Factorizing Gigantic Matrices

Part I: The Basics

Christian Bauckhage, Kristian Kersting, and Christian Thurau

universität bonn

Fraunhofer IAIS
For Starters . . .

observations:

- **data** matrix factorization has become an important tool of the trade in information retrieval, data mining, and pattern recognition
For Starters . . .

observations:

• *data* matrix factorization has become an important tool of the trade in information retrieval, data mining, and pattern recognition

• nowadays, typical data matrices are huge
For Starters . . .

observations:

- **data** matrix factorization has become an important tool of the trade in information retrieval, data mining, and pattern recognition

- nowadays, typical data matrices are **huge**

- examples include:
  - gene expression data and microarrays
  - (collections of) digital images
  - term by document matrices
  - graph adjacency matrices
For Starters . . .

observations:

- **data** matrix factorization has become an important tool of the trade in information retrieval, data mining, and pattern recognition

- nowadays, typical data matrices are huge

- examples include:
  - gene expression data and microarrays
  - (collections of) digital images
  - term by document matrices
  - graph adjacency matrices

- this tutorial is about *new approaches* to matrix factorization
For Starters . . .

observations:

• *data* matrix factorization has become an important tool of the trade in information retrieval, data mining, and pattern recognition

• nowadays, typical data matrices are huge

• examples include:
  - gene expression data and microarrays
  - (collections of) digital images
  - term by document matrices
  - graph adjacency matrices

• this tutorial is about *new approaches* to matrix factorization

• our focus is on *geometry* rather than on optimization
Outline

Setting The Stage

Basic Terms and Concepts

SVD and PCA

Non-Negative Matrix Factorization

Archetypal Analysis

K-Means

Summary and Outlook on Part II
Outline

Setting The Stage

Basic Terms and Concepts

SVD and PCA

Non-Negative Matrix Factorization

Archetypal Analysis

K-Means

Summary and Outlook on Part II
matrix factorization:
- given a matrix
  \[ V \]
- determine matrices
  \[ W \] \text{ and } \[ H \]
such that
  \[ V = WH \]
Setting The Stage

matrix factorization:
- given a matrix $V$
- determine matrices $W$ and $H$
  such that $V = WH$ or $V \approx WH$
Setting The Stage

⚠️ note:

- characteristics such as
  - entries
  - shape
  - rank

of $V$, $W$, and $H$ will depend on application context
example (1):

- solving systems of linear equations

\[
\begin{align*}
 a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
& \quad \vdots \\
a_{n1}x_1 + a_{m2}x_2 + \ldots + a_{nn}x_n &= b_n
\end{align*}
\]

- i.e., for \( A \in \mathbb{C}^{n \times n} \) and \( b \in \mathbb{C}^n \), determine \( x \in \mathbb{C}^n \) such that

\[
A x = b
\]
example (1):
  • possible approach: *QR decomposition*

\[ A x = b \iff QR x = b \iff R x = Q^\dagger b \]

where

\[ Q \in \mathbb{C}^{n \times n} \text{ (unitary)} \]
\[ R \in \mathbb{C}^{n \times n} \text{ (upper triangular)} \]
example (1):

- possible approach: **QR decomposition**

\[ A \mathbf{x} = \mathbf{b} \iff QR \mathbf{x} = \mathbf{b} \iff R \mathbf{x} = Q^\dagger \mathbf{b} \]

where

\[
Q \in \mathbb{C}^{n\times n} \text{ (unitary)}
\]

\[
R \in \mathbb{C}^{n\times n} \text{ (upper triangular)}
\]

- then

\[
\begin{bmatrix}
    r_{11} & r_{12} & \cdots & r_{1n} \\
    r_{22} & \ddots & \cdots & r_{2n} \\
    \vdots & \ddots & \ddots & \vdots \\
    r_{nn} & \cdots & \cdots & r_{nn}
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_n
\end{bmatrix}
= 
\begin{bmatrix}
    c_1 \\
    c_2 \\
    \vdots \\
    c_n
\end{bmatrix}
\]

s.t.

\[
x_i = \frac{1}{r_{ii}} \left( c_i - \sum_{k=i+1}^{n} r_{ik} x_k \right) \quad \text{where} \quad x_n = \frac{c_n}{r_{nn}}
\]
Setting The Stage

⚠️ note:

- the QR decomposition can be computed in $O(n^3)$ using
  - Gram-Schmidt orthonormalization
  - Householder reflections
  - Givens rotations

we shall *not* elaborate on these
Setting The Stage

⚠️ note:

- the QR decomposition can be computed in $O(n^3)$ using
  - Gram-Schmidt orthonormalization
  - Householder reflections
  - Givens rotations

we shall not elaborate on these

⚠️ note:

- we just saw an example involving matrices over $\mathbb{C}$
- from now on, we focus on matrices over $\mathbb{R}$
example (2):
- low-rank approximation of \( V \), i.e. factorize

\[
V \approx WH
\]

where

\[
\begin{align*}
V & \in \mathbb{R}^{m \times n} \\
W & \in \mathbb{R}^{m \times k} \\
H & \in \mathbb{R}^{k \times n}
\end{align*}
\]

and \( k \ll \min\{m, n\} \)
Setting The Stage

matrix factorization allows for:

- solving linear equations
- transforming data
- compressing data

matrix factorization facilitates subsequent processing in:

- information retrieval
- pattern recognition
- data mining
Setting The Stage

matrix factorization allows for:

• solving linear equations
• transforming data
• compressing data

matrix factorization facilitates subsequent processing in:

• information retrieval
• pattern recognition
• data mining
Outline

Setting The Stage

Basic Terms and Concepts

SVD and PCA

Non-Negative Matrix Factorization

Archetypal Analysis

K-Means

Summary and Outlook on Part II
Basic Terms and Concepts

notation:

\[ \mathbf{v} \leftrightarrow \text{a vector} \]
\[ \mathbf{v} \in \mathbb{R}^n \leftrightarrow \text{a real-valued } n \text{ vector} \]
\[ \mathbf{v}_k \leftrightarrow \text{a vector element} \]
\[ \mathbf{0} \leftrightarrow \text{vector of all zeros} \]
\[ \mathbf{1} \leftrightarrow \text{vector of all ones} \]
\[ \mathbf{v} \succeq \mathbf{0} \leftrightarrow \mathbf{v}_k \geq 0 \ \forall \ k \]

\[ \mathbf{M} \leftrightarrow \text{a matrix} \]
\[ \mathbf{M} \in \mathbb{R}^{m \times n} \leftrightarrow \text{a real-valued } m \times n \text{ matrix} \]
\[ m_{ij} \leftrightarrow \text{a matrix element} \]
\[ \mathbf{c}_j \leftrightarrow \text{the } j\text{th column of } \mathbf{M} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n] \]
\[ \mathbf{r}^j \leftrightarrow \text{the } j\text{th row of } \mathbf{M} = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_m]^T \]
Basic Terms and Concepts

The Jargon

*inner product* of $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

$$z = \langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y} = \sum_{k=1}^{n} x_k y_k$$
Basic Terms and Concepts

The Jargon

**inner product** of \( x, y \in \mathbb{R}^n \)

\[
z = \langle x, y \rangle = x^T y = \sum_{k=1}^{n} x_k y_k
\]

**inner product** of \( A, B \in \mathbb{R}^{m \times n} \)

\[
c = \langle A, B \rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} b_{ij} = tr(A^T B) = tr(AB^T)
\]
Basic Terms and Concepts

The Jargon

inner product of \( x, y \in \mathbb{R}^n \)

\[ z = \langle x, y \rangle = x^T y = \sum_{k=1}^{n} x_k y_k \]

inner product of \( A, B \in \mathbb{R}^{m \times n} \)

\[ c = \langle A, B \rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} b_{ij} = \text{tr}(A^T B) = \text{tr}(AB^T) \]

outer product of \( x \in \mathbb{R}^m, y \in \mathbb{R}^n \)

\[ Z = x \otimes y = xy^T \iff z_{ij} = x_i y_j \]
Basic Terms and Concepts

Interestingly:

- if

\[ V = WH \]

then

\[ v_{ij} = \langle w^i, h_j \rangle \]
Basic Terms and Concepts

*Euclidean norm* of \( \mathbf{x} \in \mathbb{R}^n \)

\[
\| \mathbf{x} \| = \langle \mathbf{x}, \mathbf{x} \rangle^{\frac{1}{2}} \\
\iff \| \mathbf{x} \|^2 = \langle \mathbf{x}, \mathbf{x} \rangle = \mathbf{x}^T \mathbf{x}
\]
Basic Terms and Concepts

*Euclidean norm* of \( x \in \mathbb{R}^n \)

\[
\|x\| = \langle x, x \rangle^{\frac{1}{2}}
\]

\(\iff\) \( \|x\|^2 = \langle x, x \rangle = x^T x \)

*Euclidean distance* of \( x, y \in \mathbb{R}^n \)

\[
\|x - y\| = \langle x - y, x - y \rangle^{\frac{1}{2}}
\]

\(\iff\) \( \|x - y\|^2 = \langle x - y, x - y \rangle = (x - y)^T (x - y) \)
Basic Terms and Concepts

**Euclidean norm** of \( x \in \mathbb{R}^n \)

\[
\|x\| = \langle x, x \rangle^{\frac{1}{2}}
\]

\( \iff \|x\|^2 = \langle x, x \rangle = x^T x \)

**Euclidean distance** of \( x, y \in \mathbb{R}^n \)

\[
\|x - y\| = \langle x - y, x - y \rangle^{\frac{1}{2}}
\]

\( \iff \|x - y\|^2 = \langle x - y, x - y \rangle = (x - y)^T (x - y) \)

⚠️ note:

\[
\|Ax - b\|^2 = (Ax - b)^T (Ax - b) = (x^T A^T - b^T) (Ax - b)
\]

\[
= x^T A^T Ax - 2b^T Ax + b^T b
\]

\[
= x^T Qx - 2q^T x + b^T b
\]
Basic Terms and Concepts

**Frobenius norm** of $A \in \mathbb{R}^{m \times n}$

$$\|A\|^2 = \left(\sum_{i,j} a_{ij}^2\right)^{\frac{1}{2}} \iff \|A\|^2 = \sum_{i,j} a_{ij}^2$$

**Frobenius distance** of $A, B \in \mathbb{R}^{m \times n}$ correspondingly
Basic Terms and Concepts

Frobenius norm of \( A \in \mathbb{R}^{m \times n} \)

\[
\| A \|^2 = \left( \sum_{i,j} a_{ij}^2 \right)^{\frac{1}{2}} \Leftrightarrow \| A \|^2 = \sum_{i,j} a_{ij}^2
\]

Frobenius distance of \( A, B \in \mathbb{R}^{m \times n} \) correspondingly

⚠️ note:

\[
\| WH - V \|^2 = \| Wh_1 - v_1 \|^2 + \| Wh_2 - v_2 \|^2 + \ldots
\]
Basic Terms and Concepts

\( \mathbf{x} \) is a \textit{unit vector}, if

\[
\|\mathbf{x}\| = \|\mathbf{x}\|^2 = \langle \mathbf{x}, \mathbf{x} \rangle = \mathbf{x}^T \mathbf{x} = 1
\]

\( \mathbf{x} \) is a \textit{stochastic vector}, if

\[
\sum_k x_k = \langle \mathbf{1}, \mathbf{x} \rangle = \mathbf{1}^T \mathbf{x} = 1
\]

\( \mathbf{x}_i \) and \( \mathbf{x}_j \) are \textit{orthonormal}, if

\[
\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \delta_{ij}
\]
Basic Terms and Concepts

A finite set of vectors \( \{ \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \} \) is \textit{linearly independent}, if

\[
0 = a_1 \mathbf{v}_1 + a_2 \mathbf{v}_2 + \ldots + a_n \mathbf{v}_n
\]

only admits the trivial solution

\[
a_1, a_2, \ldots, a_n = 0
\]
Basic Terms and Concepts

A finite set of vectors \( \{v_1, v_2, \ldots, v_n\} \) is linearly independent, if

\[
0 = a_1 v_1 + a_2 v_2 + \ldots + a_n v_n
\]

only admits the trivial solution

\[
a_1, a_2, \ldots, a_n = 0
\]

If \( V = \{v_1, v_2, \ldots, v_n\} \) are linearly independent vectors in \( \mathbb{R}^n \), the linear combinations

\[
v = v_1 v_1 + v_2 v_2 + \ldots + v_n v_n
\]

span the space \( \mathbb{R}^n \), \( V \) is a basis of \( \mathbb{R}^n \), and the coefficients \( v_i \) are the coordinates of \( v = [v_1, v_2, \ldots, v_n]^T \)
Basic Terms and Concepts

interestingly:
• if

\[ V = WH \]

then

\[ v_j = Wh_j = \sum_k h_{kj} w_k \]
Basic Terms and Concepts

interestingly:

• if

\[ V = WH \]

then

\[ \mathbf{v}_j = Wh_j \]
\[ = \sum_k h_{kj} \mathbf{w}_k \]

• therefore

\[ \mathbf{W} \Leftrightarrow \text{“basis” matrix} \]
\[ \mathbf{H} \Leftrightarrow \text{coefficient matrix} \]
Basic Terms and Concepts

the *rank* of \( M \in \mathbb{R}^{m \times n} \) is the number of linearly independent rows or columns of \( M \), i.e. the dimension of its row- and column-space

⚠️ note:

\[
\text{rk}(M) \leq \min\{m, n\} \\
\text{rk}(M) = \text{rk}(M^T)
\]

⚠️ note:

\[
\text{rk}(x \otimes y) = 1
\]
Basic Terms and Concepts
The Geometry of Data

*coordinate system*

⇔ *origin, axes, and unit length*

*cartesian coordinate system*

⇔ *pairwise perpendicular and positively oriented axes; unit length 1*
Basic Terms and Concepts

coordinate transformations:

- **consider**: data point \( p \) in a cartesian coordinate system

![Diagram of coordinate system with point p]
coordinate transformations:

- *consider:* data point \( p \) in a cartesian coordinate system
- *Q1:* do different coordinates yield deeper insights?
coordinate transformations:
- *consider*: data point $p$ in a cartesian coordinate system
- *Q1*: do different coordinates yield deeper insights?
- *Q2*: how to transit to the other system?
we have

\[ p = x p_x + y p_y = u p_u + v p_v \]
Basic Terms and Concepts

we have

\[ \mathbf{p} = \mathbf{x} p_x + \mathbf{y} p_y \]
\[ = \mathbf{u} p_u + \mathbf{v} p_v \]

“dotting” with \( \mathbf{x} \) yields

\[ \langle \mathbf{x}, \mathbf{x} \rangle p_x + \langle \mathbf{x}, \mathbf{y} \rangle p_y = \langle \mathbf{x}, \mathbf{u} \rangle p_u + \langle \mathbf{x}, \mathbf{v} \rangle p_v \]
we have

\[ \mathbf{p} = \mathbf{x} p_x + \mathbf{y} p_y \]
\[ = \mathbf{u} p_u + \mathbf{v} p_v \]

“dotting” with \( \mathbf{x} \) yields

\[ \langle \mathbf{x}, \mathbf{x} \rangle p_x + \langle \mathbf{x}, \mathbf{y} \rangle p_y = \langle \mathbf{x}, \mathbf{u} \rangle p_u + \langle \mathbf{x}, \mathbf{v} \rangle p_v \]

since \( \mathbf{x} \perp \mathbf{y} \) and \( \|\mathbf{x}\| = 1 \), this is tantamount to

\[ p_x = \langle \mathbf{x}, \mathbf{u} \rangle p_u + \langle \mathbf{x}, \mathbf{v} \rangle p_v \]

and similarly

\[ p_y = \langle \mathbf{y}, \mathbf{u} \rangle p_u + \langle \mathbf{y}, \mathbf{v} \rangle p_v \]
Basic Terms and Concepts

in matrix notation

\[
\begin{bmatrix}
  p_x \\ p_y
\end{bmatrix} = 
\begin{bmatrix}
  \langle x, u \rangle & \langle x, v \rangle \\ 
  \langle y, u \rangle & \langle y, v \rangle
\end{bmatrix} \begin{bmatrix}
  p_u \\ p_v
\end{bmatrix}
\]
Basic Terms and Concepts

in matrix notation

\[
\begin{bmatrix}
  p_x \\
  p_y \\
\end{bmatrix} = \begin{bmatrix}
  \langle x, u \rangle & \langle x, v \rangle \\
  \langle y, u \rangle & \langle y, v \rangle \\
\end{bmatrix} \begin{bmatrix}
  p_u \\
  p_v \\
\end{bmatrix} = \begin{bmatrix}
  \cos \varphi & \sin \varphi \\
  -\sin \varphi & \cos \varphi \\
\end{bmatrix} \begin{bmatrix}
  p_u \\
  p_v \\
\end{bmatrix}
\]

since \( x, y, u, \) and \( v \) are unit vectors, we have

\[
\begin{align*}
\langle x, u \rangle &= \|x\| \|u\| \cos \varphi = \cos \varphi \\
\langle x, v \rangle &= \|x\| \|v\| \cos\left(\frac{\pi}{2} - \varphi\right) = \cos\left(\frac{\pi}{2} - \varphi\right) \\
\langle y, u \rangle &= \|y\| \|u\| \cos\left(\frac{\pi}{2} + \varphi\right) = \cos\left(\frac{\pi}{2} + \varphi\right) \\
\langle y, u \rangle &= \|y\| \|u\| \cos \varphi = \cos \varphi
\end{align*}
\]

together with \( \cos(\alpha \pm \beta) = \cos \alpha \cos \beta \mp \sin \alpha \sin \beta \) this yields the above result
Basic Terms and Concepts

in other words:

\[ p_{xy} = R \ p_{uv} \]
in other words:

\[ p_{xy} = R p_{uv} \]

observe:

- given a data matrix \( M_{uv} = [p_{uv}^1, p_{uv}^2, \ldots, p_{uv}^n] \), the transformed data would become

\[ M_{xy} = RM_{uv} \]
Basic Terms and Concepts

in other words:

\[ p_{xy} = R \hat{p}_{uv} \]

observe:

• given a data matrix \( M_{uv} = [p_{uv}^1, p_{uv}^2, \ldots, p_{uv}^n] \), the transformed data would become

\[ M_{xy} = RM_{uv} \]

• in this sense:

matrix factorization \( \Leftrightarrow \) data transformation
Basic Terms and Concepts

⚠️ note:
- $R$ is an orthogonal matrix
Basic Terms and Concepts

⚠️ note:

- $R$ is an **orthogonal matrix**
- it is square
- its columns $c_i$ are orthogonal unit vectors with $\langle c_i, c_j \rangle = \delta_{ij}$
- its rows $r^i$ are orthogonal unit vectors with $\langle r^i, r^j \rangle = \delta_{ij}$
- its transpose equals its inverse

\[ R^T = R^{-1} \]

so that

\[ R^T R = RR^T = I \]
Basic Terms and Concepts

note:

- \( R \) is an *orthogonal matrix*
- it is square
- its columns \( \mathbf{c}_i \) are orthogonal unit vectors with \( \langle \mathbf{c}_i, \mathbf{c}_j \rangle = \delta_{ij} \)
- its rows \( \mathbf{r}^i \) are orthogonal unit vectors with \( \langle \mathbf{r}^i, \mathbf{r}^j \rangle = \delta_{ij} \)
- its transpose equals its inverse
  \[
  R^T = R^{-1}
  \]

so that

\[
R^T R = RR^T = I
\]

- \( R \) is a *rotation matrix*, because additionally \( \det R = \cos^2 \varphi + \sin^2 \varphi = 1 \)
Outline

Setting The Stage

Basic Terms and Concepts

SVD and PCA

Non-Negative Matrix Factorization

Archetypal Analysis

K-Means

Summary and Outlook on Part II
SVD and PCA

**singular value decomposition (SVD)**

- a singularly valuable decomposition
- every real-valued $m \times n$ matrix $A$ can be factorized as

$$A = U \Sigma V^T$$

where

- $U \in \mathbb{R}^{m \times m} \Leftrightarrow$ orthogonal matrix of *left singular vectors*
- $V \in \mathbb{R}^{n \times n} \Leftrightarrow$ orthogonal matrix of *right singular vectors*
- $\Sigma \in \mathbb{R}^{m \times n} \Leftrightarrow$ diagonal matrix of *singular values*
SVD and PCA

**visually**: e.g. for $m \leq n$
SVD and PCA

Note:

\[ A = U \Sigma V^T = \sum_{j=1}^{r} \sigma_j u_j v_j^T \]

where \( r = \text{rk}(A) \)
SVD and PCA

important property:

• the problem

\[
\min_{\text{rk}(Z)=k} \|A - Z\|\]

has solution

\[
Z = A_k = \sum_{j=1}^{k} \sigma_j u_j v_j^T
\]
SVD and PCA

important property:
  • the problem

\[ \min_{\text{rk}(Z)=k} \| A - Z \| \]

has solution

\[ Z = A_k = \sum_{j=1}^{k} \sigma_j u_j v_j^T \]

• the truncation error is

\[ \| A - A_k \|^2 = \sum_{j=k+1}^{r} \sigma_j^2 \]
SVD and PCA

example (1):

- matrix $A \in \mathbb{R}^{256 \times 232}$ where $0 \leq a_{ij} \leq 255$
SVD and PCA

example (1):

• low rank approximations

\[ A \approx A_k = \sum_{j=1}^{k} \sigma_j u_j v_j^T = W_k \Sigma_k V_k^T \]

and residuals where \( k \in \{1, 3, 9, 18, 36, 72, 144\} \)
SVD and PCA

example (2):

• matrix $A \in \mathbb{R}^{248 \times 365}$ where $0 \leq a_{ij} \leq 255$
SVD and PCA

eexample (2):

- low rank approximations

\[ A \approx A_k = \sum_{j=1}^{k} \sigma_j u_j v_j^T = W_k \Sigma_k V_k^T v_j^T \]

and residuals where \( k \in \{1, 3, 9, 18, 36, 72, 144\} \)
SVD and PCA

observe:

- instead of storing $A_k$ as an array of $m \cdot n$ entries, it may be represented using

\[ k \cdot m + k \cdot n + k = k(m + n + 1) \]

values only

- typically, $k(m + n + 1) \ll mn$, if $k \ll \text{rk}(A)$
SVD and PCA

observe:

• instead of storing $A_k$ as an array of $m \cdot n$ entries, it may be represented using

$$k \cdot m + k \cdot n + k = k(m + n + 1)$$

values only

• typically, $k(m + n + 1) \ll mn$, if $k \ll \text{rk}(A)$

• in this sense:

  matrix rank reduction $\Leftrightarrow$ data compression
SVD and PCA

⚠️ note:

- the SVD can be computed in $O(\min\{m^2 n, mn^2\})$
- Monte Carlo approaches and methods for streamed data exist, we shall *not* elaborate on these
SVD and PCA

observe:

\[ A^T A = (U \Sigma V^T) U \Sigma V^T \]
\[ = V \Sigma^T U^T U \Sigma V^T \]
\[ = V \Sigma^2 V^T \]
\[ = V \Lambda V^T \]

\[ A^T A = U \Sigma V^T (U \Sigma V^T)^T \]
\[ = U \Sigma V^T V \Sigma^T U^T \]
\[ = U \Sigma^2 U^T \]
\[ = U \Lambda U^T \]
SVD and PCA

observe:

\[ A^T A = (U \Sigma V^T) \Sigma V^T \]
\[ = V \Sigma^T U^T \Sigma V^T \]
\[ = V \Sigma^2 V^T \]
\[ = V \Lambda V^T \]

\[ A^T A = U \Sigma V^T (U \Sigma V^T)^T \]
\[ = U \Sigma V^T V \Sigma^T U^T \]
\[ = U \Sigma^2 U^T \]
\[ = U \Lambda U^T \]

note:

• \( A^T A \) and \( A^T A \) are *symmetric*, i.e. they are square and

\[ (A^T A)^T = A^T A \]
\[ (AA^T)^T = AA^T \]
SVD and PCA

*spectral theorem*

- every symmetric matrix $C$ can be decomposed as

$$C = WW^T$$

where

$W \leftrightarrow$ orthogonal matrix of *eigenvectors*

$\Lambda \leftrightarrow$ diagonal matrix of real valued *eigenvalues*
SVD and PCA

**spectral theorem**

- every symmetric matrix $C$ can be decomposed as

$$C = W \Lambda W^T$$

where

$W \iff$ orthogonal matrix of eigenvectors

$\Lambda \iff$ diagonal matrix of real valued eigenvalues

- terminology due to

$$CW = W \Lambda$$

$\iff$ $Cw_j = \lambda_i w_j$
SVD and PCA

applications:

• powers of $A \in \mathbb{R}^{n \times n}$

$$A^k = (W \Lambda W^T)^k = W \Lambda W^T W \Lambda W^T \ldots W \Lambda W^T = W \Lambda^k W^T$$

where

$$\Lambda^k = \begin{bmatrix} \lambda_{11}^k & 0 & \cdots \\ 0 & \lambda_{22}^k & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$
SVD and PCA

applications:

• matrix exponential of $A \in \mathbb{R}^{n \times n}$

$$e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k = W \left( \sum_{k=0}^{\infty} \frac{1}{k!} \Lambda^k \right) W^T = We^\Lambda W^T$$

where

$$e^\Lambda = \begin{bmatrix} e_\lambda^{11} & 0 & \cdots \\ 0 & e_\lambda^{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$
SVD and PCA

applications:
• matrix exponential of \( A \in \mathbb{R}^{n \times n} \)

\[
e^{A} = \sum_{k=0}^{\infty} \frac{1}{k!} A^k = W \left( \sum_{k=0}^{\infty} \frac{1}{k!} \Lambda^k \right) W^T = We^\Lambda W^T
\]

where

\[
e^\Lambda = \begin{bmatrix}
e^{\lambda}_{11} & 0 & \cdots \\
0 & e^{\lambda}_{22} & \cdots \\
\vdots & \vdots & \ddots
\end{bmatrix}
\]

• interesting in the context of graph diffusion kernels
SVD and PCA

applications:
• principal component analysis
principal component analysis (PCA)

- assume data matrix $X$ of zero mean columns

$$X = \begin{bmatrix} x_1, x_2, \ldots, x_n \end{bmatrix} \in \mathbb{R}^{m\times n}$$

- consider data covariance matrix

$$C = \frac{1}{n} \sum_i x_i x_i^T = \frac{1}{n} XX^T \in \mathbb{R}^{m\times m}$$

- compute the solutions to

$$Cw_j = \lambda_j w_j$$

- transform the data according to

$$\hat{X} = W^T X$$
SVD and PCA
The Geometry of PCA

**properties of principal axes:**
- minimal projection error (distance minimization)
- maximal projection spread (variance maximization)
- e.g. for $w_1$, where we assume $\lambda_1 \geq \lambda_2 \geq \ldots$
SVD and PCA

PCA as variance maximization (1):

- consider the \textit{constrained optimization problem}

\[
\max_{\mathbf{w}_1} \frac{1}{n} \sum_i \langle \mathbf{w}_1, \mathbf{x}_i \rangle^2
\]

s.t. \( \langle \mathbf{w}_1, \mathbf{w}_1 \rangle = 1 \)
SVD and PCA

PCA as variance maximization (1):

• consider the *constrained optimization problem*

\[
\max_{w_1} \frac{1}{n} \sum_i \langle w_1, x_i \rangle^2
\]

s.t. \( \langle w_1, w_1 \rangle = 1 \)

• observe that

\[
\langle w_1, x_i \rangle^2 = \langle w_1, x_i \rangle \langle x_i, w_1 \rangle = w_1^T x_i x_i^T w_1
\]
SVD and PCA

PCA as variance maximization (2):

- define

\[ C = \frac{1}{n} \sum_i x_i x_i^T \]

- thus

\[ \frac{1}{n} \sum_i \langle w_1, x_i \rangle^2 = w_1^T C w_1 \]

and the optimization problem reads

\[ \max_{w_1} w_1^T C w_1 \]

s.t. \( w_1^T w_1 = 1 \)
SVD and PCA

PCA as variance maximization (3):

- consider the Lagrangian

\[ L(w_1, \lambda_1) = w_1^T C w_1 - \lambda_1 (w_1^T w_1 - 1) \]
SVD and PCA

PCA as variance maximization (3):

- consider the Lagrangian

\[ L(w_1, \lambda_1) = w_1^T Cw_1 - \lambda_1 (w_1^T w_1 - 1) \]

- determine \( w_1 \) from

\[ \frac{\partial L}{\partial w_1} = 2Cw_1 - 2\lambda_1 w_1 = 0 \]
SVD and PCA

PCA as variance maximization (3):

- consider the Lagrangian

\[ L(w_1, \lambda_1) = w_1^T C w_1 - \lambda_1 (w_1^T w_1 - 1) \]

- determine \( w_1 \) from

\[ \frac{\partial L}{\partial w_1} = 2Cw_1 - 2\lambda_1 w_1 = 0 \]

- which again leads to

\[ Cw_1 = \lambda_1 w_1 \]
PCA for dimensionality reduction (1):

- view data $x_i \in \mathbb{R}^m$ as a linear combination of the $w_j \in \mathbb{R}^m$

$$x_i = x_1^i w_1 + x_2^i w_2 + \ldots + x_d^i w_d = \sum_{j=1}^{m} x_j^i w_j$$

where

$$x_j^i = \langle x_i, w_j \rangle \Leftrightarrow x'_i = W^T x_i$$
SVD and PCA

PCA for dimensionality reduction (2):

• now, determine $1 \leq k < m$, such that

$$x_{ki}^k = \sum_{j=1}^{k} x_{ji} w_j \approx x_i$$
SVD and PCA

PCA for dimensionality reduction (2):

- now, determine $1 \leq k < m$, such that

$$x^k_i = \sum_{j=1}^{k} x^j_i w_j \approx x_i$$

⚠️ note:

- while $x_i \in \mathbb{R}^m$ and $x^k_i \in \mathbb{R}^m$, the coefficient vector $x^{'k}_i \in \mathbb{R}^k$
SVD and PCA

PCA for dimensionality reduction (2): 

- now, determine $1 \leq k < m$, such that

\[
\mathbf{x}_i^k = \sum_{j=1}^{k} \mathbf{x}_j^i \mathbf{w}_j \approx \mathbf{x}_i
\]

\[\text{note:}\]
- while $\mathbf{x}_i \in \mathbb{R}^m$ and $\mathbf{x}_i^k \in \mathbb{R}^m$, the coefficient vector $\mathbf{x}_i^k \in \mathbb{R}^k$
- also

\[
\mathbf{X} \approx \mathbf{W}_k \mathbf{X}'_k
\]
SVD and PCA

PCA for dimensionality reduction (2):

• now, determine $1 \leq k < m$, such that

$$x_i^k = \sum_{j=1}^{k} x_j^i w_j \approx x_i$$

⚠️ note:

• while $x_i \in \mathbb{R}^m$ and $x_i^k \in \mathbb{R}^m$, the coefficient vector $x_i^{'}^k \in \mathbb{R}^k$
• also

$$X \approx W_k X_k^{' k}$$

• in this sense:

matrix rank reduction $\Leftrightarrow$ dimensionality reduction
PCA for dimensionality reduction (3):

- the *average truncation error* is

\[
\epsilon = \frac{1}{n} \sum_{i} \|x_i - x^k_i\|^2 = \frac{1}{n} \sum_{i} \left\| \sum_{j=k+1}^{m} x'_{ij} v_j \right\|^2
\]

\[
= \frac{1}{n} \sum_{i} \sum_{j=k+1}^{m} \langle x_i, v_j \rangle^2 = \sum_{j=k+1}^{m} \langle v_j, Cv_j \rangle
\]

\[
= \sum_{j=k+1}^{m} \lambda_j
\]
PCA for dimensionality reduction (4):

- select $k$ according to the spectrum $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$
SVD and PCA

Example:

(a) projection into space spanned by 1st and 2nd principal axis
(b) projection into space spanned by 1st and 3rd principal axis
(c) projection into space spanned by 2nd and 3rd principal axis
SVD and PCA

Summary

⚠️ **take home message:**

- SVD / PCA are baseline matrix factorization methods in data analysis
- they are concerned with a constrained quadratic optimization problem

\[
\min_{W,H} \| V - WH \|^2
\]

s.t. \( W^T W = I \)

- they may violate non-negativity \( V \geq 0 \)
- reification is tempting but possibly harmful
Outline

Setting The Stage

Basic Terms and Concepts

SVD and PCA

Non-Negative Matrix Factorization

Archetypal Analysis

K-Means

Summary and Outlook on Part II
Non-Negative Matrix Factorization

NMF:

- for \( V \succeq 0 \), determine factorization

\[
V \approx WH
\]

such that

\[
W \succeq 0 \quad \quad H \succeq 0
\]
Non-Negative Matrix Factorization

baseline algorithm (1):

- popularized by D.D. Lee and H.S. Seung, Learning the Parts of Objects by Non-Negative Matrix Factorization, Nature, 401(6755), 1999
- determine $W$ and $H$ by minimizing Frobenius norm

$$||V - WH||^2$$

or divergence measure

$$D(V \parallel WH),$$ where

$$D(A \parallel B) = \sum_{i,j} (a_{ij} \log \frac{a_{ij}}{b_{ij}} - a_{ij} + b_{ij})$$
baseline algorithm (2):

• e.g. for minimizing Frobenius norm

• observe that

\[ \| V - WH \|^2 \]

is convex in either \( W \) or \( H \) but not in \( WH \)

• therefore, minimization problem does not have a closed form solution and suffers from (many) local minima

• hence, randomly initialize \( W, H \geq 0 \) and \ldots \
Non-Negative Matrix Factorization

**baseline algorithm (3):**

- minimize through alternating iterative updates of $W$ and $H$, e.g.:

  $$h_{ij} \leftarrow h_{ij} + \eta_{ij} \left[ (W^T V)_{ij} - (W^T WH)_{ij} \right]$$

- set $\eta_{ij} = \frac{h_{ij}}{(W^T WH)_{ij}}$ to obtain *multiplicative* updates

  $$h_{ij} \leftarrow h_{ij} \frac{(W^T V)_{ij}}{(W^T WH)_{ij}}$$

- and similarly

  $$w_{ik} \leftarrow w_{ik} \frac{(VH^T)_{ik}}{(WHH^T)_{ik}}$$
observe:

- NMF is not unique, since

\[ WH = W AA^{-1} H \]

and

\[ \tilde{W} = WA \quad \tilde{H} = A^{-1} H \]

- decomposition into “parts” because data form a \textit{cone}
Non-Negative Matrix Factorization

Summary

⚠️ take home message (1):

- for \( \mathbf{V} \succeq 0 \), NMF is concerned with a constrained quadratic optimization problem

\[
E = \min_{\mathbf{W}, \mathbf{H}} \| \mathbf{V} - \mathbf{WH} \|^2
\]

\[
\text{s.t. } \mathbf{W} \succeq 0, \quad \mathbf{H} \succeq 0
\]

- a globally optimal solution provably exists; algorithms guaranteed to find it remain elusive; exact NMF is NP hard


take home message (2):

- NMF is a very active area of research
- many approaches have been proposed
  - Bayesian, affine, geometric, online, ...
- practical solutions require at least $O(n^2)$
  with a large constant
Outline

Setting The Stage

Basic Terms and Concepts

SVD and PCA

Non-Negative Matrix Factorization

Archetypal Analysis

K-Means

Summary and Outlook on Part II
Archetypes

- **Plato:**
  ideals and/or pure forms that embody fundamental characteristics of a thing rather than its specific peculiarities

- **C.G. Jung:**
  innate, universal forms (the hero, the great mother, the wise old man, . . .) that channel experiences and emotions, resulting in recognizable and typical behaviors with certain probable outcomes

  
  archetypal analysis ⇔ new way of data analysis for multivariate data
Archetypal Analysis

archetypal analysis (AA):
- assumes a data matrix
  \[ X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m \times n} \]
and considers a constrained optimization problem

\[ \text{RSS}(p) = \min_{A,B} \left\| X - XBA \right\|^2 \]

where

\[ A \in \mathbb{R}^{p \times n}, \; A \succeq 0, \; \sum_{k=1}^{p} a_{kl} = 1, \; l = 1, \ldots, n \]

\[ B \in \mathbb{R}^{n \times p}, \; B \succeq 0, \; \sum_{j=1}^{n} b_{jl} = 1, \; l = 1, \ldots, p \]
Archetypal Analysis

why?

• substituting $Z = XB \in \mathbb{R}^{m \times p}$ (usually $p \ll n$) yields

$$z_k = \sum_{j=1}^{n} x_j b_{jk} \quad \text{and} \quad \| x_i - \sum_{k=1}^{p} z_k a_{ki} \|^2$$

$\Leftrightarrow$ the archetypes $z_k$ are sparse, convex mixtures of the data $x_i$

$\Leftrightarrow$ the data $x_i$ are sparse, convex mixtures of archetypes $z_k$
Archetypal Analysis
The Geometry of AA

**in a nutshell:**
- determine *extreme points* of a data set
- express the data as *convex combinations* of these *basic types*

\[ x_i = Z a_i \]

- note that \( a_i \) is stochastic
Archetypal Analysis

properties:

• archetypes provably reside on the data convex hull

• increasing $p$ approximates the hull

$p = 2$  $p = 3$  $p = 4$  $p = 5$

$p = 6$  $p = 7$  \ldots  convex hull
Archetypal Analysis

properties:

- the coefficients $a_{ki}$ can be thought of as $P(x_i|z_k)$
- (soft)clustering, classification, ranking, …

$k = 4$ means $p = 4$ archetypes simplex projection
Archetypal Analysis

properties:

- coefficient vectors $\mathbf{a}$ can be computed for any $\mathbf{x}$
- AA bridges geometry and probability

\[
P(\mathbf{x}|\mathbf{z}_1) \quad P(\mathbf{x}|\mathbf{z}_2) \quad P(\mathbf{x}|\mathbf{z}_3) \quad P(\mathbf{x}|\mathbf{z}_4) \quad P(\mathbf{x}|\mathbf{z}_5) \quad P(\mathbf{x}|\mathbf{z}_6)
\]
Archetypal Analysis

algorithm according to Cutler and Breiman:
Archetypal Analysis

algorithm according to Cutler and Breiman:

- determine coefficients $a_{ki}$ by solving $n$ constrained problems

$$\min \|Za_i - x_i\|^2 \text{ s.t. } a_{ki} \geq 0 \text{ and } \sum_k a_{ki} = 1$$
Archetypal Analysis

algorithm according to Cutler and Breiman:

• determine coefficients $a_{ki}$ by solving $n$ constrained problems

$$\min \| Za_i - x_i \|^2 \text{ s.t. } a_{ki} \geq 0 \text{ and } \sum_k a_{ki} = 1$$

• given the updated $a_{ki}$, compute intermediate archetypes

$$\tilde{Z} = XA^T (AA^T)^{-1}$$
Archetypal Analysis

algorithm according to Cutler and Breiman:

- determine coefficients $a_{ki}$ by solving $n$ constrained problems
  
  $$\min \| Za_i - x_i \|^2 \text{ s.t. } a_{ki} \geq 0 \text{ and } \sum_k a_{ki} = 1$$

- given the updated $a_{ki}$, compute intermediate archetypes
  
  $$\tilde{Z} = XA^T(AA^T)^{-1}$$

- determine coefficients $b_{jk}$ by solving $p$ constrained problems
  
  $$\min \| Xb_k - \tilde{z}_k \|^2 \text{ s.t. } b_{jk} \geq 0 \text{ and } \sum_j b_{jk} = 1$$
Archetypal Analysis

algorithm according to Cutler and Breiman:

- determine coefficients $a_{ki}$ by solving $n$ constrained problems

$$\min \| Z a_i - x_i \|^2 \text{ s.t. } a_{ki} \geq 0 \text{ and } \sum_k a_{ki} = 1$$

- given the updated $a_{ki}$, compute intermediate archetypes

$$\tilde{Z} = X A^T (A A^T)^{-1}$$

- determine coefficients $b_{jk}$ by solving $p$ constrained problems

$$\min \| X b_k - \tilde{z}_k \|^2 \text{ s.t. } b_{jk} \geq 0 \text{ and } \sum_j b_{jk} = 1$$

- update the archetypes by setting $Z = XB$
Archetypal Analysis algorithm according to Cutler and Breiman:

- determine coefficients $a_{ki}$ by solving $n$ constrained problems
  \[
  \min \| Z a_i - x_i \|^2 \quad \text{s.t. } a_{ki} \geq 0 \text{ and } \sum_k a_{ki} = 1
  \]
- given the updated $a_{ki}$, compute intermediate archetypes
  \[
  \tilde{Z} = X A^T (A A^T)^{-1}
  \]
- determine coefficients $b_{jk}$ by solving $p$ constrained problems
  \[
  \min \| X b_k - \tilde{z}_k \|^2 \quad \text{s.t. } b_{jk} \geq 0 \text{ and } \sum_j b_{jk} = 1
  \]
- update the archetypes by setting $Z = X B$
- compute the new RSS; unless it falls below a threshold or only marginally improves the old RSS, continue at 1.
Archetypal Analysis

analysis:

• in step 1: $i = 1, \ldots, n$ problems involving matrices of size $p^2$
• in step 3: $k = 1, \ldots, p$ problems

$$\min \frac{1}{2} b_k^T R b_k - r^T b_k, \quad R = X^T X \in \mathbb{R}^{n \times n}, \quad r = X^T \tilde{z}_k \in \mathbb{R}^n$$

s.t. $lb_k \geq 0$

$$1^T b_k = 1$$

• recall: $p = \text{number of archetypes}$ and $n = \text{number of data points}$
Archetypal Analysis

analysis:

- in step 1: \( i = 1, \ldots, n \) problems involving matrices of size \( p^2 \)
- in step 3: \( k = 1, \ldots, p \) problems

\[
\min \frac{1}{2} \mathbf{b}_k^T \mathbf{R} \mathbf{b}_k - \mathbf{r}^T \mathbf{b}_k, \quad \mathbf{R} = \mathbf{X}^T \mathbf{X} \in \mathbb{R}^{n \times n}, \quad \mathbf{r} = \mathbf{X}^T \tilde{\mathbf{z}}_k \in \mathbb{R}^n
\]

\[\text{s.t.} \quad \mathbf{1} \mathbf{b}_k \geq 0
\]
\[\mathbf{1}^T \mathbf{b}_k = 1
\]

- recall: \( p = \) number of archetypes and \( n = \) number of data points

\( \Rightarrow \) step 3 involves matrices of size \( n^2 \) and costs dearly
Archetypal Analysis

improvement (I): working sets

• in each iteration, consider $X = X^+ \cup X^-$ where

$$X^- = \{ x_i \in X | x_i = Za_i \}$$
$$X^+ = \{ x_i \in X | x_i \neq Za_i \}$$

• hence $X = [X^+ \ X^-]$ where

$$X^+ \in \mathbb{R}^{m \times n'}$$
$$X^- \in \mathbb{R}^{m \times (n-n')}$$

and $n' < n$
Archetypal Analysis

improvement (I): working sets

- this yields:

\[
\| X - ZA \|^2 = \| [X^+ X^-] - Z [A^+ A^-] \|^2
\]

\[
= \left( \| X^+ - ZA^+ \|^2 + \| X^- - ZA^- \|^2 \right.
\]

\[
\left. \phantom{\| X^+ - ZA^+ \|^2 + \| X^- - ZA^- \|^2} \right\}_{\neq 0} + \left. \phantom{\| X^+ - ZA^+ \|^2 + \| X^- - ZA^- \|^2} \right\}_{= 0}
\]

and with \( Z = XB \), where \( B^- = 0 \), it further reduces to

\[
\| X^+ - ZA^+ \|^2 = \| X^+ - [X^+ X^-] \begin{bmatrix} B^+ \\ B^- \end{bmatrix} A^+ \|^2
\]

\[
= \| X^+ - X^+ B^+ A^+ \|^2
\]
Archetypal Analysis

improvement (I): working sets

- this yields:

\[
\|X - ZA\|^2 = \| [X^+ X^-] - Z [A^+ A^-] \|^2
= \| X^+ - ZA^+ \|^2 + \| X^- - ZA^- \|^2
\]

and with \( Z = XB \), where \( B^- = 0 \), it further reduces to

\[
\|X^+ - ZA^+\|^2 = \|X^+ - [X^+ X^-] \begin{bmatrix} B^+ \\ B^- \end{bmatrix} A^+\|^2
= \|X^+ - X^+ B^+ A^+\|^2
\]

- effort in step 3 reduces to \( O(n'^2) < O(n^2) \)
Archetypal Analysis

**improvement (II): sampling the convex hull**

- archetypes are mixtures of points on the data convex hull
  \[ \Rightarrow \text{restrict algorithm to } X^H \subseteq X \]

- in \( \mathbb{R}^m \), convex hull computation is “expensive” \( (\Theta(n^{(m/2)}) ) \)
  \[ \Rightarrow \text{consider (many) 2D projections of the data and sample the hull} \]
Archetypal Analysis

improvement (II): sampling the convex hull

- every image of a polytope \( P \) under an affine map \( \pi : x \rightarrow Mx + t \) is a polytope
- in particular, every vertex of an affine image of \( P \) corresponds to a vertex of \( P \)
- sampling the hull is “cheap”
- effort is then \( O(n''^2) \ll O(n^2) \)
- \( n'' \) is \( \Omega(\sqrt{\log n}) \)
Archetypal Analysis

Summary

⚠️ take home message:

- AA is concerned with a constrained quadratic optimization problem

\[
E = \min_{W,H} \| V - VWH \|^2
\]

s.t. \( W \) is stochastic
\( H \) is stochastic

- it yields novel and intuitive insights into data
- traditional solution requires \( O(n^2) \) with large constant
- speedup is possible

C. Bauckhage and C. Thurau, Making Archetypal Analysis Practical, Proc. DAGM, 2009

- efficient approximations are possible (more on this later!)
Outline

Setting The Stage

Basic Terms and Concepts

SVD and PCA

Non-Negative Matrix Factorization

Archetypal Analysis

K-Means

Summary and Outlook on Part II
K-Means

**k-means clustering:**

- given a set of data
  \[ \mathcal{X} = \{ \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \}, \quad \mathbf{x}_i \in \mathbb{R}^m \]

and an initial set of *cluster centers*

\[ \mathcal{M} = \{ \mathbf{m}_1^{(1)}, \mathbf{m}_2^{(1)}, \ldots, \mathbf{m}_k^{(1)} \}, \quad \mathbf{m}_j \in \mathbb{R}^m, \quad k \ll n \]
K-Means

**k-means clustering:**

- given a set of data
  \[ \mathcal{X} = \{ \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \}, \quad \mathbf{x}_i \in \mathbb{R}^m \]

and an initial set of *cluster centers*

\[ \mathcal{M} = \{ \mathbf{m}_1^{(1)}, \mathbf{m}_2^{(1)}, \ldots, \mathbf{m}_k^{(1)} \}, \quad \mathbf{m}_j \in \mathbb{R}^m, \quad k \ll n \]

- perform iterative expectation maximization according to

\[ S_j^{(t)} = \{ \mathbf{x}_i \mid \| \mathbf{x}_i - \mathbf{m}_j^{(t)} \| \leq \{ \mathbf{x}_i \mid \| \mathbf{x}_i - \mathbf{m}_h^{(t)} \| \forall h = 1, \ldots, k \} \]

\[ \mathbf{m}_j^{(t+1)} = w_j \sum_{\mathbf{x}_i \in S_j^{(t)}} \mathbf{x}_i, \quad w_j = \frac{1}{|S_j^{(t)}|} \]
K-Means

**k-means clustering:**

- given a set of data
  \[ \mathcal{X} = \{ \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \}, \quad \mathbf{x}_i \in \mathbb{R}^m \]

and an initial set of *cluster centers*

\[ \mathcal{M} = \{ \mathbf{m}_1^{(1)}, \mathbf{m}_2^{(1)}, \ldots, \mathbf{m}_k^{(1)} \}, \quad \mathbf{m}_j \in \mathbb{R}^m, \quad k \ll n \]

- perform iterative expectation maximization according to
  \[ S_j^{(t)} = \{ \mathbf{x}_i \mid \| \mathbf{x}_i - \mathbf{m}_j^{(t)} \| \leq \{ \mathbf{x}_i \mid \| \mathbf{x}_i - \mathbf{m}_h^{(t)} \| \forall h = 1, \ldots, k \} \]
  \[ \mathbf{m}_j^{(t+1)} = w_j \sum_{\mathbf{x}_i \in S_j^{(t)}} \mathbf{x}_i, \quad w_j = \frac{1}{|S_j^{(t)}|} \]

- once the *m* subscripts have converged to local modes, \( \mathbf{x}_i \in S_j \rightarrow \mathbf{m}_j \)
K-Means

k-means clustering is matrix factorization!
k-means clustering is matrix factorization!

• consider, for instance, the product

\[
\begin{pmatrix}
  x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\
  x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\
  x_{31} & x_{32} & x_{33} & \cdots & x_{3n} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  x_{m1} & x_{m2} & x_{m3} & \cdots & x_{mn}
\end{pmatrix}
\begin{pmatrix}
  0 & w_2 & w_3 \\
  w_1 & 0 & w_3 \\
  w_1 & w_2 & 0 \\
  0 & w_2 & 0 \\
  w_1 & 0 & 0 \\
  \vdots & \vdots & \vdots \\
  \vdots & \vdots & \vdots \\
  0 & 0 & 0 \\
  \vdots & \vdots & \vdots \\
  0 & 0 & 0 \\
  0 & 1 & \cdots & 0 \\
\end{pmatrix}
\begin{pmatrix}
  \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \end{bmatrix} \\
  \begin{bmatrix} 0 & 0 & 1 & \cdots & 0 \end{bmatrix}
\end{pmatrix}
\]
K-Means

k-means clustering is matrix factorization!

- consider, for instance, the product

\[
\begin{bmatrix}
  x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\
  x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\
  x_{31} & x_{32} & x_{33} & \cdots & x_{3n} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  x_{m1} & x_{m2} & x_{m3} & \cdots & x_{mn}
\end{bmatrix}
\begin{bmatrix}
  0 & w_2 & w_3 \\
  w_1 & 0 & w_3 \\
  w_1 & w_2 & 0 \\
  0 & w_2 & 0 \\
  w_1 & 0 & 0 \\
  \vdots & \vdots & \vdots \\
  \vdots & \vdots & \vdots \\
  0 & 0 & w_3
\end{bmatrix}
\begin{bmatrix}
  1 & 1 & 0 & \cdots & 0 \\
  0 & 0 & 0 & \cdots & 1 \\
  0 & 0 & 1 & \cdots & 0
\end{bmatrix}
\]

- i.e. for \( X \in \mathbb{R}^{m \times n} \) and \( B \in \mathbb{R}^{n \times 3} \) and \( A \in \mathbb{R}^{3 \times n} \) as above, the product

\[
XBA = MA
\]

realizes an assignment

\[
x_i \rightarrow m_j, \quad \text{where } m_j = Xb_j
\]
K-Means
The Geometry of K-Means

k-means vs. AA:

![Data Basis vector](chart1.png)  ![Data Basis vector](chart2.png)
K-Means
Summary

⚠️ **take home message:**

- k-means is concerned with a constrained quadratic optimization problem

\[ E = \min_{W,H} \| V - VH \|^2 \]

s.t.  \( W \) is stochastic

\( H \) is binary

- the EM algorithm for k-means tacitly assumes Gaussianity
- finding an optimal k-means partitioning of data is NP hard in general
Outline

Setting The Stage

Basic Terms and Concepts

SVD and PCA

Non-Negative Matrix Factorization

Archetypal Analysis

K-Means

Summary and Outlook on Part II
Summary and Outlook on Part II
Matrix Factorization Variants

PCA:

\[ E = \min_{W,H} \| V - WH \|^2 \]

s.t. \( W^T W = I \)

NMF:

\[ E = \min_{W,H} \| V - WH \|^2 \]

s.t. \( W \succeq 0 \)
\[ H \succeq 0 \]

\( k \)-means:

\[ E = \min_{W,H} \| V - VWH \|^2 \]

s.t. \( W \) is stochastic
\( H \) is binary

AA:

\[ E = \min_{W,H} \| V - VWH \|^2 \]

s.t. \( W \) is stochastic
\( H \) is stochastic
Summary and Outlook on Part II

data (sample):

PCA:

NMF:

k-means:

AA:
Summary and Outlook on Part II
Web-Scale Matrix Factorization

observations:

• data may not fit into memory
• methods might not scale ($O(n^2)$ is fine only for small $n$)
• multi-core systems are not always helpful!

remedy (I): hierarchical data formats (HDF5):

• storage of very large data sets on a single machine
• efficient data compression and handling

remedy (II): map-reduce:

• efficient, distributed storage and handling of large data
Practical issues: very large datasets

- Often data does not fit into memory
- Multi-core systems are not always helpful!
- Methods might not scale
- $O(n^2)$ is fine for small $n$...
Practical issues: very large datasets
How to store/handle data?

Hierarchical Data Format (HDF5)

- Storage of very large datasets on a single machine
- Efficient data compression and handling
- Many APIs, e.g. pytables, h5py, ...
Practical issues: very large datasets II
How to store/handle data?

- Efficient, distributed storage and handling of large data
- Implementation of MapReduce
- Support for various languages: Java, Python, C, ...
Practical issues: very large datasets

Bottlenecks

- Bottleneck for many data analysis methods is data transfer from memory/hard disk to CPU
- Each sample has to be loaded once
- Operations rather cheap (multiple cores do not always help!)
- Distributed file-systems often more efficient
Algorithmic Solutions

Three main solutions for handling gigantic matrices:

- **Parallelization**
  - Usually great for scaling algorithms!
  - Sometimes difficult to implement
  - For $O(n^2)$ type algorithms we might run out of machines

- **Sampling/randomized methods**
  - Ignores the vast majority of data
  - Not everyone likes random results

- **More efficient non-randomized approximations**
  - Far from trivial, sometimes impossible
  - Often not as good as the original algorithm
CUR matrix decomposition [Mahoney et al.(2009)]

- **Goal:** SVD-like decomposition of large matrices that works with sparse data and emphasizes interpretability
- **Simple idea:** select number of rows and columns from a matrix s.t. it can be accurately approximated
- **Difficult combinatorial problem:** optimal solution $O(m^c)$ (or $O(n^r)$)

**Algorithm 1:** Main steps of CGR/CUR

**Input:** Matrix $X \in \mathbb{R}^{m \times n}$, integer $c, r$

1. Select $c$ columns from $X$ and construct $C \in \mathbb{R}^{m \times c}$;
2. Select $r$ columns from $X^T$ and construct $R \in \mathbb{R}^{r \times n}$;
3. $U = C^+XR^+$, where $C^+$, $R^+$ denote the Moore-Penrose generalized inverse of the matrices $C$ and $R$;
4. **Return** $C \in \mathbb{R}^{m \times c}$, $U \in \mathbb{R}^{c \times r}$, $R \in \mathbb{R}^{r \times n}$
CUR selection strategies
Euclidean norm

- What makes a good row/column selection?
- **Intuitive idea:** larger norm rows/columns contribute more to the Frobenius norm

**Algorithm 2:** CUR with importance sampling based on the Euclidean-norm of columns.

**Input:** Matrix $X \in \mathbb{R}^{m \times n}$, integer $c$

1. **for** $x = 1 \ldots n$ **do**
   2. \[ P(x) = \sum_i X^2_{i,x} / \sum_i \sum_j X^2_{i,j}; \]
   3. **for** $i = 1 \ldots c$ **do**
   4. Randomly select $W_{*,i} \in X$ based on $P(x)$
5. **Return** $W \in \mathbb{R}^{m \times c}$
CUR selection strategies
Statistical leverage

- Select rows/columns that exhibit a high *statistical leverage*
- **But:** requires computation of the top-\(k\) singular vectors

**Algorithm 3:** CUR with importance sampling based on the Euclidean-norm of columns.

**Input:** Matrix \(\mathbf{X} \in \mathbb{R}^{m \times n}\), integer \(c\)

1. **for** \(x = 1 \ldots n\) **do**
   2. \(P(x) = \frac{1}{k} \sum_{\xi=1}^{k} (v_{X}^{\xi})^2\), where
   3. \(v_{X}^{\xi}\) is the \(x\)-th coordinate of the \(\xi\)-th right singular vector
4. **for** \(i = 1 \ldots c\) **do**
   5. Randomly select \(\mathbf{W}_{*,i} \in \mathbf{X}\) based on \(P(x)\)
6. **Return** \(\mathbf{W} \in \mathbb{R}^{m \times c}\)
Maximizing matrix volumes
[Çivril(2009), Goreinov et al.(2001), Frieze et al.(2004)]

- We define a *good* subset as one maximizing the volume of the parallelepiped (the value of the determinant $\det(C)$) spanned by the columns of $C$.
- Criterion is referred to as maximum volume (Max-Vol) criterion.
- Given a matrix $X^{m \times n}$, we select $c$ of its columns s.t. the volume $\text{Vol}(C^{m \times c}) = |\det C|$ is maximized, where $C^{m \times c}$ contains the selected columns.
CUR selection strategies
Greedy maximum volume [Çivril(2009)]

- **Idea:** greedily select columns that maximize the volume
- Simple greedy approach that removes the projection of selected columns from $X$ (adapted from [Çivril(2009)])

### Algorithm 4: Greedy Max. Vol.

**Input:** Matrix $X \in \mathbb{R}^{m \times n}$, integer $c$

1. $X_1 := X$ // Start with the original data-matrix
2. for $i = 1 \ldots c$ do
   - // Selected $i$-th column $W_i$
   3. $W_{*,i} = \arg\max_j \|v_j\|, X_j \in V_i$;
   - // Remove projection of $W_{*,i}$ from $X_i$
   4. $X_{i+1} = X_i - \text{proj}(W^T_{*,i}, X_i)$
5. Return $W \in \mathbb{R}^{m \times c}$
CUR selection strategies
Greedy maximum volume
CUR selection strategies
Fastmap [Faloutsos et al.(1995)]

- