Data Mining using Matrix/Tensor Factorizations & Applications

Evrim Acar

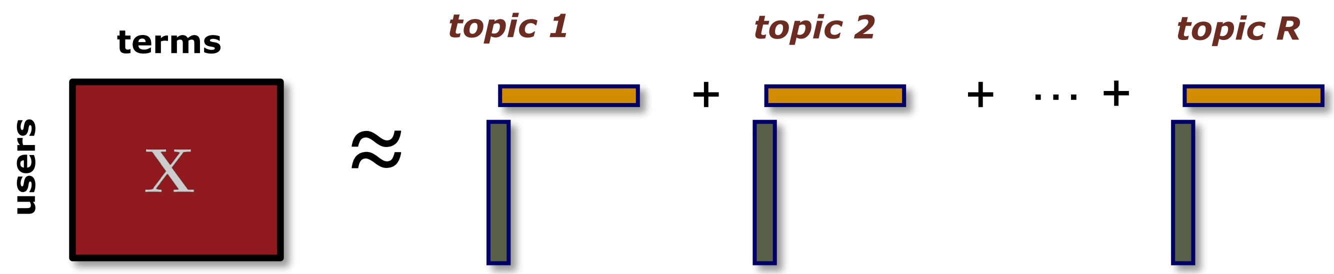
Simula Metropolitan Center for Digital Engineering

Oslo, Norway

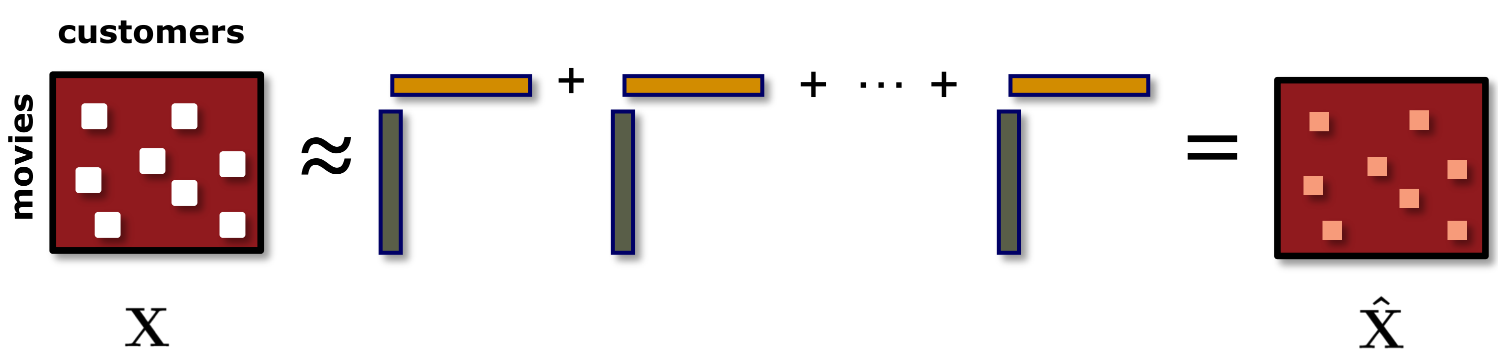
simula

Matrix Factorizations in Data Mining

Capturing underlying hidden factors/patterns



Incomplete Matrix Factorization/ Matrix Completion



simula

Data is often squeezed to be two-way!

2-way

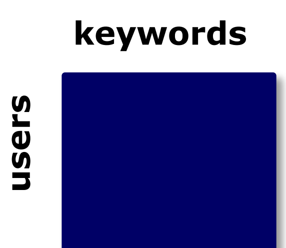
Social networks : **<users, keywords>**
<employee, employee>
<users, users>

Text mining: **< documents, terms>**

Recommender
Systems: **<customers, items>**

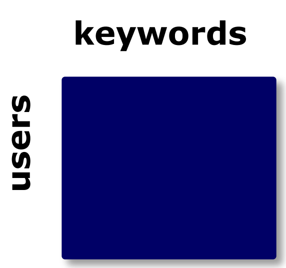
Computer vision: **<people, pixels>**

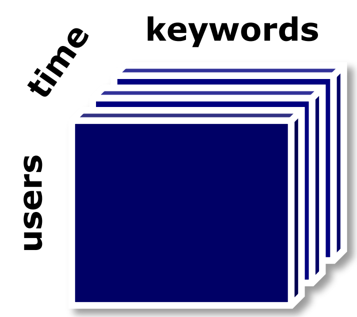
Neuroscience: **<electrodes, time>**



Data is often squeezed to be two-way!

	2-way	Multi-way
Social networks :	**<users, keywords>**	**< users, keywords, time>**
	<employee, employee>	**< employee, employee, time>**
	<users, users>	**< users, users, communication type>**
Text mining:	**< documents, terms>**	**< documents, terms, terms>**
Recommender Systems:	**<customers, items>**	**<customers, items, tags>**
Computer vision:	**<people, pixels>**	**<people, pixels, viewpoints>**
Neuroscience:	**<electrodes, time>**	**<electrodes, time, frequency>**
		<electrodes, time, subjects>

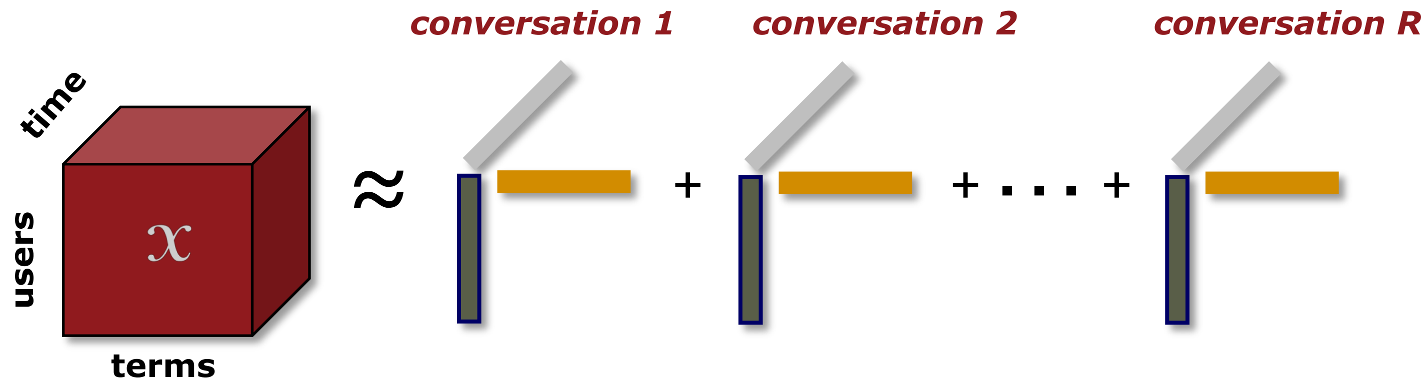




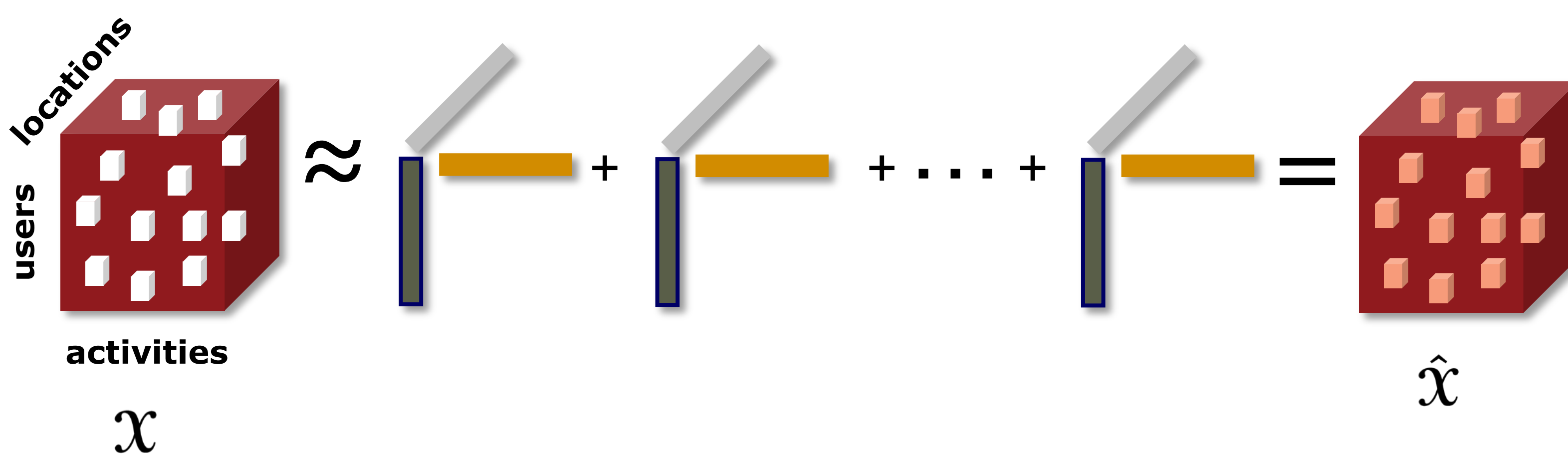
simula

Data is often multi-way!

Capturing underlying hidden factors/patterns

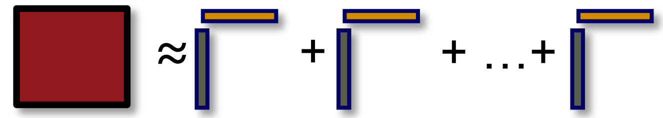


Incomplete Tensor Factorization/ Tensor Completion

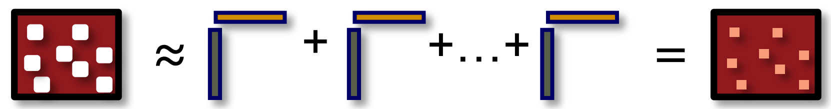


Matrix Factorizations in Data Mining

Matrix Factorizations

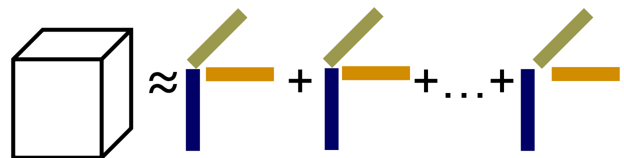


Matrix Completion

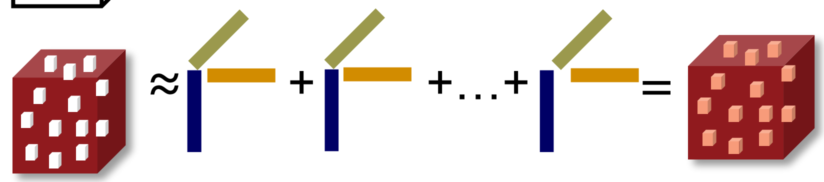


Data sets are often multi-way: Tensor Factorizations

Tensor Factorizations

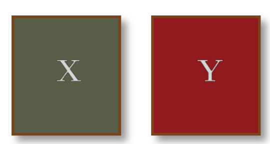


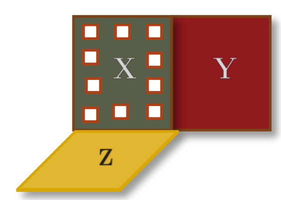
Tensor Completion



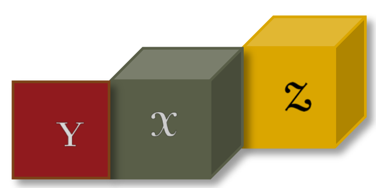
Data sets often come from multiple sources: Data Fusion

