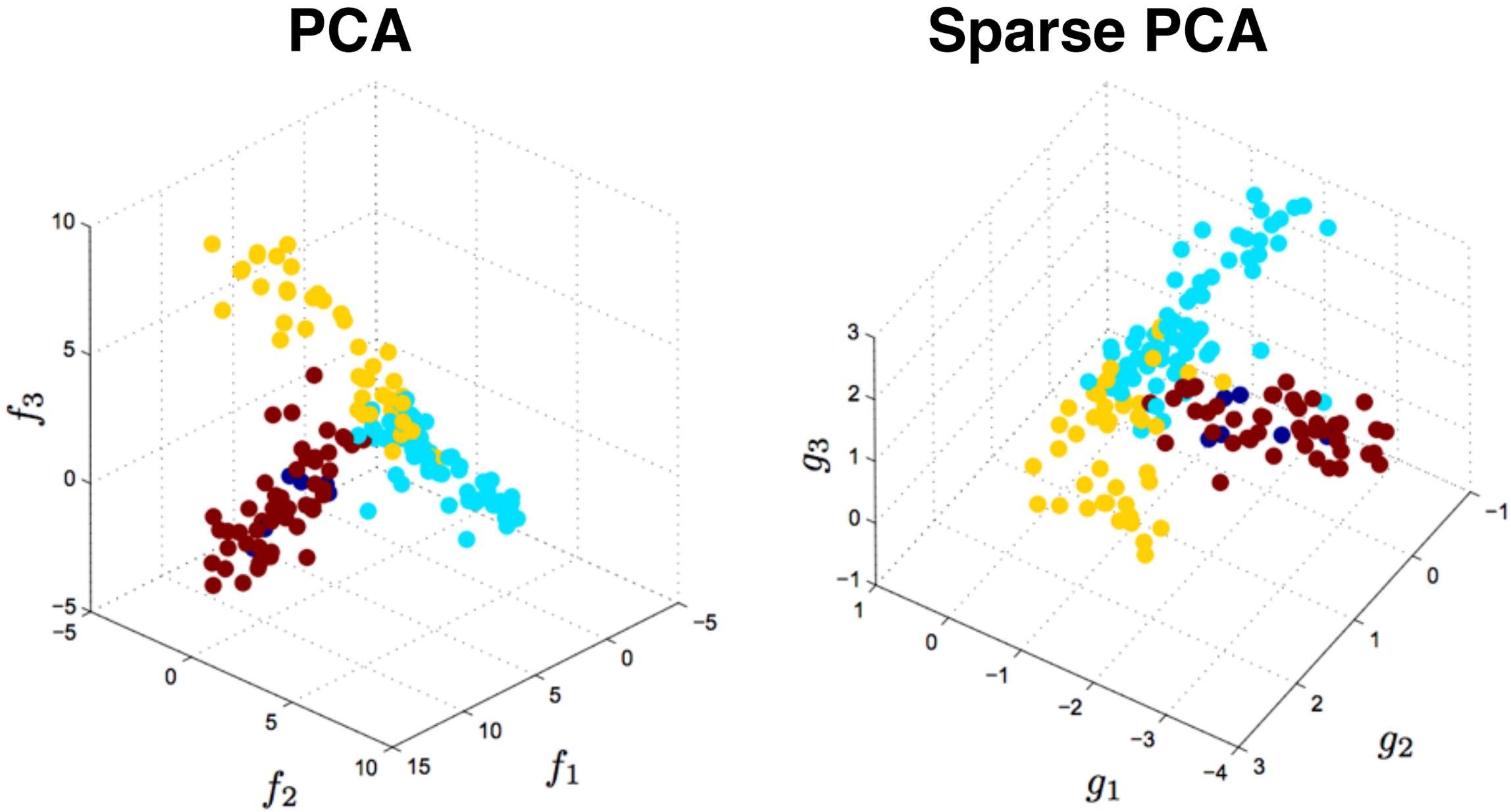
* Several variants of this model with different properties appear in the literature. Originally it was proposed by Zou et al. (2006).

Why L1 penalties result in sparse factors



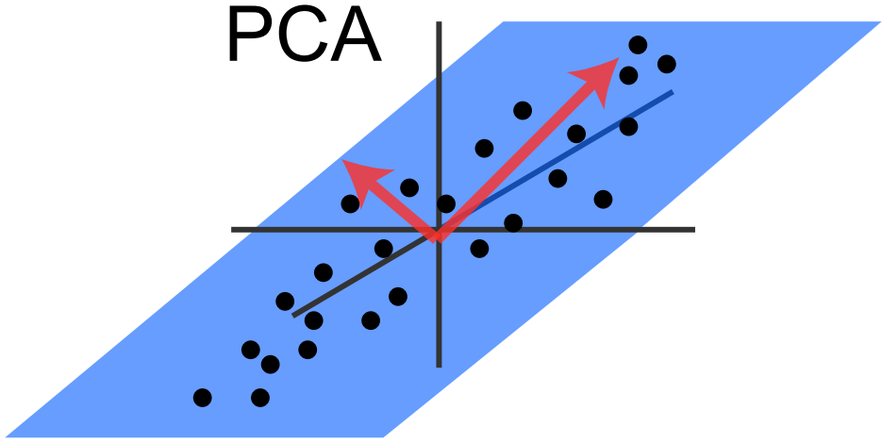


Sparse PCA



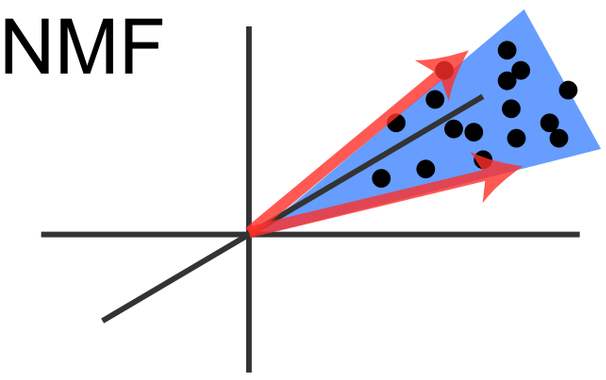
(D'Aspremont et al., 2007)

PCA



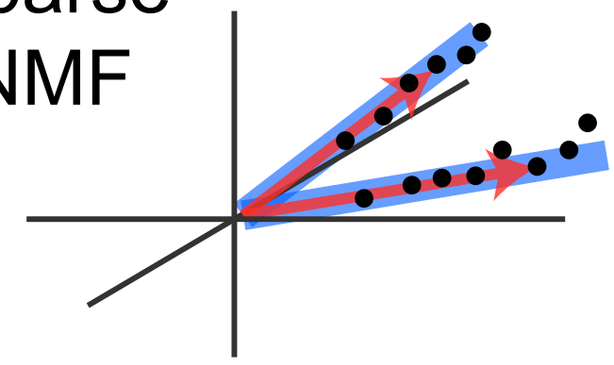
$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2$$
$$\text{subject to} \quad \mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}$$

NMF



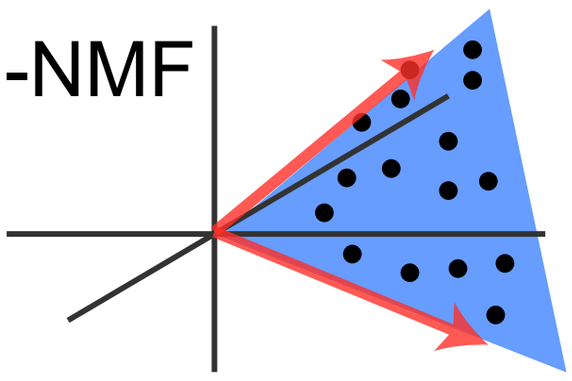
$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2$$
$$\text{subject to} \quad \mathbf{U} \geq 0,\ \mathbf{V} \geq 0$$

Sparse NMF



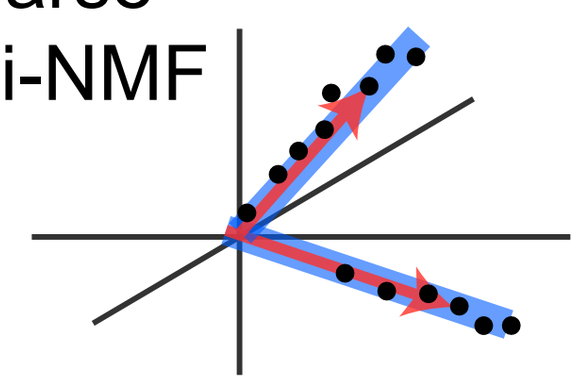
$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \lambda_u \sum_i \|\mathbf{u}_{i:}\|_1$$
$$\text{subject to} \quad \mathbf{U} \geq 0,\ \mathbf{V} \geq 0$$

Semi-NMF



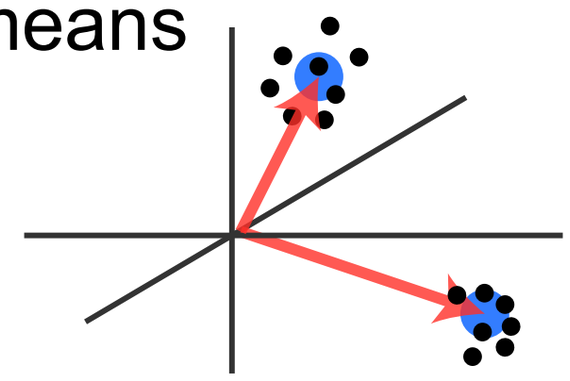
$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2$$
$$\text{subject to} \quad \mathbf{U} \geq 0$$

Sparse semi-NMF



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

K-means



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

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:

posterior *likelihood* *prior*

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

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

Matrix decomposition can be interpreted probabilistically, via Bayes Rule:

posterior *likelihood* *prior*

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

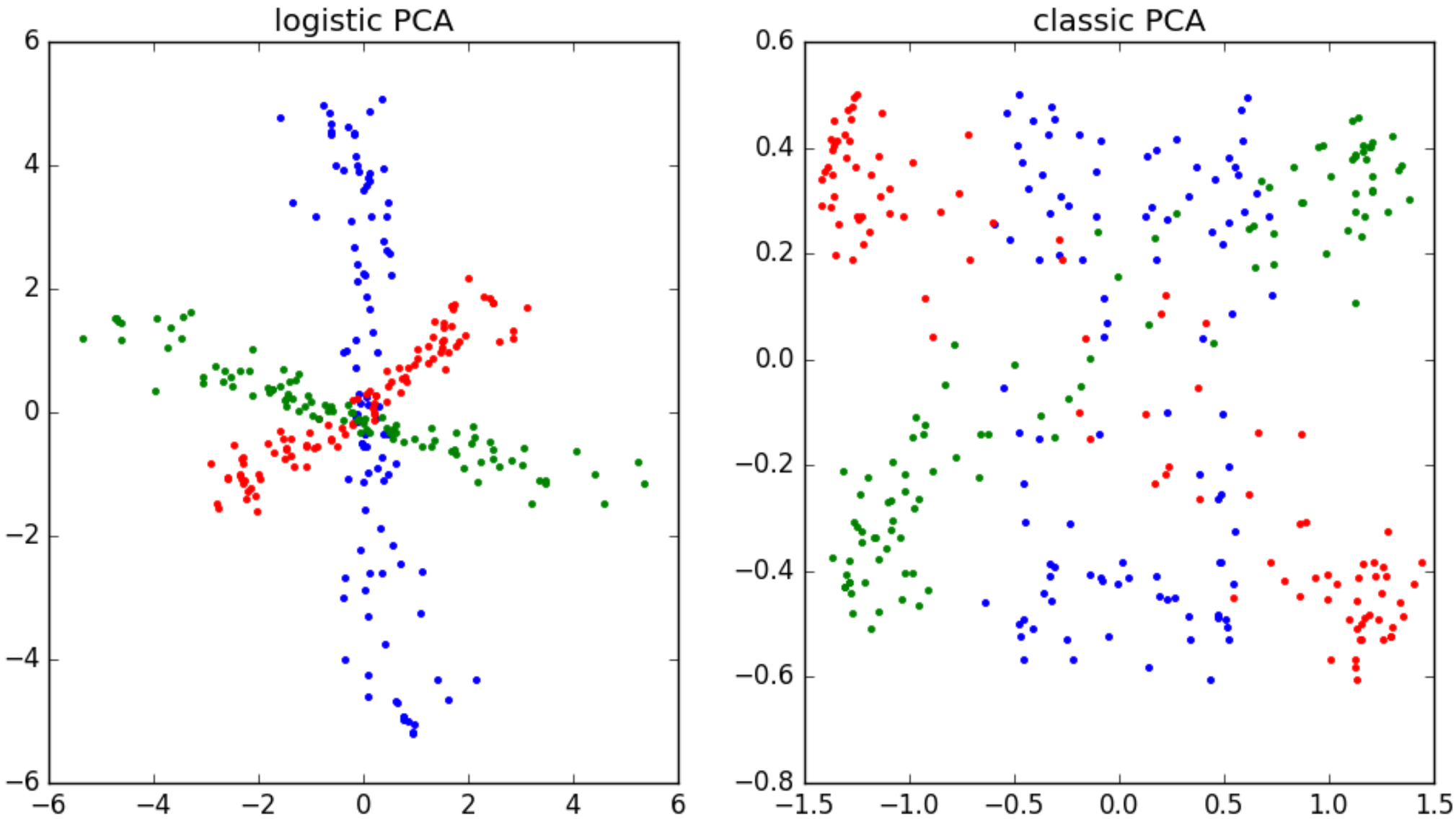
$$-\ln p(\text{model} \mid \text{data}) \propto -\ln p(\text{data} \mid \text{model}) - \ln p(\text{model})$$

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)

Combinatorial menu of models

loss functions

quadratic
(real data)

(integer data)

circular
(angular data)

regularizers/constraints

L2 norm

Derivative penalties
(smooth factors)

Further Reading:

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

Talk Outline

1. Long list of matrix decomposition models

~~**2. Optimization and model fitting**~~

3. Visualization and model assessment

4. Tensor decomposition

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

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

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
```

NMF can also be solved by alternating minimization

Each step is ***nonnegative least squares*** problem

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

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

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)

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

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

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)
```

Talk Outline

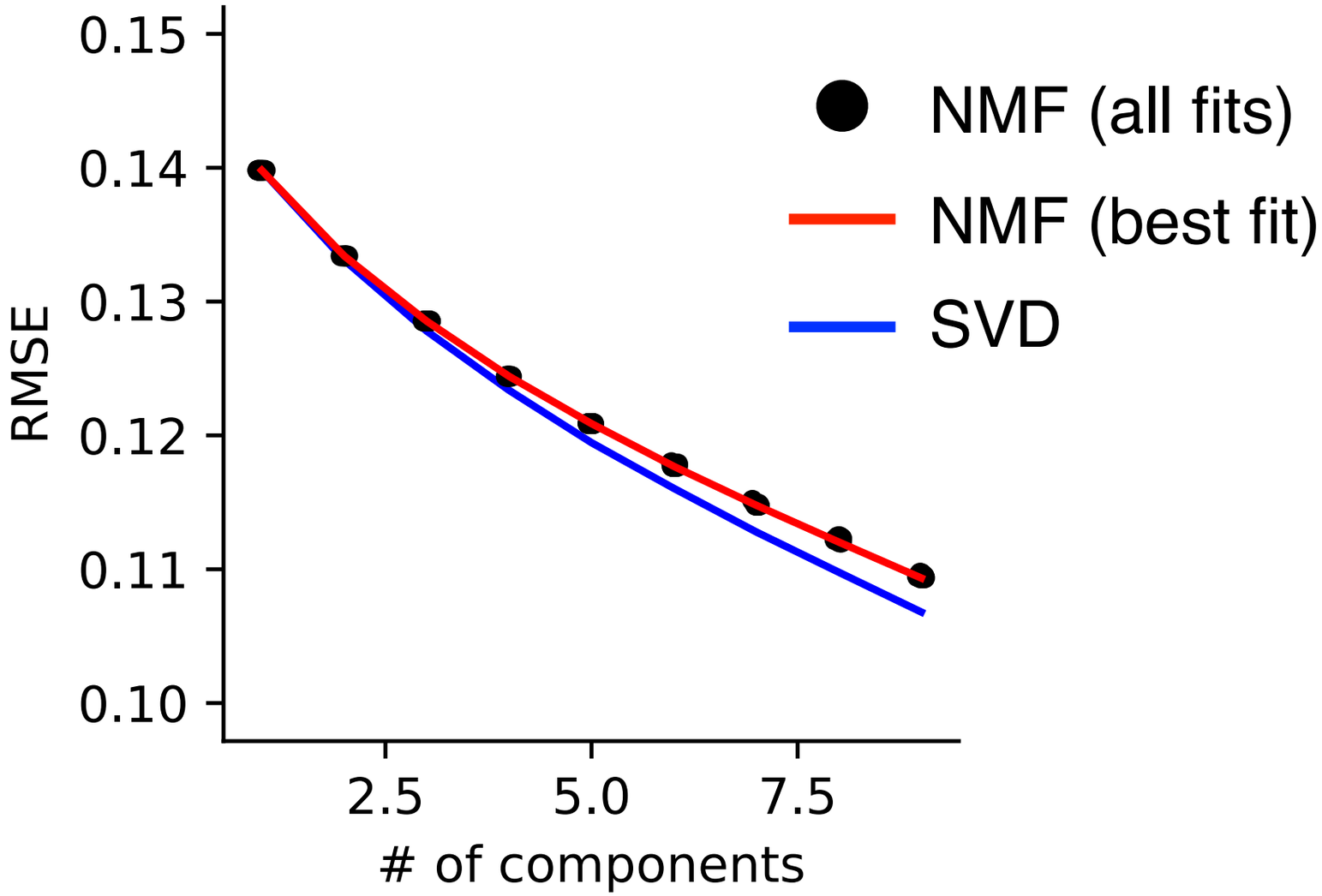
1. Long list of matrix decomposition models

~~2. Optimization and model fitting~~

3. Visualization and model assessment

4. Tensor decomposition

Error Plot — How well am I fitting the data?



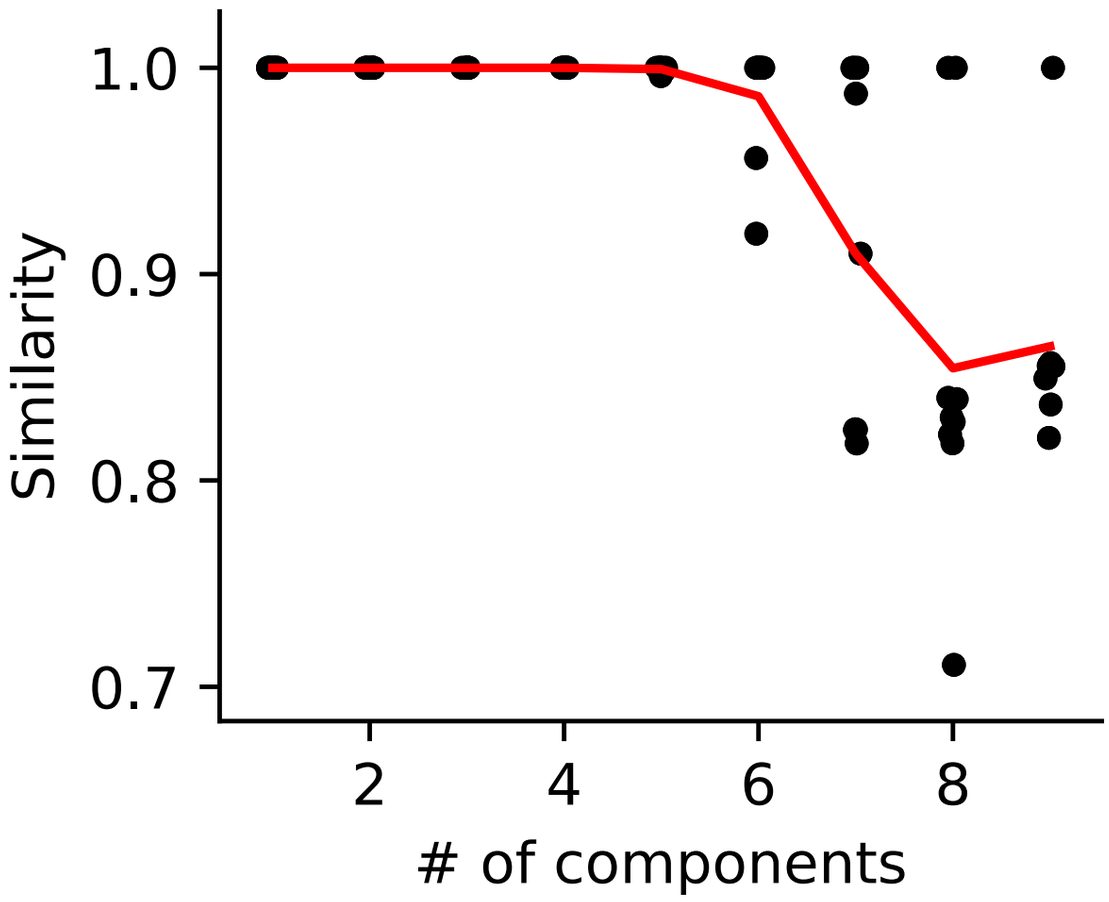
Interpretation: NMF converges to similar error from different initializations, and nearly achieves the optimal lower bound on performance set by SVD.

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

Define the similarity of two factor matrices as:

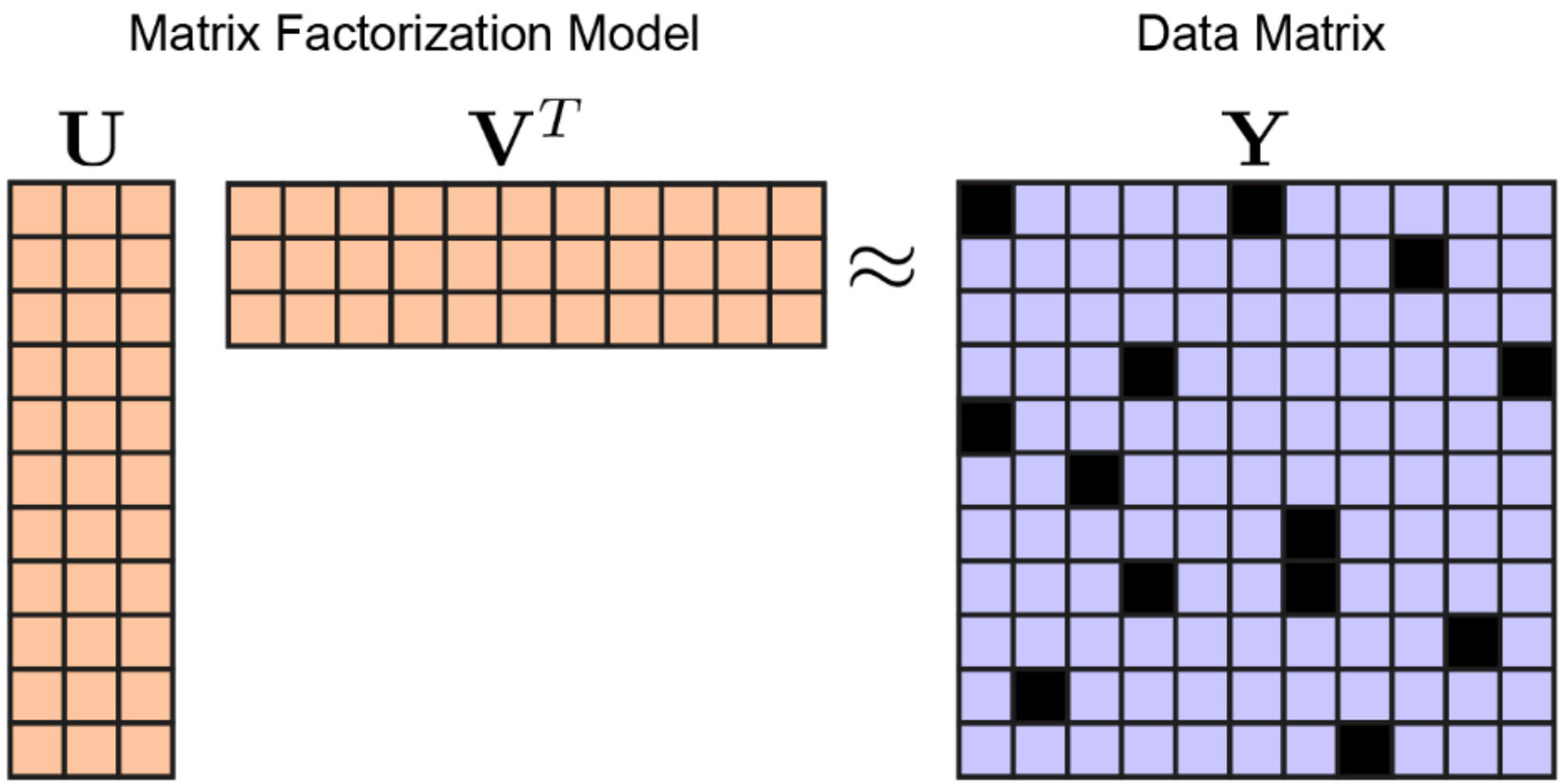
$$S(\mathbf{U}, \mathbf{U}') = \max_{\Pi} \frac{1}{r} \mathrm{Tr}\left[\mathbf{U}^T \mathbf{U}' \Pi\right]$$

where Π, is an r x r permutation matrix.



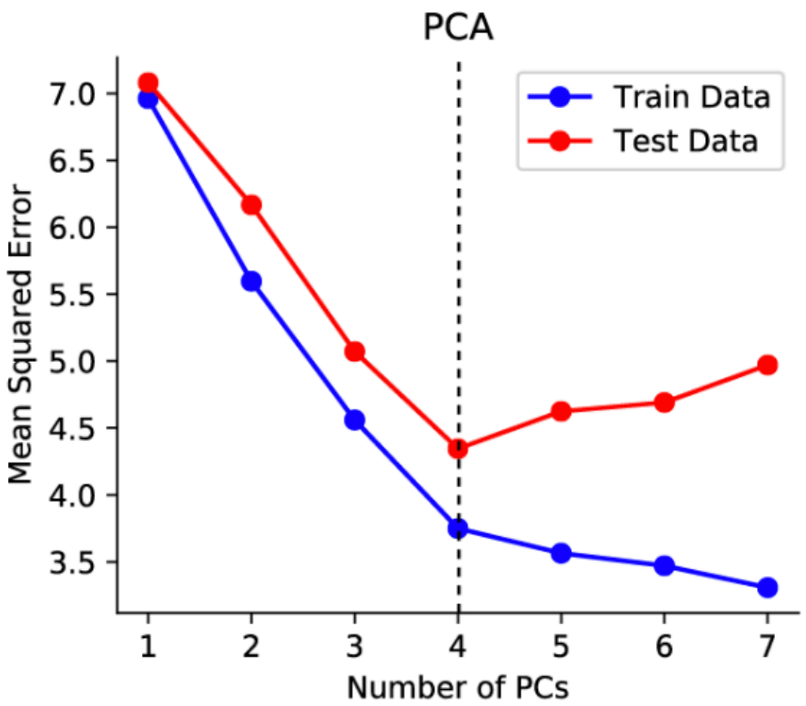
Wu et al. (2016). “Stability NMF” *PNAS*
https://www.pnas.org/doi/10.1073/pnas.1521171113

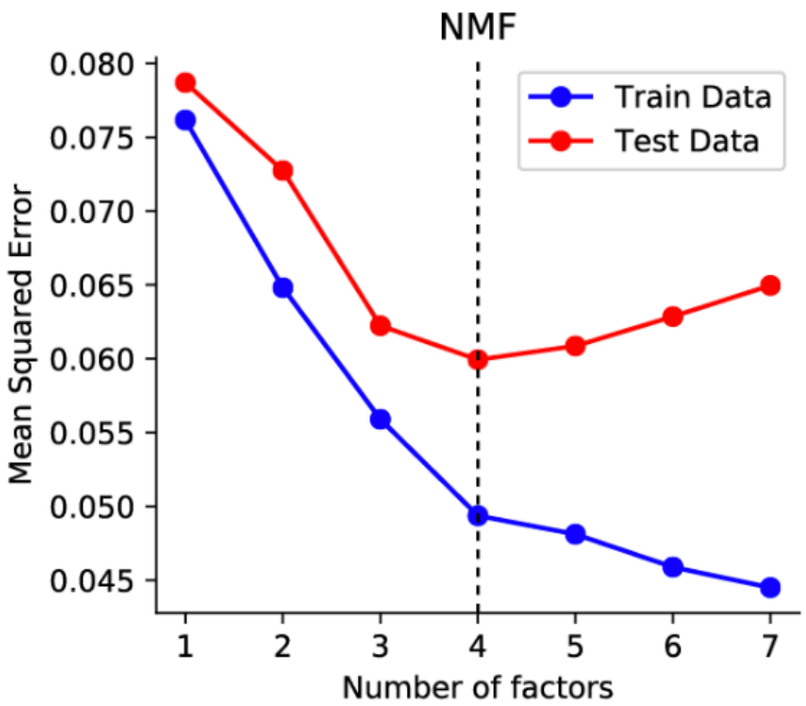
Cross-Validation

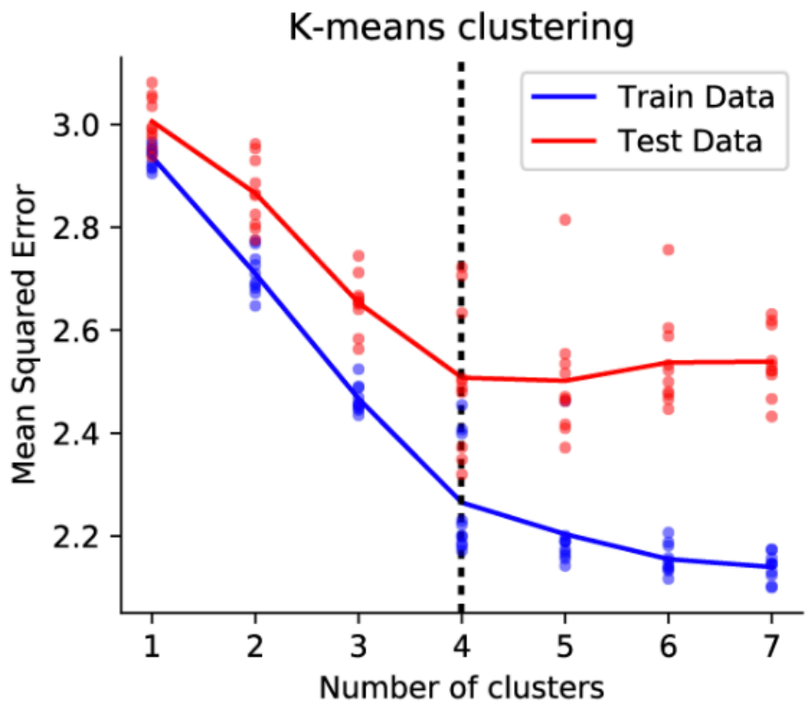


https://alexhwilliams.info/itsneuronalblog/2018/02/26/crossval/

Cross-Validation







https://alexhwilliams.info/itsneuronalblog/2018/02/26/crossval/

Talk Outline

1. Long list of matrix decomposition models

~~2. Optimization and model fitting~~

3. Visualization and model assessment

4. Tensor decomposition

<go to tensor slides>

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.

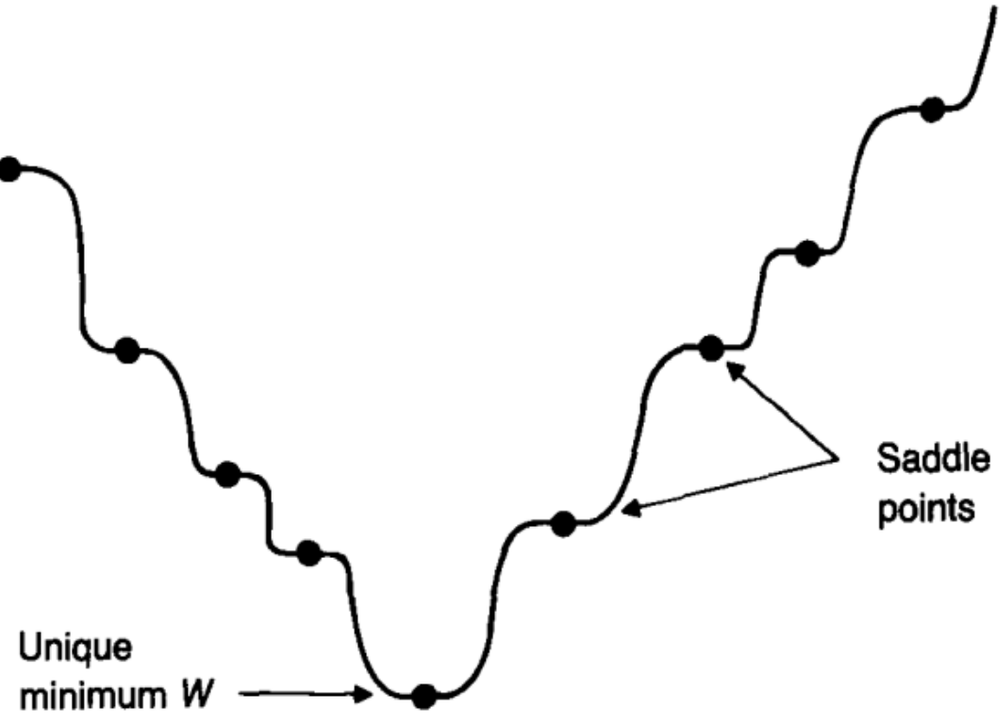
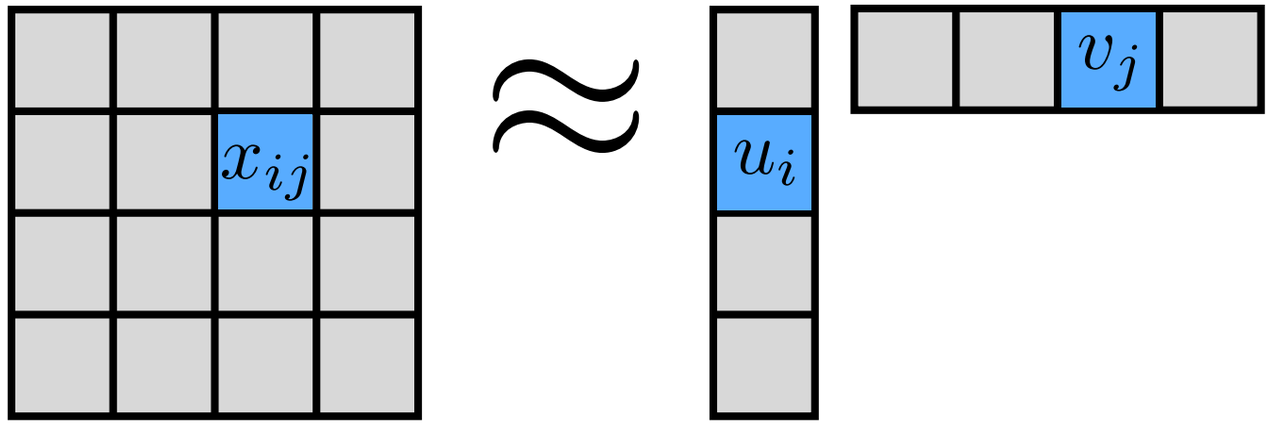


FIGURE 2. The landscape of *E*.

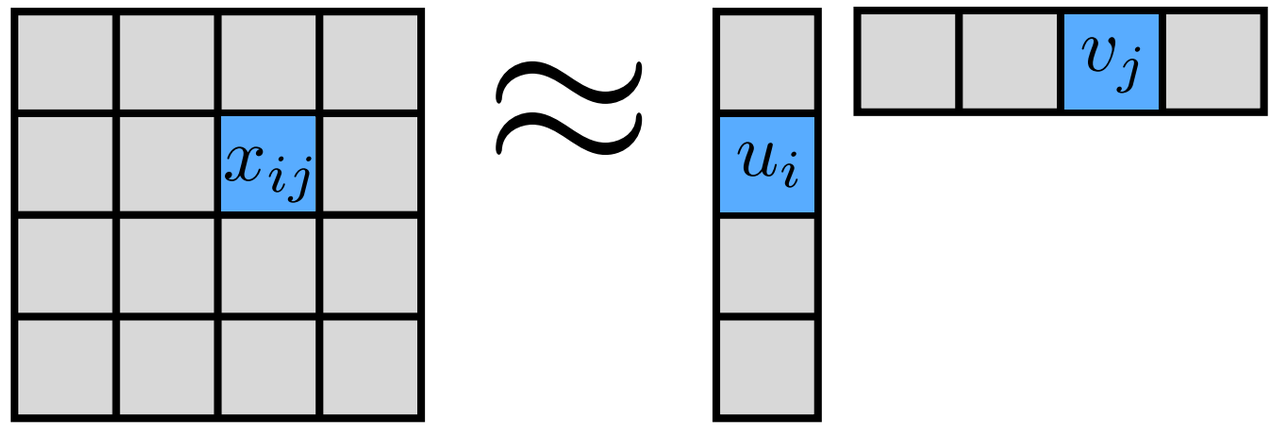
(Baldi & Hornik, 1989).

* 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

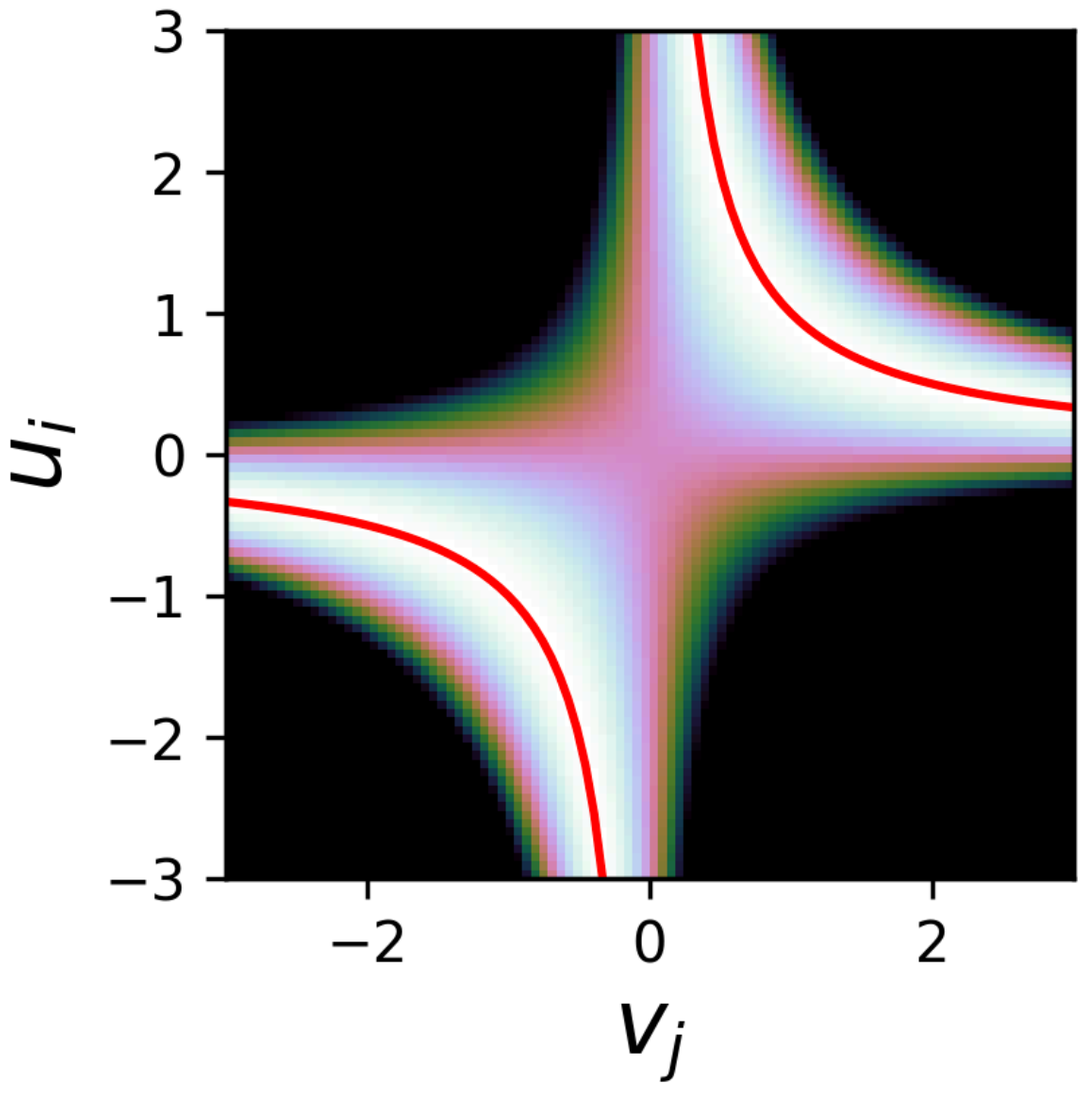
Consider the PCA loss for a single matrix element



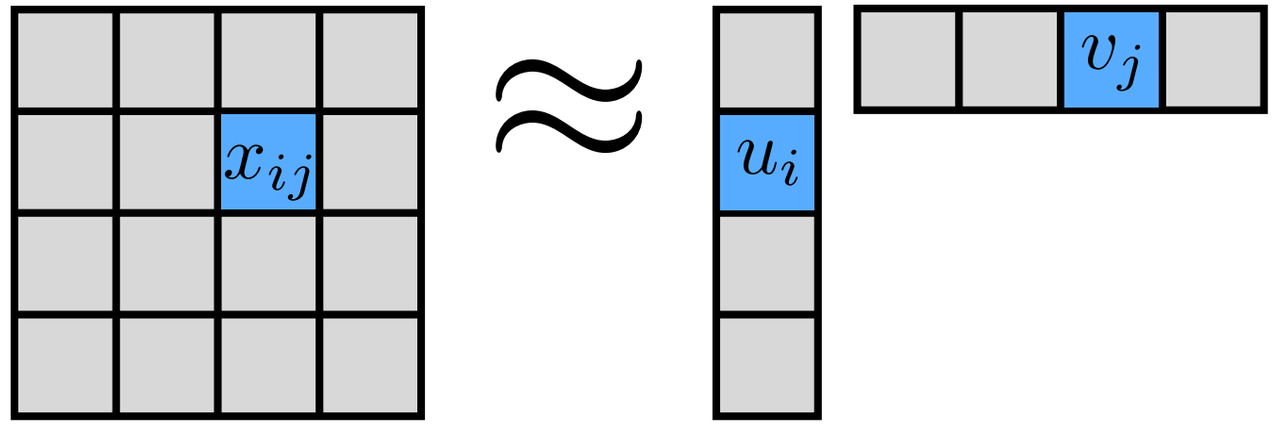
Consider the PCA loss for a single matrix element



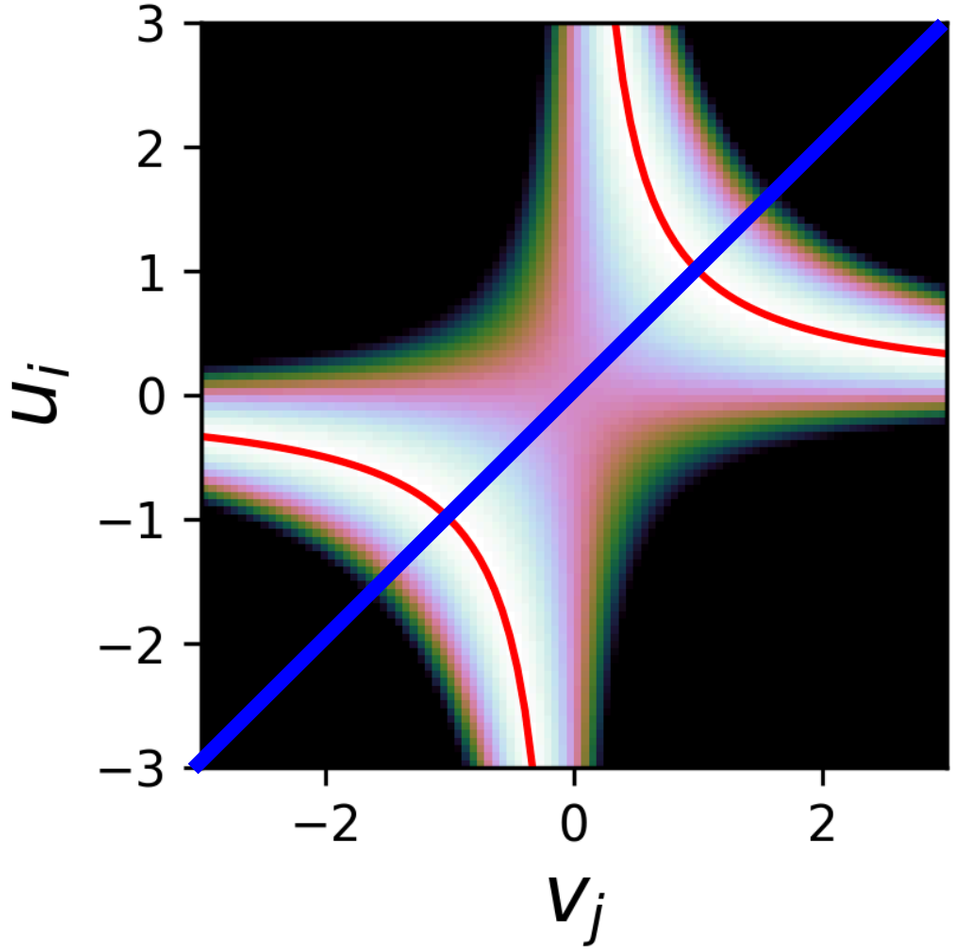
$$\ell_{ij}(u_i, v_j) = (x_{ij} - u_i v_j)^2$$

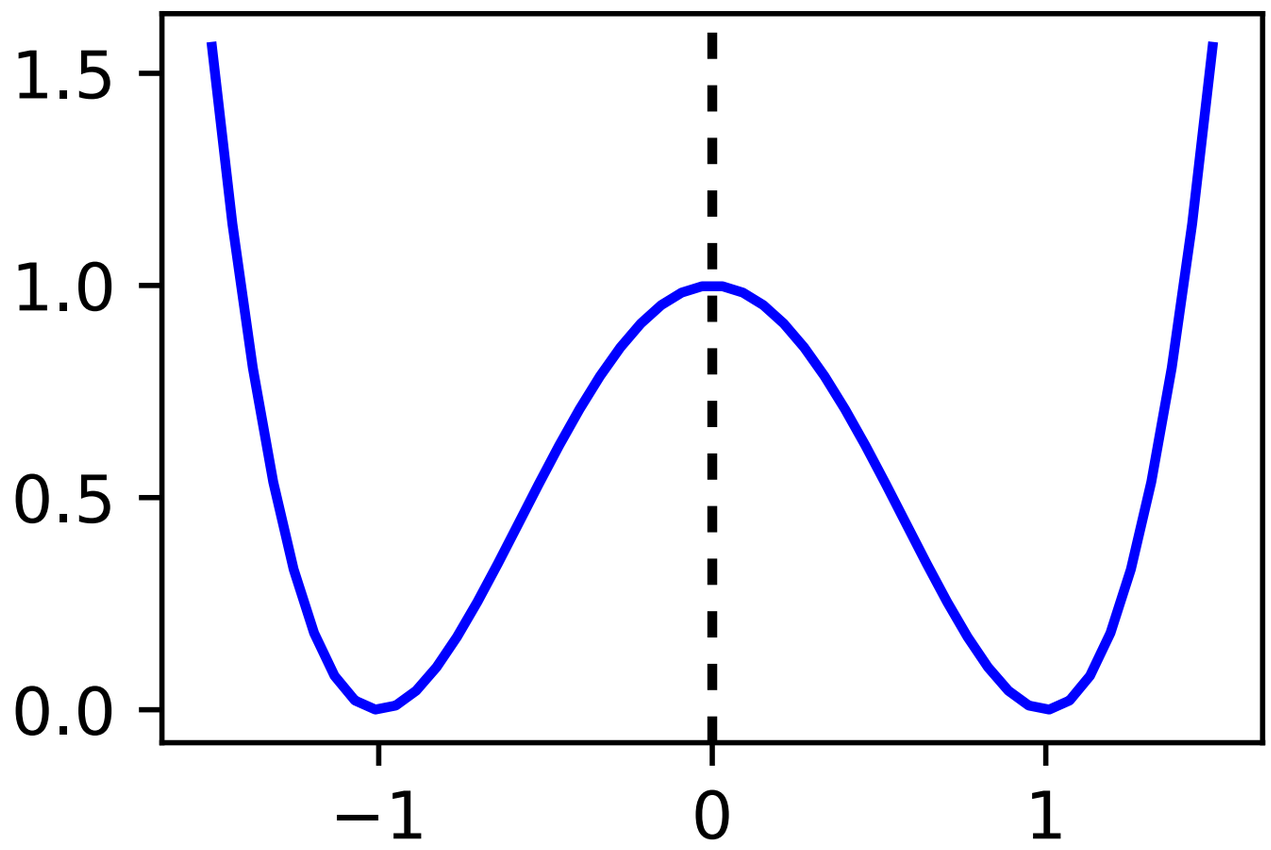


Consider the PCA loss for a single matrix element

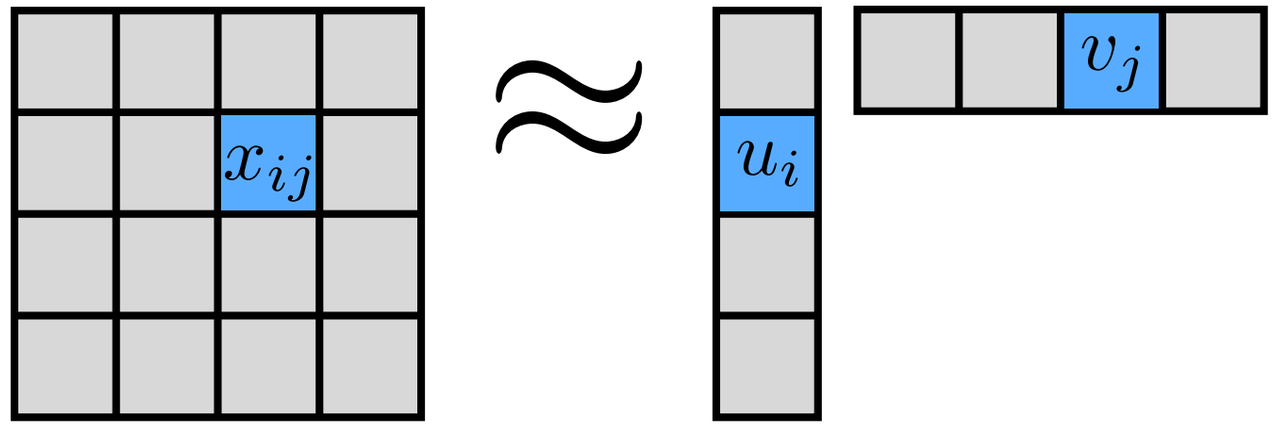


$$\ell_{ij}(u_i, v_j) = (x_{ij} - u_i v_j)^2$$



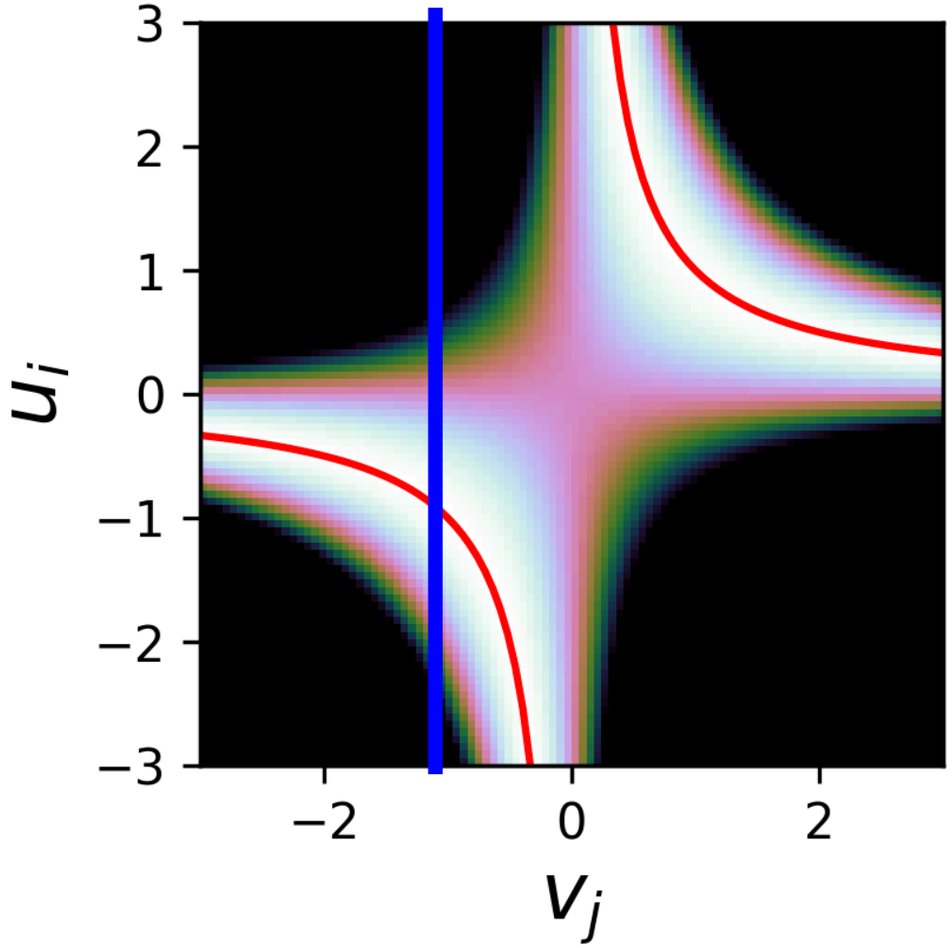


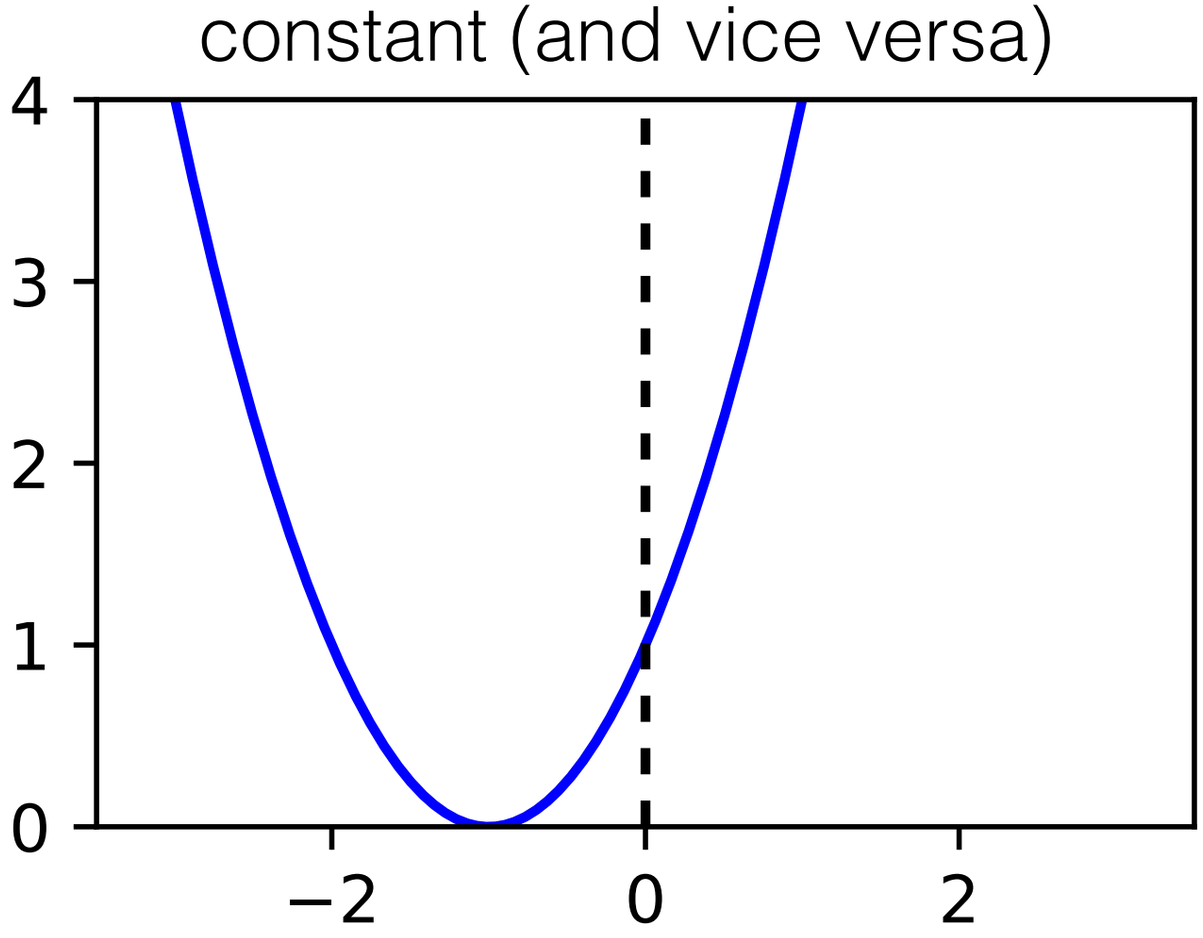
Consider the PCA loss for a single matrix element



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

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

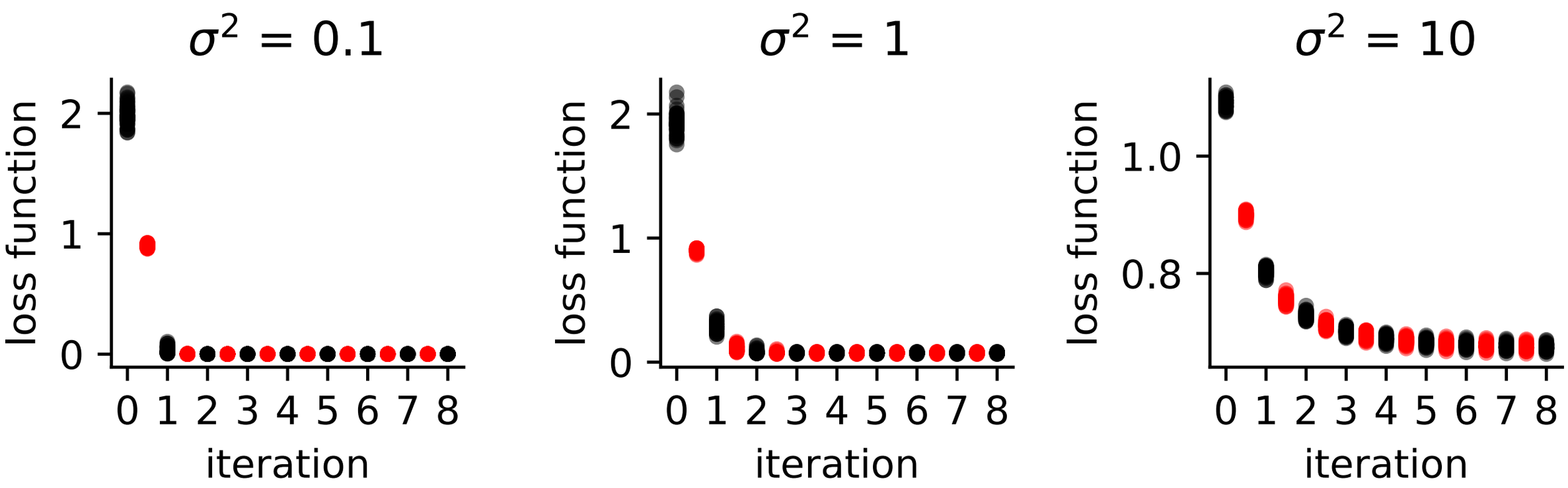




Alternating minimization is super effective in practice

Generally, not that many iterations are needed.

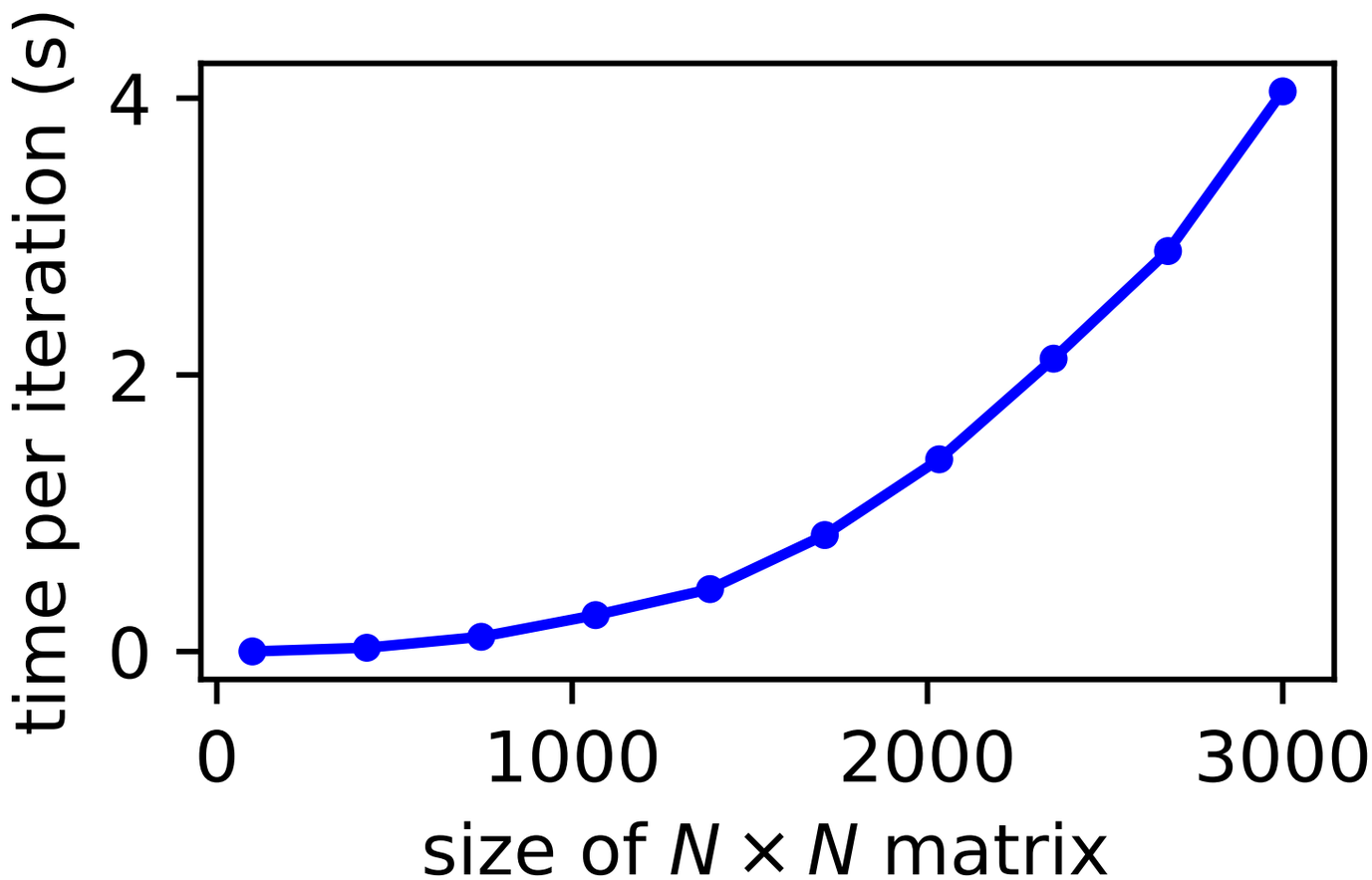
Update **U** Update **V**



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*