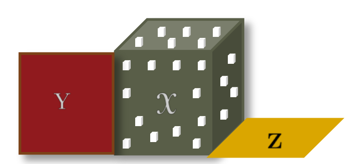
Coupled Matrix Factorizations





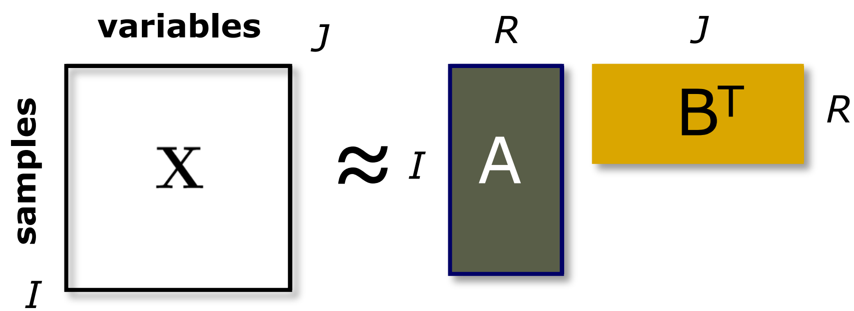
Coupled Tensor Factorizations





Matrix Factorizations

Matrix Factorizations, e.g., SVD (Singular Value Decomposition), NMF (Nonnegative Matrix Factorization), are commonly used in data mining.



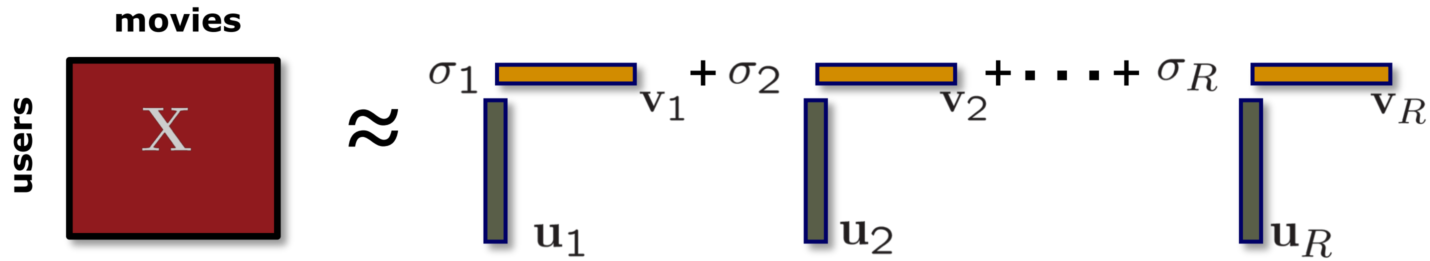
$$\mathbf{X} \approx \sum_{r=1}^{R} \mathbf{a}_r \mathbf{b}_r^\top$$
$$\approx \mathbf{A}\mathbf{B}^\top$$

$$\mathbf{A} \in \mathbb{R}^{I \times R} = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_R \end{bmatrix}$$

$$\mathbf{B} \in \mathbb{R}^{J \times R} = \begin{bmatrix} \mathbf{b}_1 & \cdots & \mathbf{b}_R \end{bmatrix}$$

Matrix Factorizations: Finding Underlying Structures

We can use SVD to find the underlying structures in a data set. For instance, using SVD of a *users* by *movies* matrix, we can capture the movie types and user groups.

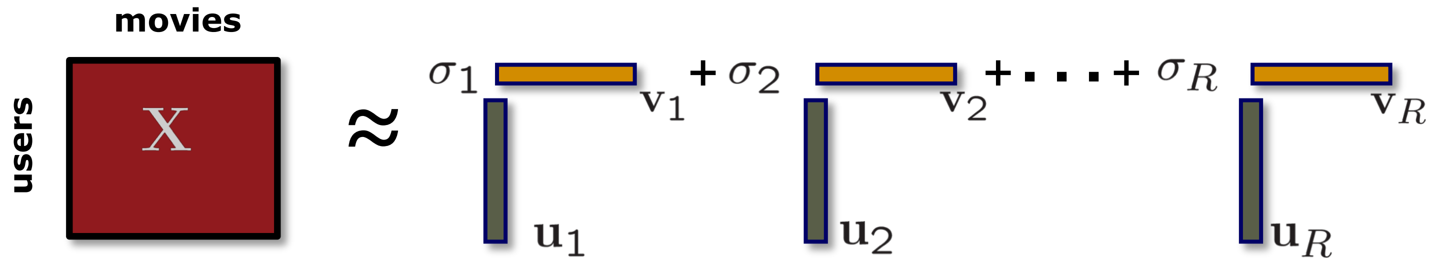


$$
\begin{aligned}
\mathbf{X} &\approx \sum_{r=1}^{R} \sigma_r \mathbf{u}_r \mathbf{v}_r^{\mathsf{T}} \\
&\approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{T}}
\end{aligned}
$$

$$
\begin{aligned}
\min_{\mathbf{U},\mathbf{V},\sigma} \; & \left\| \mathbf{X} - \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{T}} \right\|^2 \\
\texttt{s.t.} \quad & \mathbf{U}^{\mathsf{T}}\mathbf{U} = \mathbf{I}, \mathbf{V}^{\mathsf{T}}\mathbf{V} = \mathbf{I} \\
& \sigma_r \geq 0 \text{ for } r \in \{1 : R\}.
\end{aligned}
$$

Matrix Factorizations: Finding Underlying Structures

We can use SVD to find the underlying structures in a data set. For instance, using SVD of a *users* by *movies* matrix, we can capture the movie types and user groups.



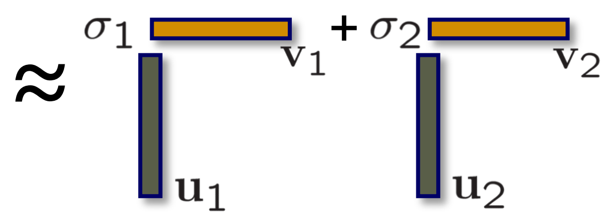
	Aloha	Star Wars	American Pie	Hunger Games	Silver Linings	Maze Runner
User 1	5	2	4	1	3	2
User 2	1	4	1	4	1	4
User 3	3	1	5	2	5	2
User 4	1	4	1	5	3	4
User 5	3	1	4	2	3	2
User 6	2	4	2	4	2	4

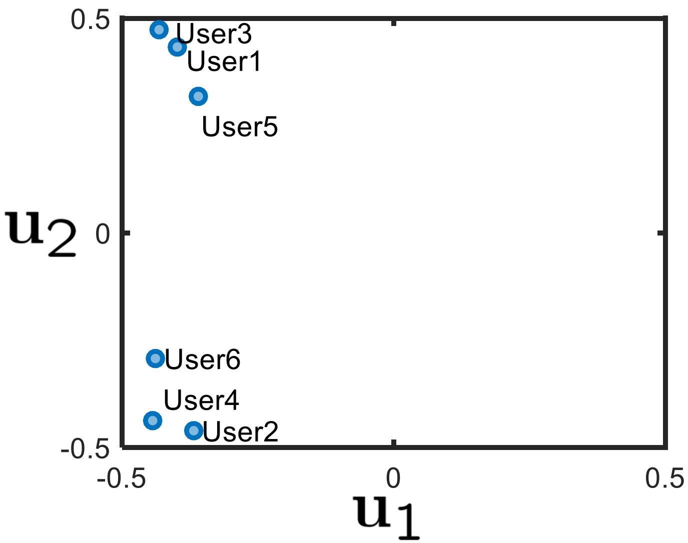
Who likes what type of movies?

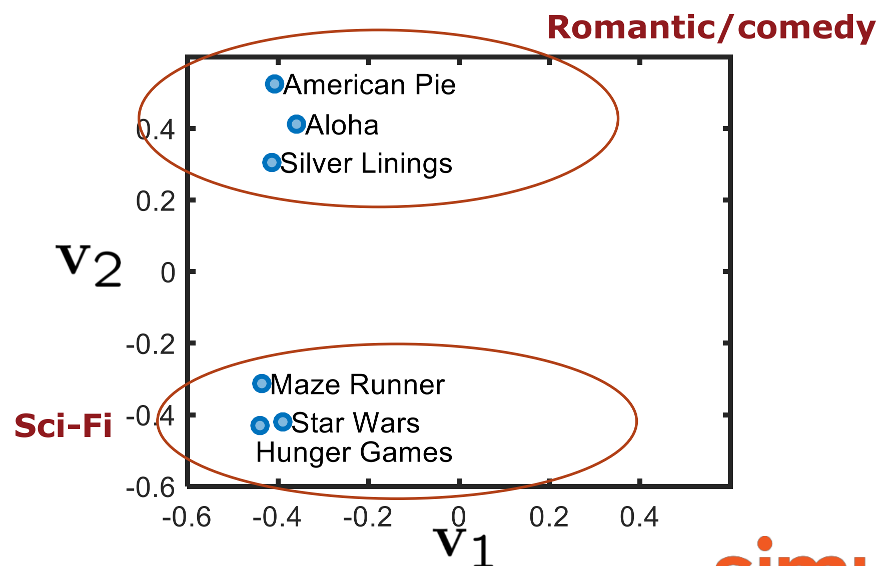
Matrix Factorizations: Finding Underlying Structures

Tool: Singular Value Decomposition

$$
\begin{array}{l}
 \\
\text{User 1} \\
\text{User 2} \\
\text{User 3} \\
\text{User 4} \\
\text{User 5} \\
\text{User 6}
\end{array}
\begin{array}{c}
\begin{array}{cccccc}
\text{Aloha} & \text{Star Wars} & \text{American Pie} & \text{Hunger Games} & \text{Silver Linings} & \text{Maze Runner}
\end{array} \\
\begin{bmatrix}
5 & 2 & 4 & 1 & 3 & 2 \\
1 & 4 & 1 & 4 & 1 & 4 \\
3 & 1 & 5 & 2 & 5 & 2 \\
1 & 4 & 1 & 5 & 3 & 4 \\
3 & 1 & 4 & 2 & 3 & 2 \\
2 & 4 & 2 & 4 & 2 & 4
\end{bmatrix}
\end{array}
$$







Principal Component Analysis (PCA)

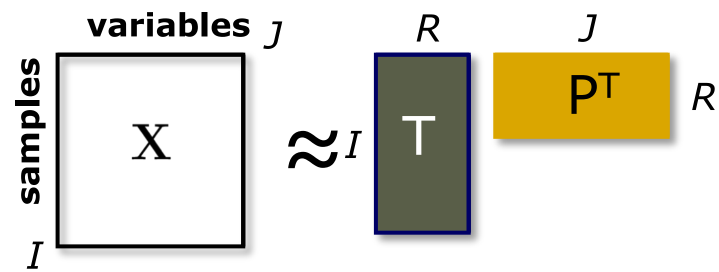
PCA is a dimension reduction method often used for exploratory analysis to find the main axes of variation in the data.

Let **X** be a *sample* by *variable* matrix and centered across the samples mode, e.g., each column has mean zero. PCA reveals scores and loadings, where columns of **P** show the principal component directions and **T** will have the corresponding principal component scores.

$$\mathbf{X} \approx \mathbf{T}\mathbf{P}^{\mathsf{T}}$$

loadings

scores



How do we find the scores and loadings?

Direction of the first principal component p_1 can be found by solving the following:

$$\max_{\|\mathbf{p}_1\|=1} \mathrm{var}(\mathbf{t}_1), \text{ where } \mathbf{t}_1 = \mathbf{X}\mathbf{p}_1$$

The second direction p_2 is then found in the same way with the constraint that p_1 is orthogonal to p_2

Eigenvalue decomposition of the covariance matrix gives the principal component directions.

$$\mathbf{X}^\mathsf{T}\mathbf{X} = \mathbf{P}\Lambda\mathbf{P}^\mathsf{T}$$

How does this relate to SVD?

$$\begin{aligned} \mathbf{X} &= \mathbf{U}\Sigma\mathbf{V}^\mathsf{T} \\ \mathbf{X}^\mathsf{T}\mathbf{X} &= \mathbf{V}\Sigma\mathbf{U}^\mathsf{T}\mathbf{U}\Sigma\mathbf{V}^\mathsf{T} \\ &= \mathbf{V}\Sigma^2\mathbf{V}^\mathsf{T} \end{aligned}$$

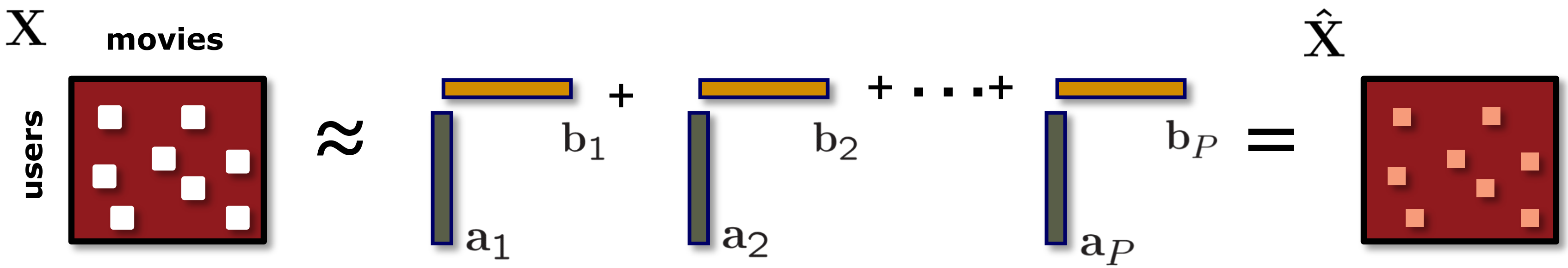
Right singular vectors are the principal component directions

The variance captured by each component is the square of the singular values

simula

Matrix Factorizations: Missing Data Estimation (a.k.a. matrix completion)

In the case of incomplete data, matrix factorizations can be used to fill in the missing entries:



Finding low-rank approximation:

$$\min_{\mathbf{A},\mathbf{B}} \left\| \mathbf{W} * (\mathbf{X} - \mathbf{A}\mathbf{B}^\top) \right\|^2$$

$$w_{ij} = \begin{cases} 1 & \text{if } x_{ij} \text{ is known,} \\ 0 & \text{if } x_{ij} \text{ is missing.} \end{cases}$$

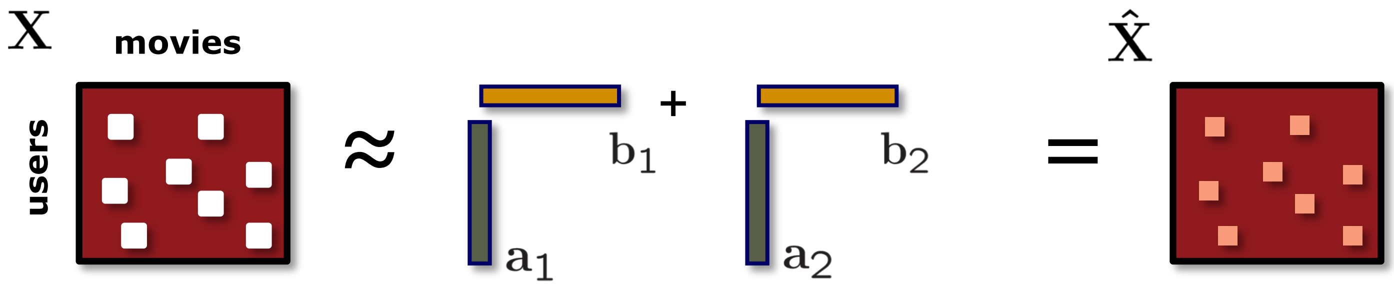
Data reconstruction:

$$\hat{\mathbf{X}} = \mathbf{A}\mathbf{B}^\top$$

It is possible to recover an unknown low-rank matrix from a nearly minimal set of entries. For more on matrix completion [Candes and Plan, 2010]

Matrix Factorizations: Missing Data Estimation

For instance, we can still factorize the incomplete **X** and then use those factors to reconstruct the matrix and estimate the missing entries.



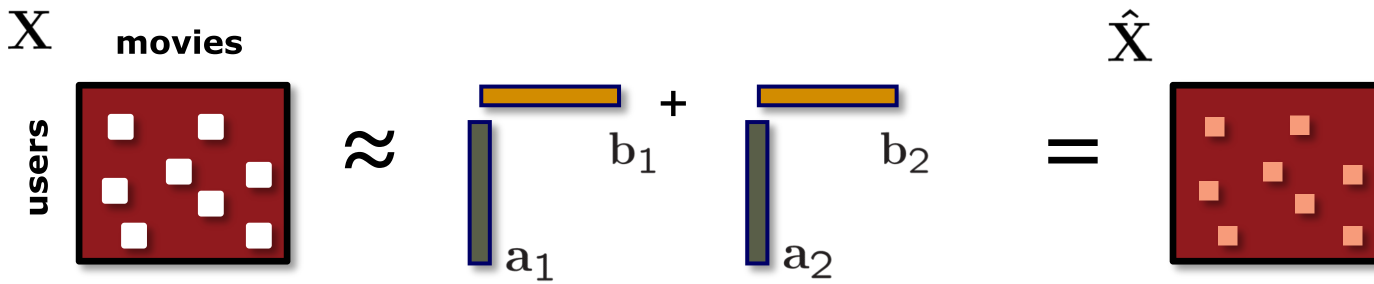
$$\hat{\mathbf{X}} = \mathbf{A}\mathbf{B}^{\top}$$

$$\mathbf{X} \quad \begin{array}{l|cccccc} & \text{Aloha} & \text{Star Wars} & \text{American Pie} & \text{Hunger Games} & \text{Silver Linings} & \text{Maze Runner} \\ \hline \text{User 1} & 5 & 2 & 4 & 1 & 3 & \color{red}{?} \\ \text{User 2} & 1 & 4 & 1 & 4 & 1 & 4 \\ \text{User 3} & 3 & 1 & 5 & 2 & 5 & 2 \\ \text{User 4} & 1 & 4 & 1 & \color{red}{?} & 3 & 4 \\ \text{User 5} & 3 & 1 & 4 & 2 & 3 & 2 \\ \text{User 6} & 2 & \color{red}{?} & 2 & 4 & 2 & 4 \end{array}$$

$$\begin{bmatrix} 3.72 & 1.27 & 4.41 & 1.60 & 3.78 & \color{red}{1.91} \\ 0.82 & 3.91 & 0.74 & 4.05 & 1.53 & 3.91 \\ 4.06 & 1.37 & 4.82 & 1.72 & 4.12 & 2.07 \\ 1.37 & 4.06 & 1.39 & \color{red}{4.24} & 2.08 & 4.15 \\ 3.15 & 1.38 & 3.71 & 1.65 & 3.25 & 1.90 \\ 1.77 & \color{red}{3.76} & 1.89 & 3.97 & 2.40 & 3.94 \end{bmatrix}$$

Matrix Factorizations: Missing Data Estimation

For instance, we can still factorize the incomplete **X** and then use those factors to reconstruct the matrix and estimate the missing entries.



$$\hat{\mathbf{X}} = \mathbf{A}\mathbf{B}^\top$$

$$\mathbf{X} \quad \begin{array}{l} \\ \textbf{User 1} \\ \textbf{User 2} \\ \textbf{User 3} \\ \textbf{User 4} \\ \textbf{User 5} \\ \textbf{User 6} \end{array} \begin{array}{c} \begin{array}{cccccc} \textbf{Aloha} & \textbf{Star Wars} & \textbf{American Pie} & \textbf{Hunger Games} & \textbf{Silver Linings} & \textbf{Maze Runner} \end{array} \\ \begin{bmatrix} 5 & 2 & 4 & 1 & 3 & 2 \\ 1 & 4 & 1 & 4 & 1 & 4 \\ 3 & 1 & 5 & 2 & 5 & 2 \\ 1 & 4 & 1 & 5 & 3 & 4 \\ 3 & 1 & 4 & 2 & 3 & 2 \\ 2 & 4 & 2 & 4 & 2 & 4 \end{bmatrix} \end{array}$$

$$\begin{bmatrix} 3.72 & 1.27 & 4.41 & 1.60 & 3.78 & 1.91 \\ 0.82 & 3.91 & 0.74 & 4.05 & 1.53 & 3.91 \\ 4.06 & 1.37 & 4.82 & 1.72 & 4.12 & 2.07 \\ 1.37 & 4.06 & 1.39 & 4.24 & 2.08 & 4.15 \\ 3.15 & 1.38 & 3.71 & 1.65 & 3.25 & 1.90 \\ 1.77 & 3.76 & 1.89 & 3.97 & 2.40 & 3.94 \end{bmatrix}$$

Text Mining

Suppose we have the documents with the following titles and we want to group them into topics:

Document 1: Tiger stopped playing golf

Document 2*:* **News** about **Tiger** and his **golf career**

Document 3: Golf career of **Tiger** in **jeopardy**

Document 4: Tiger and his **wife** in the **news**

Document 5: The **new zoo featuring** the **big cat family**: **tigers** and **lions**

Document 6: Tigers – the **big cats** – in the **new zoo**

Document 7: Tigers and **lions**, **which** are the **biggest cats**?

	tiger	stop	play	golf	news	career	jeopardy	wife	zoo	featuring	new	big	cat	family	lions	which
Doc1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Doc2	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
Doc3	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
Doc4	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
Doc5	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
Doc6	1	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0
Doc7	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1

Text Mining (cont.)

Document 1: **Tiger stopped playing golf**

Document 2: **News** about **Tiger** and his **golf career**

Document 3: **Golf career** of **Tiger** in **jeopardy**

Document 4: **Tiger** and his **wife** in the **news**

Document 5: The **new zoo featuring** the **big cat family**: **tigers** and **lions**

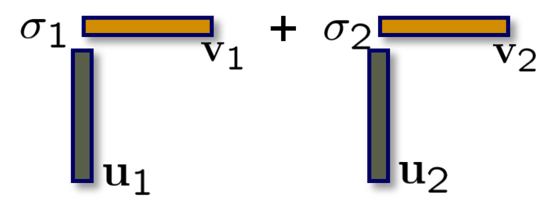
Document 6: **Tigers** – the **big cats** – in the **new zoo**

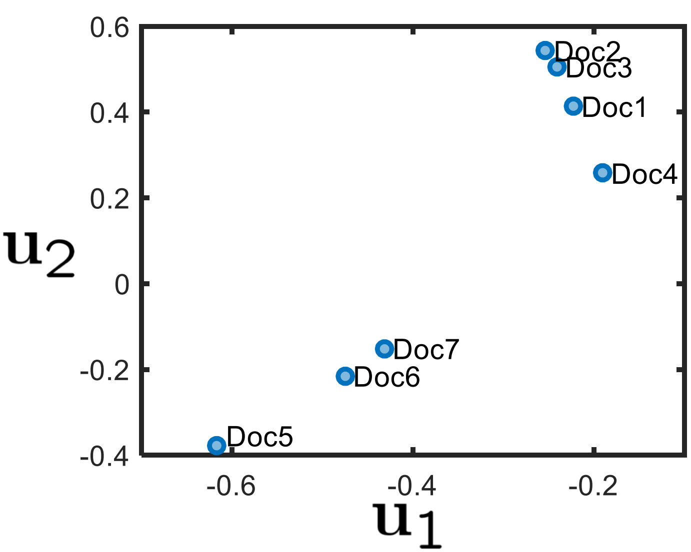
Document 7: **Tigers** and **lions**, **which** are the **biggest cats**?

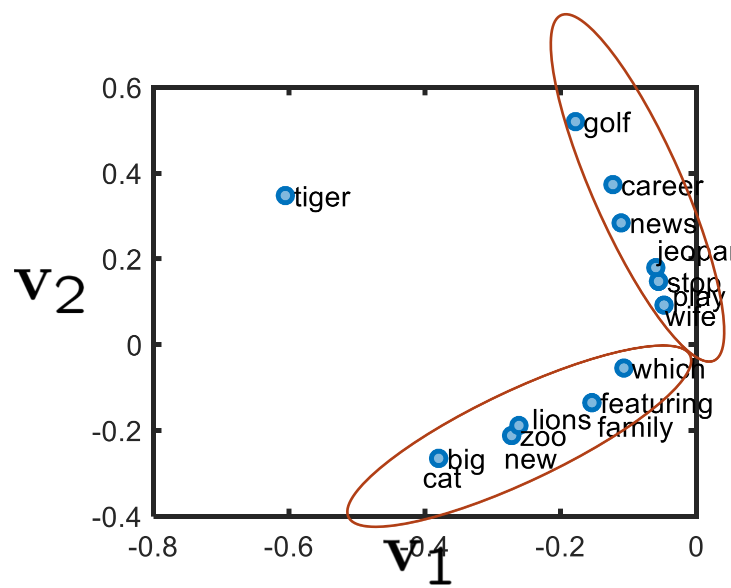
X

	tiger	stop	play	golf	news	career	jeopardy	wife	zoo	featuring	new	big	cat	family	lions	which
Doc1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Doc2	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
Doc3	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
Doc4	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
Doc5	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
Doc6	1	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0
Doc7	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top \approx$$







You may be interested in:

- ***Clustering***
- ***Document classification***

Text Mining (cont.)

Document 1: **Tiger stopped playing golf**

Document 2: **News** about **Tiger** and his **golf career**

Document 3: **Golf career** of **Tiger** in **jeopardy**

Document 4: **Tiger** and his **wife** in the **news**

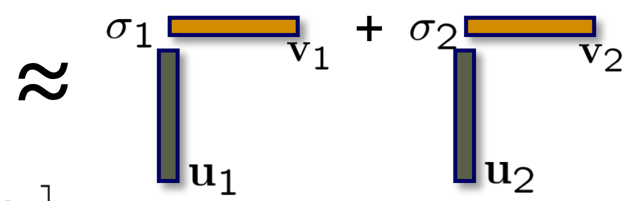
Document 5: The **new zoo featuring** the **big cat family**: **tigers** and **lions**

Document 6: **Tigers** – the **big cats** – in the **new zoo**

Document 7: **Tigers** and **lions**, **which** are the **biggest cats**?

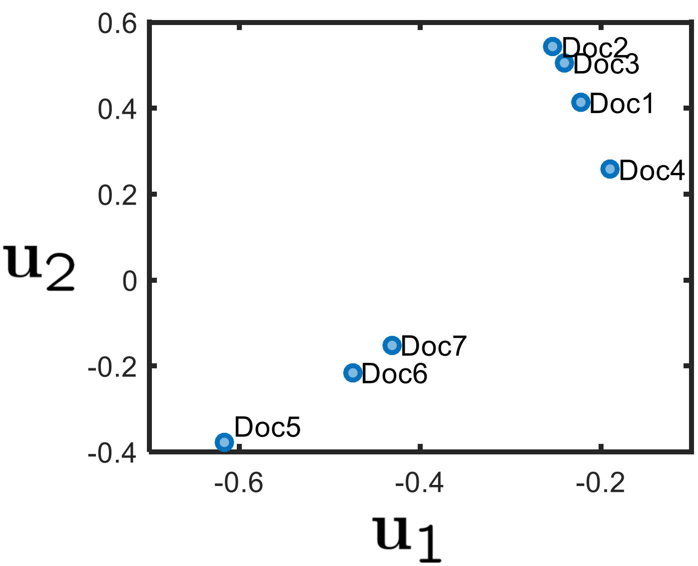
$\mathbf{X}$

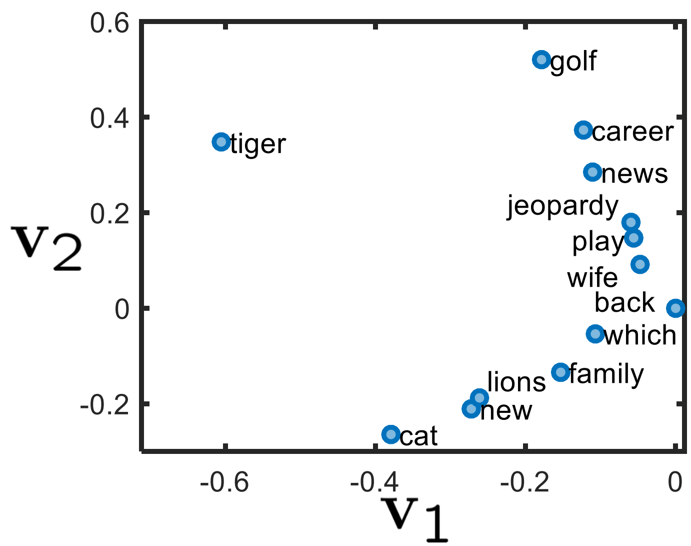
	tiger	stop	play	golf	news	career	jeopardy	wife	zoo	featuring	new	big	cat	family	lions	which
Doc1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Doc2	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
Doc3	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
Doc4	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
Doc5	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
Doc6	1	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0
Doc7	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1



Document 8: **Tiger** back to **golf**

$$\mathrm{q}^{\mathsf{T}} = \left[\begin{array}{cccccccccccccccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$





You may be interested in:
- ***Clustering***
- ***Document classification***

simula

Text Mining (cont.)

Tool: Singular Value Decomposition

Document 1: Tiger stopped playing golf

Document 2: News about **Tiger** and his **golf career**

Document 3: Golf career of **Tiger** in **jeopardy**

Document 4: Tiger and his **wife** in the **news**

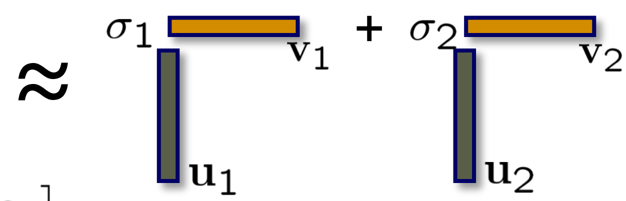
Document 5: The **new zoo featuring** the **big cat family**: **tigers** and **lions**

Document 6: Tigers – the **big cats** – in the **new zoo**

Document 7: Tigers and **lions**, **which** are the **biggest cats**?

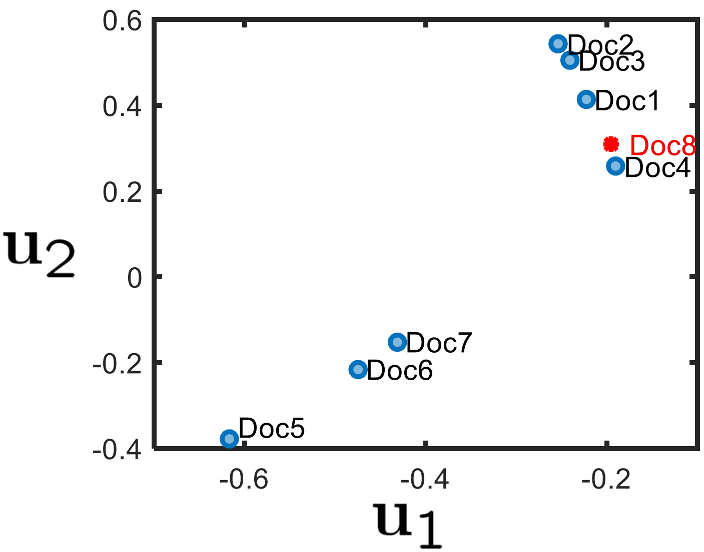
X

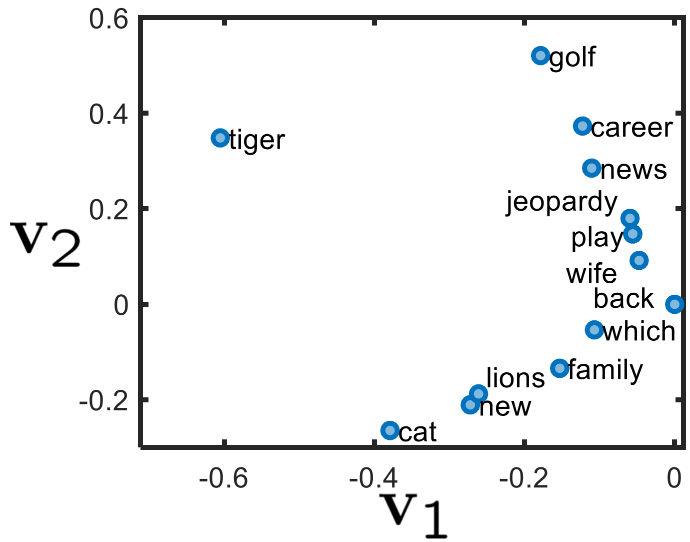
	tiger	stop	play	golf	news	career	jeopardy	wife	zoo	featuring	new	big	cat	family	lions	which
Doc1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Doc2	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
Doc3	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
Doc4	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
Doc5	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
Doc6	1	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0
Doc7	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1



Document 8: Tiger back to **golf**

$$q^T = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$





You may be interested in:

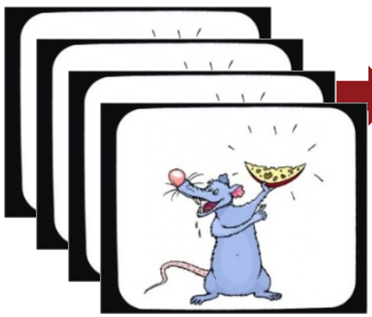
- ***Clustering***
- ***Document classification***

simula

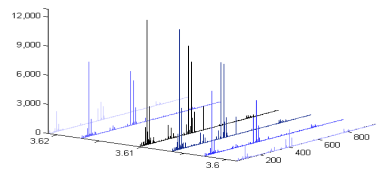
Metabolomics

The goal is to **detect** a wide range of **metabolites** in biological samples, e.g., blood or urine, and to **discover** the significant metabolic **biomarkers** related to certain conditions such as food intake or various diseases.

Some are fed with 10g apple while some are controls (no apple)



Liquid Chromatography-Mass Spectrometry (LC-MS)



$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{T}}$$

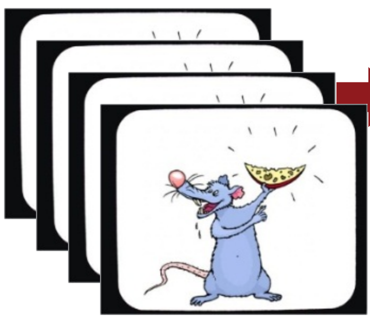
$$\approx \sigma_1 \, \mathbf{u}_1 \mathbf{v}_1 + \sigma_2 \, \mathbf{u}_2 \mathbf{v}_2 + \cdots + \sigma_R \, \mathbf{u}_R \mathbf{v}_R$$

Metabolomics

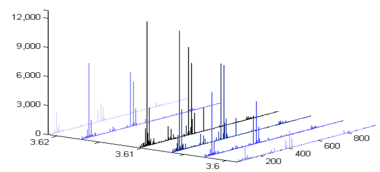
Tool: Singular Value Decomposition

The goal is to **detect** a wide range of **metabolites** in biological samples, e.g., blood or urine, and to **discover** the significant metabolic **biomarkers** related to certain conditions such as food intake or various diseases.

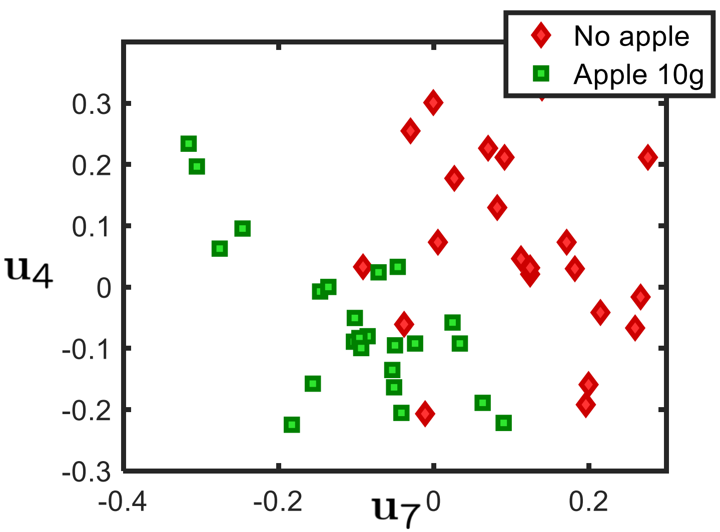
Some are fed with 10g apple while some are controls (no apple)



Liquid Chromatography-Mass Spectrometry (LC-MS)

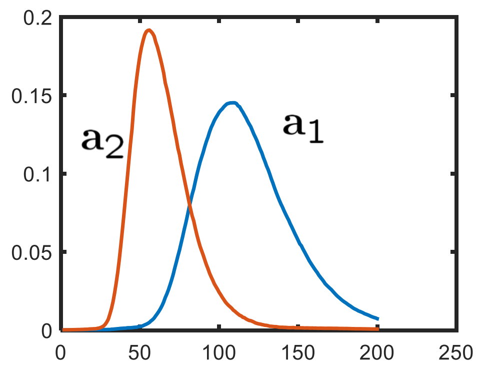


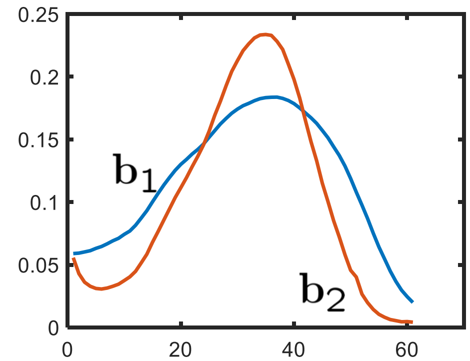
$$X = U\Sigma V^{T}$$



You may then check the corresponding columns in V to discover the metabolites

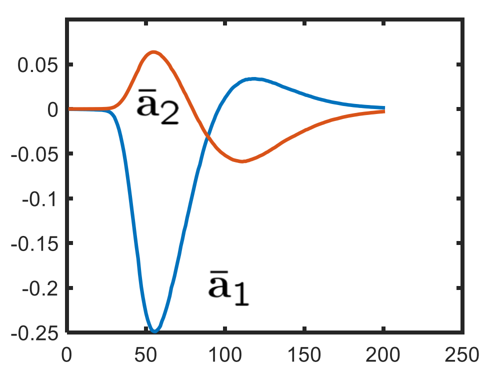
Matrix factorizations are not unique!

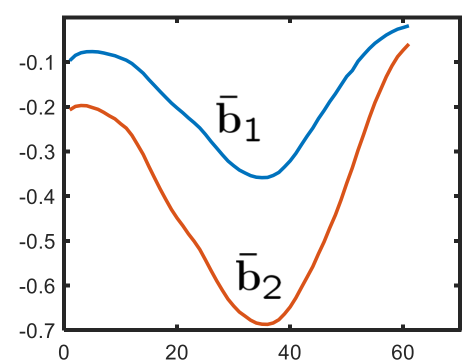




$\mathbf{A} \in \mathbb{R}^{201\times 2} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2\end{bmatrix}$ $\mathbf{B} \in \mathbb{R}^{61\times 2} = \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2\end{bmatrix}$

$$\begin{aligned}\mathbf{X} &= \mathbf{A}\mathbf{B}^\mathsf{T} \\ &= \mathbf{A}\mathbf{M}\mathbf{M}^{-1}\mathbf{B}^\mathsf{T} \\ &= \bar{\mathbf{A}}\bar{\mathbf{B}}^\mathsf{T}\end{aligned}$$





simula

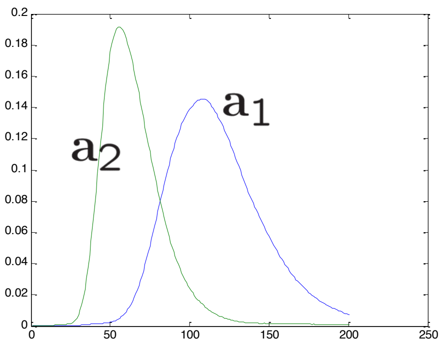
Uniqueness is an issue

$$\mathbf{X} = \mathbf{A}\mathbf{B}^\mathsf{T} = \mathbf{A}\mathbf{M}\mathbf{M}^{-1}\mathbf{B}^\mathsf{T} = \bar{\mathbf{A}}\bar{\mathbf{B}}^\mathsf{T}$$

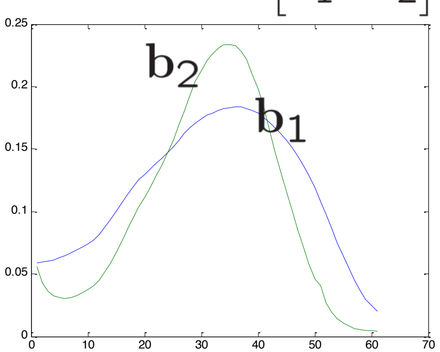
Constraints are used to deal with the uniqueness problem, e.g., SVD. However, factorizations with constraints may not be meaningful in terms of the application.

True factors

$$\mathbf{A} \in \mathbb{R}^{201\times 2} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2\end{bmatrix}$$

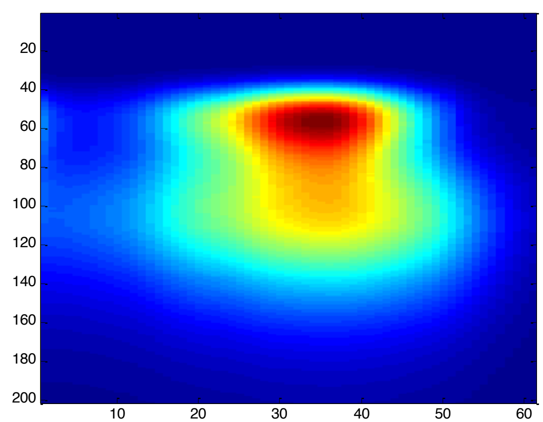


$$\mathbf{B} \in \mathbb{R}^{61\times 2} = \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2\end{bmatrix}$$



Data matrix

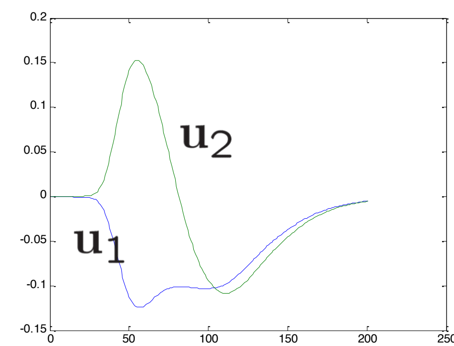
$$\mathbf{X} = \mathbf{A}\mathbf{B}^\mathsf{T}$$

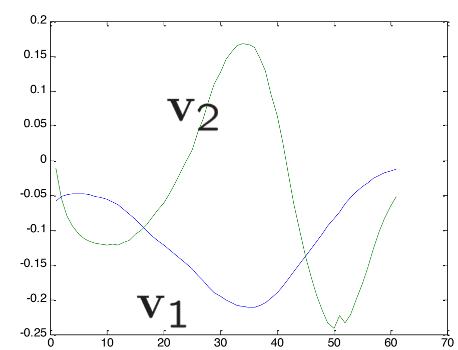


Given X, can we recover the true factors?

SVD captures...

$$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\mathsf{T}$$





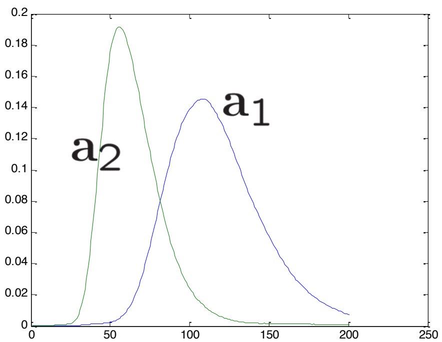
Uniqueness is an issue

$$\mathbf{X} = \mathbf{A}\mathbf{B}^\top = \mathbf{A}\mathbf{M}\mathbf{M}^{-1}\mathbf{B}^\top = \bar{\mathbf{A}}\bar{\mathbf{B}}^\top$$

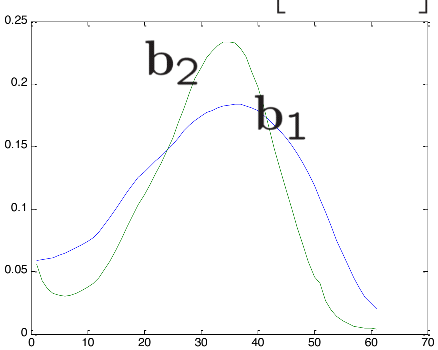
Constraints are used to deal with the uniqueness problem, e.g., SVD. However, factorizations with constraints may not be meaningful in terms of the application.

True factors

$$\mathbf{A} \in \mathbb{R}^{201\times 2} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2\end{bmatrix}$$

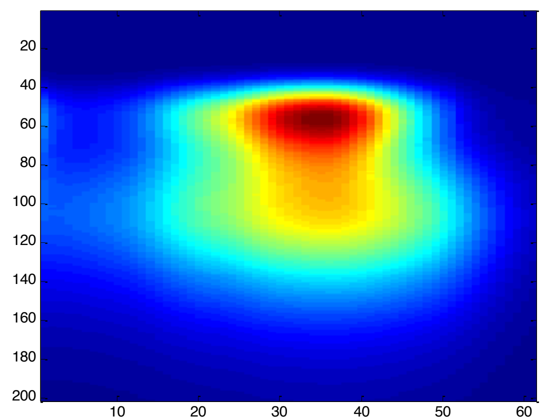


$$\mathbf{B} \in \mathbb{R}^{61\times 2} = \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2\end{bmatrix}$$



Data matrix

$$\mathbf{X} = \mathbf{A}\mathbf{B}^\top$$

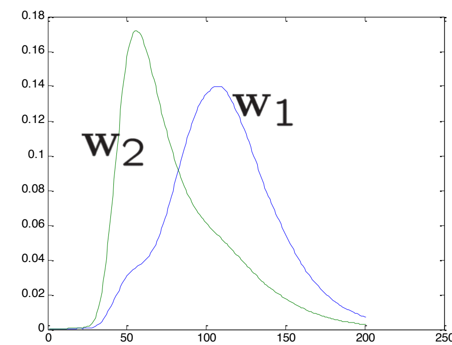


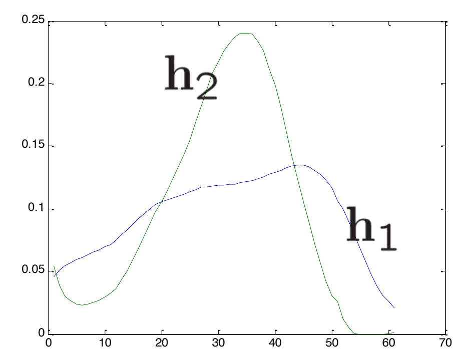
Given X, can we recover the true factors?

NMF captures...

$$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{W}\mathbf{H}^\top$$

$$w_{ir}, h_{jr} \geq 0$$





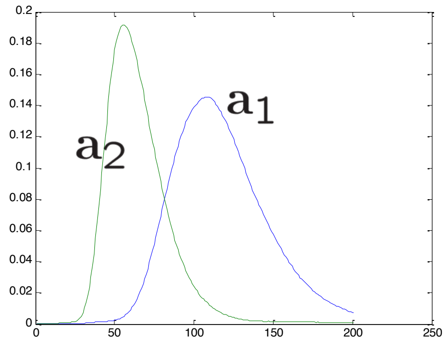
Uniqueness is an issue

$$\mathbf{X} = \mathbf{A}\mathbf{B}^\top = \mathbf{A}\mathbf{M}\mathbf{M}^{-1}\mathbf{B}^\top = \bar{\mathbf{A}}\bar{\mathbf{B}}^\top$$

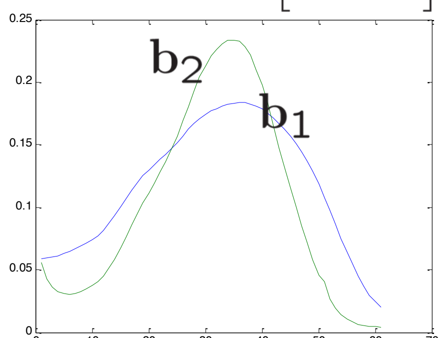
Constraints are used to deal with the uniqueness problem, e.g., SVD. However, factorizations with constraints may not be meaningful in terms of the application.

True factors

$$\mathbf{A} \in \mathbb{R}^{201\times 2} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2\end{bmatrix}$$

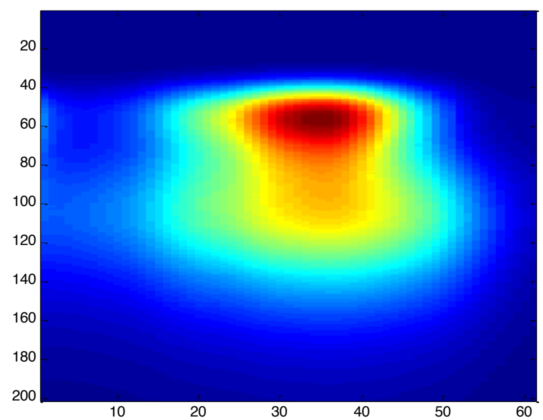


$$\mathbf{B} \in \mathbb{R}^{61\times 2} = \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2\end{bmatrix}$$



Data matrix

$$\mathbf{X} = \mathbf{A}\mathbf{B}^\top$$

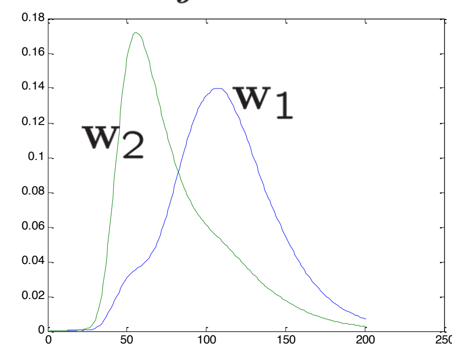


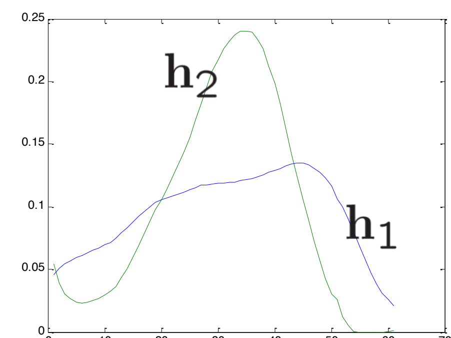
Given X, can we recover the true factors?

NMF captures...

$$\mathbf{X} \approx \hat{\mathbf{X}} = \underbrace{\mathbf{W}\mathbf{M}}_{\bar{\mathbf{W}}}\underbrace{\mathbf{M}^{-1}\mathbf{H}^\top}_{\bar{\mathbf{H}}^\top}$$

$$\bar{w}_{ir}, \bar{h}_{jr} \geq 0$$

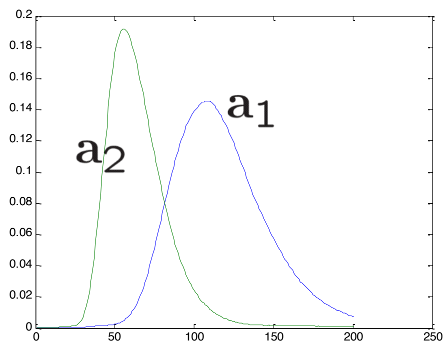




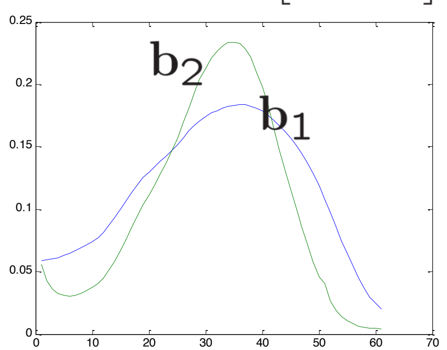
What if we have multiple matrices with the same underlying factors but in different proportions…

True factors

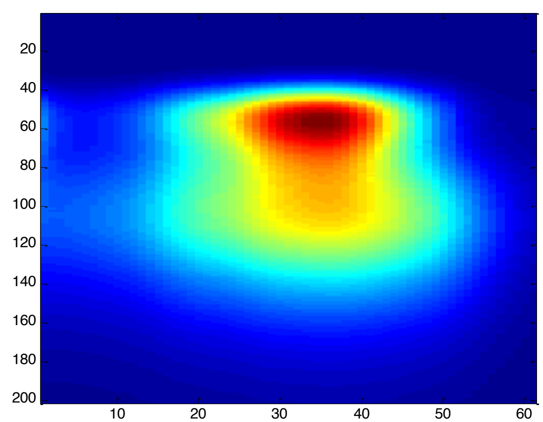
$$\mathbf{A} \in \mathbb{R}^{201\times 2} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2\end{bmatrix}$$



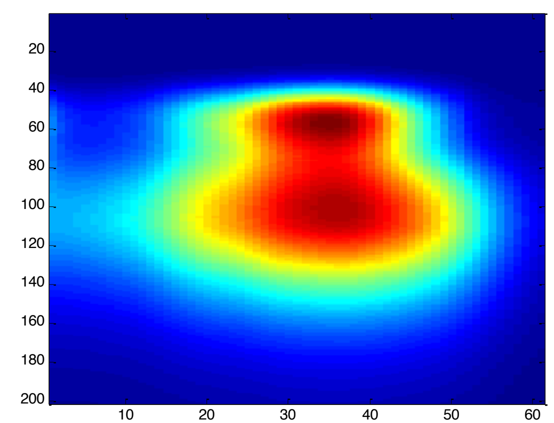
$$\mathbf{B} \in \mathbb{R}^{61\times 2} = \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2\end{bmatrix}$$



$$\mathbf{X}_1 = \mathbf{A}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\mathbf{B}^\mathsf{T}$$



$$\mathbf{X}_2 = \mathbf{A}\begin{bmatrix}3 & 0\\ 0 & 2\end{bmatrix}\mathbf{B}^\mathsf{T}$$



We can recover the true factors uniquely up to trivial indeterminacies, i.e., scaling and permutation.

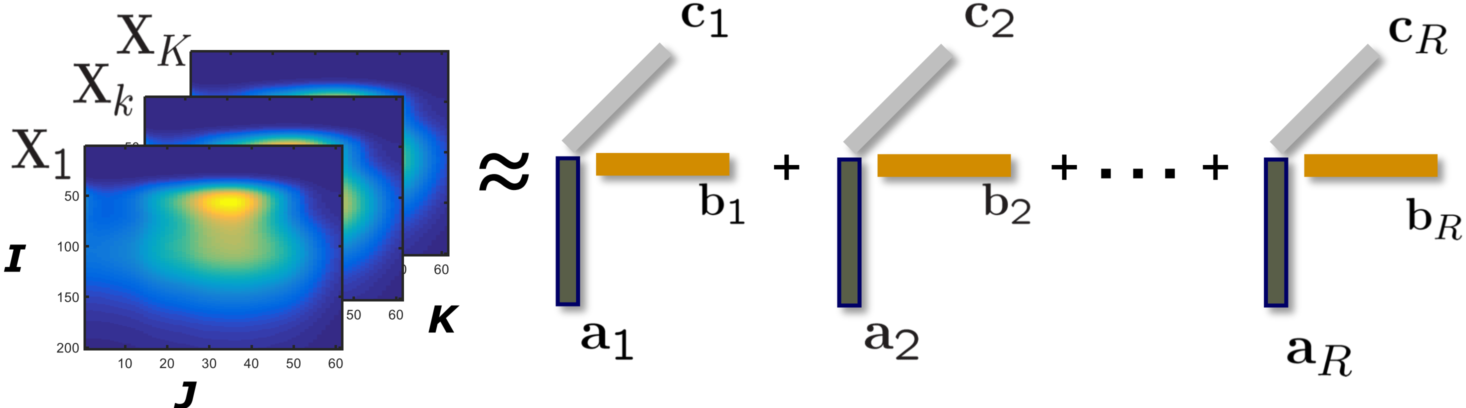
simula

A popular tensor factorization model: CP

Canonical Polyadic (CP)
CANDECOMP/PARAFAC (CP)

Hitchcock, 1927: Polyadic form of a tensor
Harshman, 1970: Parallel Factor Analysis (PARAFAC)
Carroll & Chang, 1970: Canonical Decomposition (CANDECOMP)

As an extension of matrix factorizations to higher-order tensors (multi-way arrays), tensor factorizations are used to extract the underlying factors in higher-order data sets. The CP model represents a tensor as a sum of rank-one tensors:



$$X_k \approx A \begin{bmatrix} c_{k1} & & \\ & \ddots & \\ & & c_{kR} \end{bmatrix} B^{\mathsf{T}}$$

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

$$\mathbf{A} \in \mathbb{R}^{I \times R} = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_R \end{bmatrix}$$

$$\mathbf{B} \in \mathbb{R}^{J \times R} = \begin{bmatrix} \mathbf{b}_1 & \cdots & \mathbf{b}_R \end{bmatrix}$$

$$\mathbf{C} \in \mathbb{R}^{K \times R} = \begin{bmatrix} \mathbf{c}_1 & \cdots & \mathbf{c}_R \end{bmatrix}$$

CP is unique up to scaling and permutation under the condition that

$$\text{krank}(\mathbf{A}) + \text{krank}(\mathbf{B}) + \text{krank}(\mathbf{C}) \geq 2R + 2$$

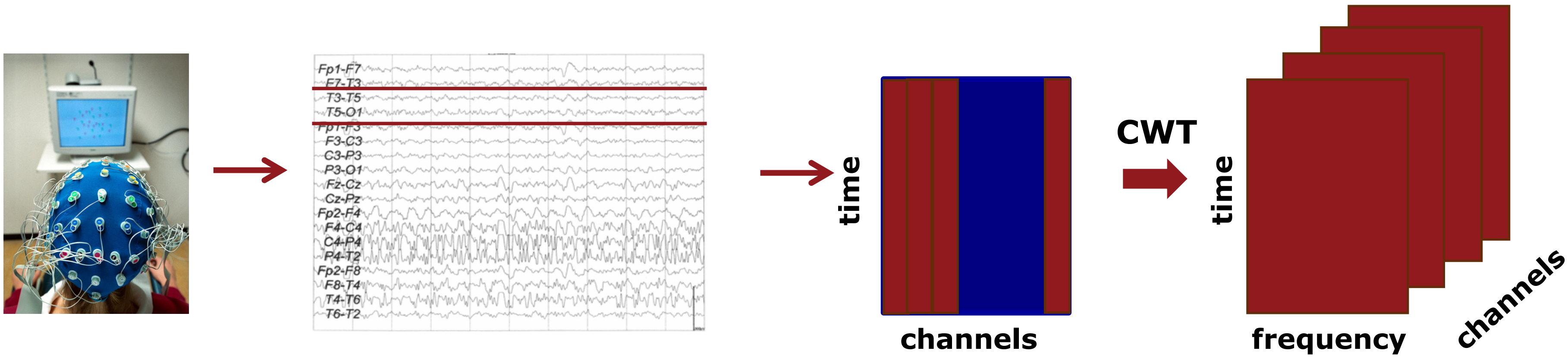
[Kruskal, 1977; Sidiropoulos and Bro, 2000]

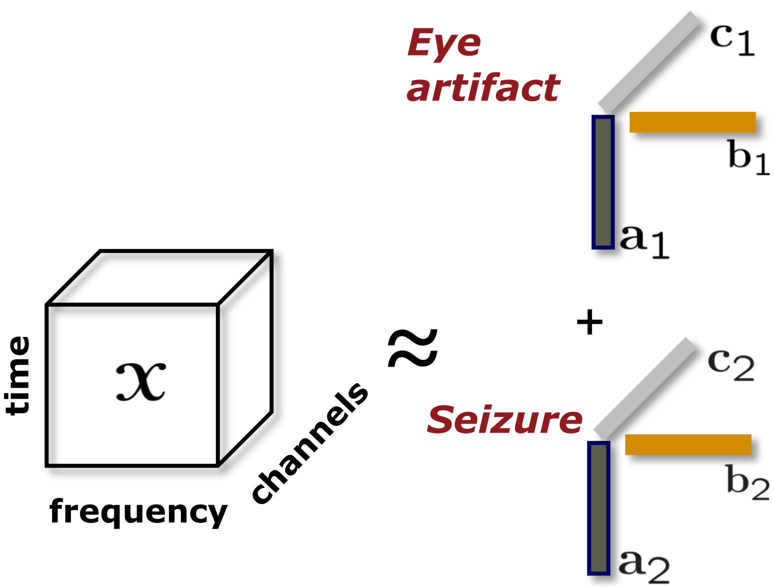
where krank (A) = max. value of *k* such that any *k* columns of A is linearly independent.

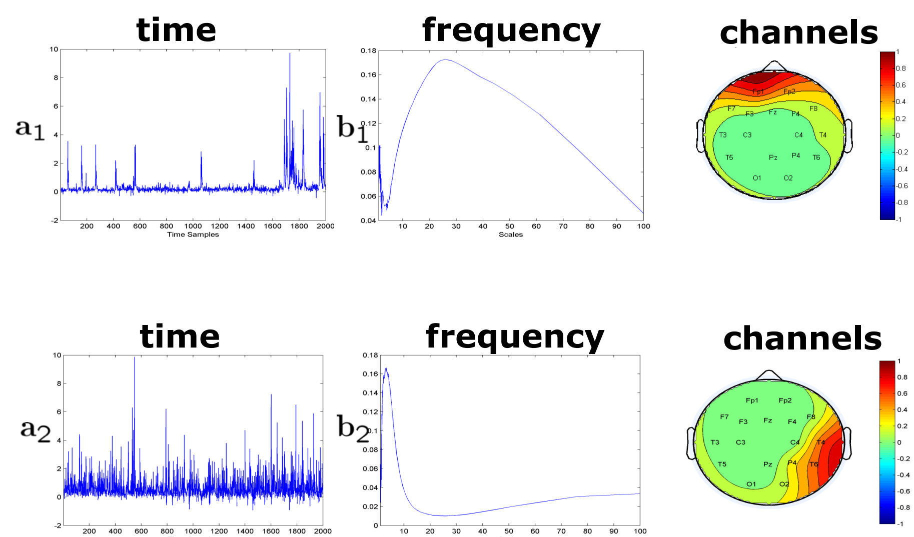
Neuroscience: CP has proved useful in epileptic seizure localization

[Acar et al., 2007; De Vos et al., 2007]

Tool: Tensor Factorizations







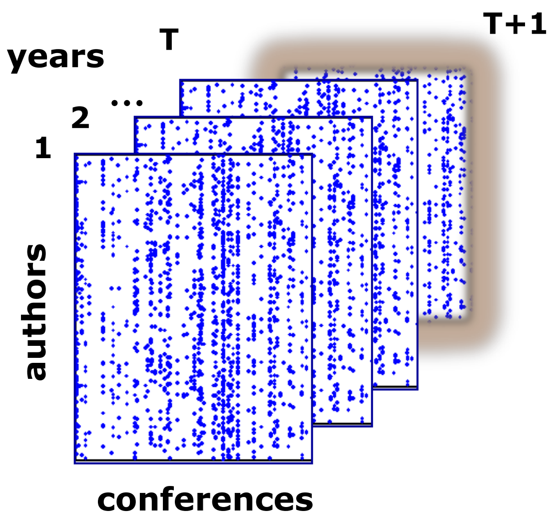
Recommender Systems: CP can capture temporal patterns useful for link prediction

Tool: Tensor Factorizations

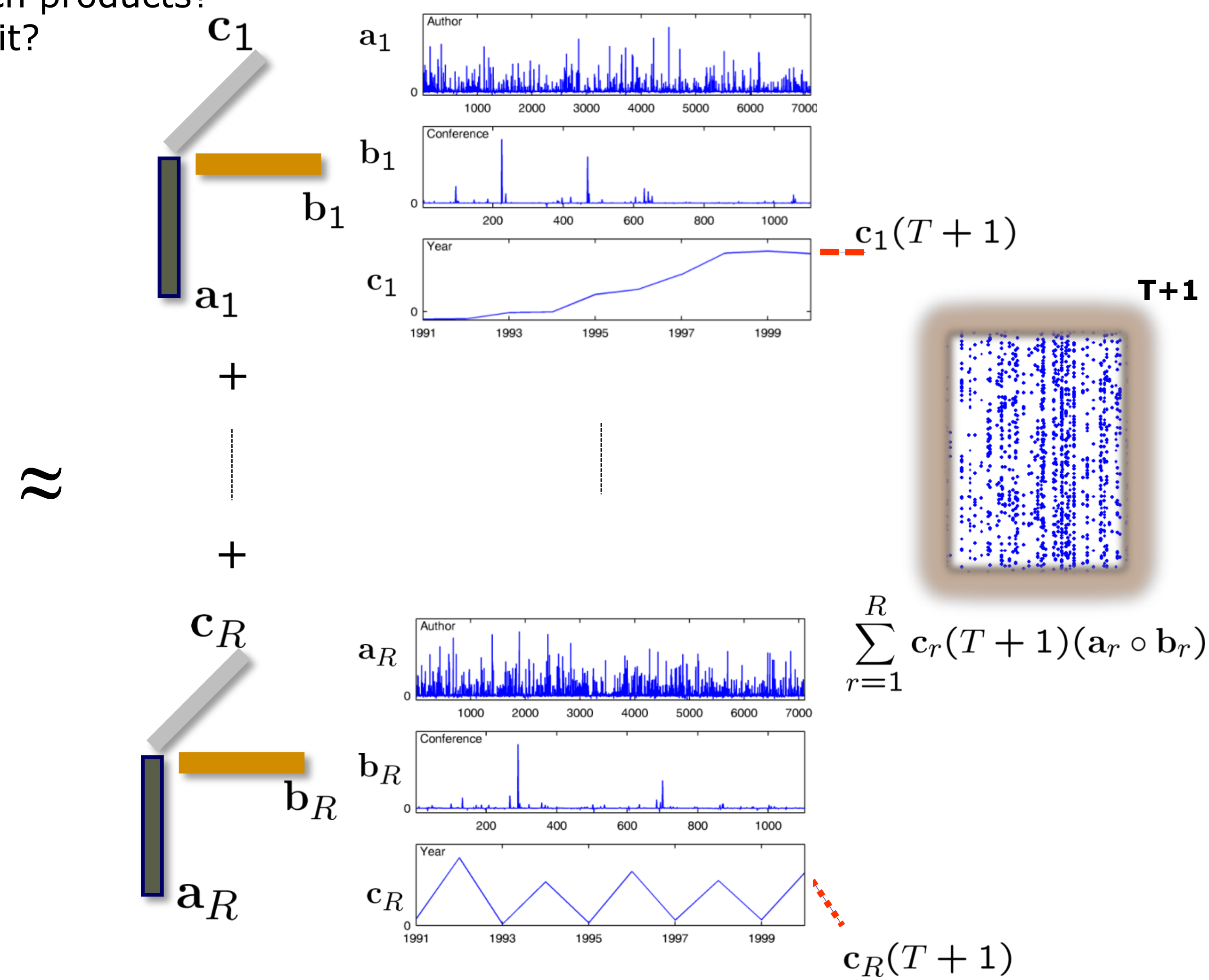
[Dunlavy, Kolda, Acar, 2011]

Temporal Link Prediction

Which customers will buy which products?
Which webpages will users visit?



$\approx$

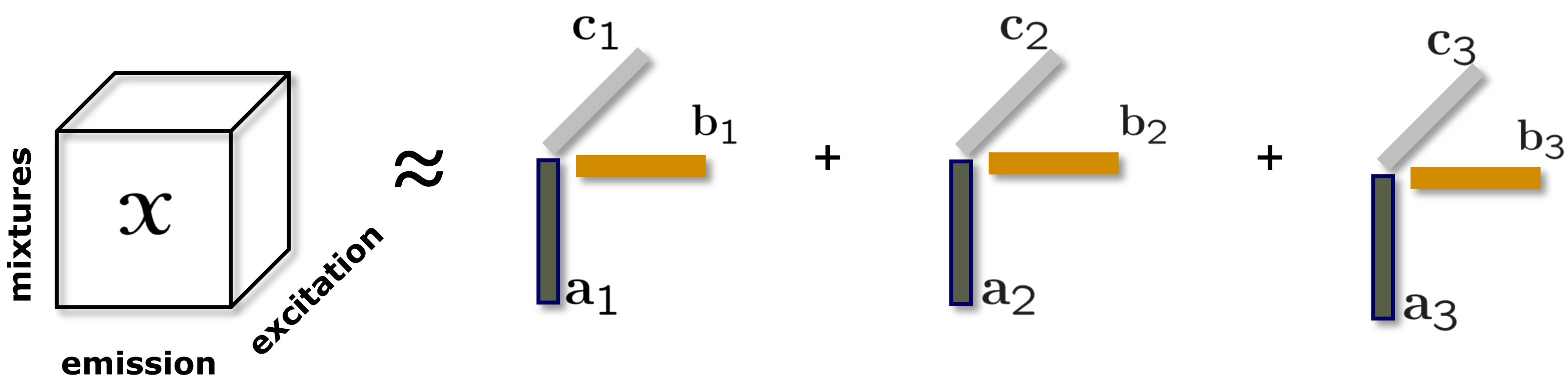


simula

Chemometrics: CP can separate the chemicals in mixtures

[Andersen and Bro, 2003]

A popular application of the CP model is the separation of individual chemicals from mixtures of chemicals measured using fluorescence spectroscopy.



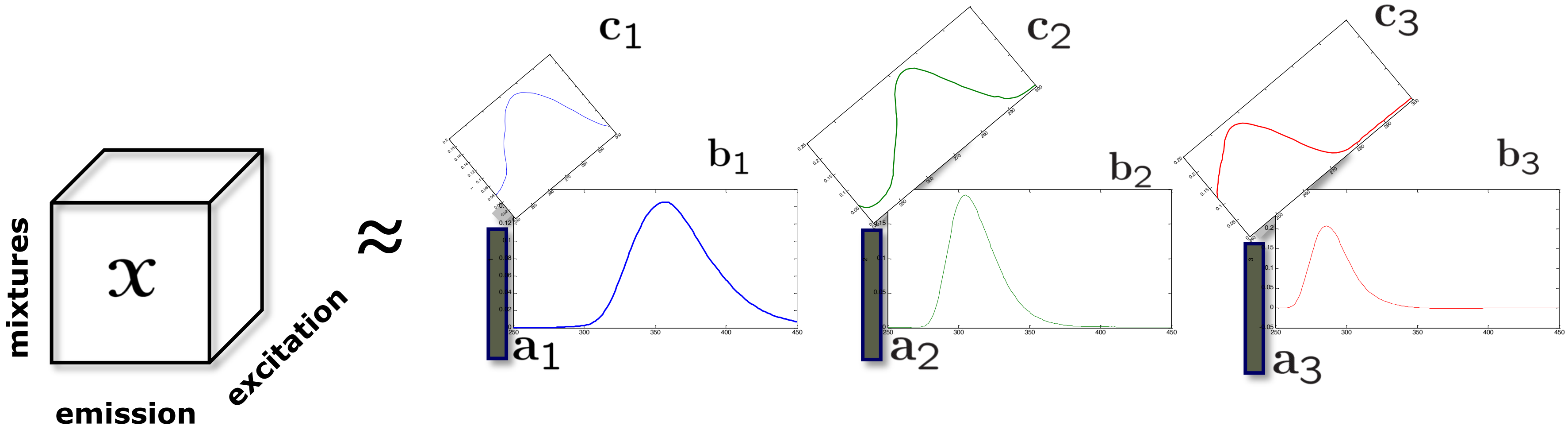
Amino Acid Data
http://www.models.life.ku.dk/

simula

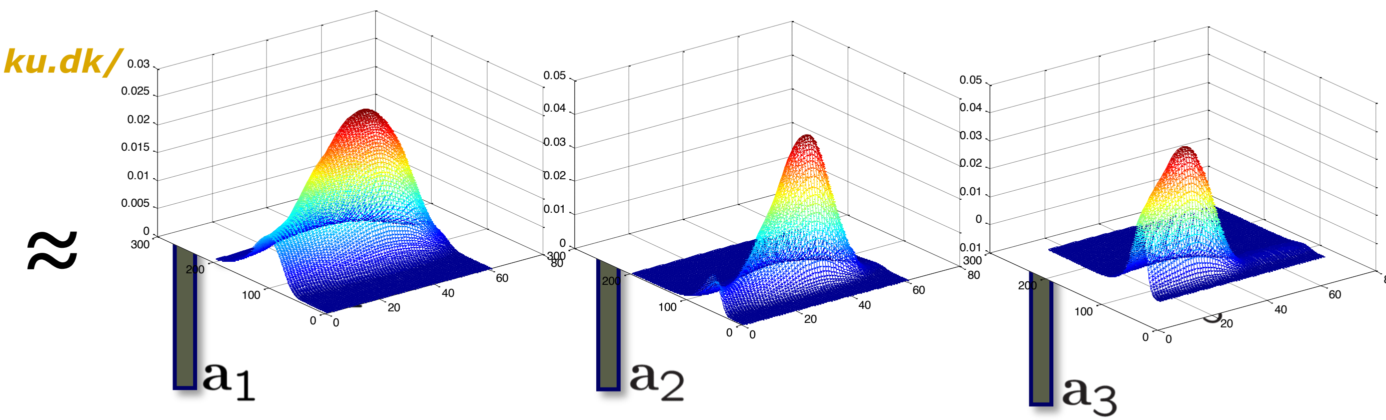
Chemometrics: CP can separate the chemicals in mixtures

Tool: Tensor Factorizations **[Andersen and Bro, 2003]**

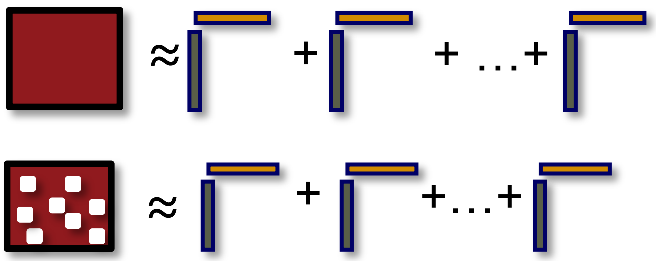
A popular application of the CP model is the separation of individual chemicals from mixtures of chemicals measured using fluorescence spectroscopy.



Amino Acid Data
http://www.models.life.ku.dk/



Summary

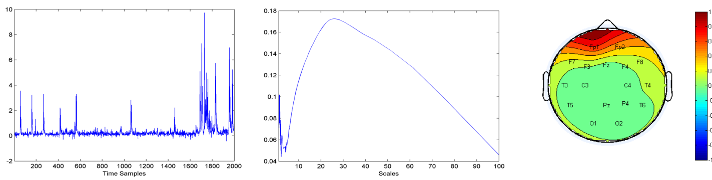


Matrix Factorizations: How to use matrix factorizations to
(i) find the underlying factors,
(ii) predict missing entries

Matrix factorizations without constraints are not unique.

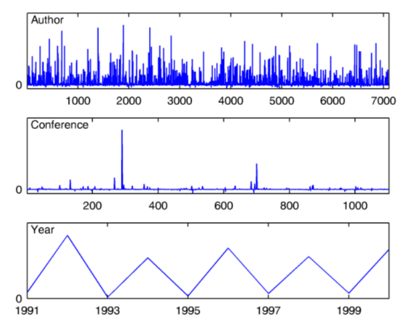
Constraints need to make sense in terms of the application; otherwise, the model will not reveal what we are looking for.

Tensor Factorizations



Tensor factorization methods such as CP have better uniqueness properties compared to matrix factorizations.

How to use tensor factorizations, in particular, CP, to find the underlying factors, e.g., chemometrics, neuroscience, temporal link prediction



Details, algorithms and more applications tomorrow!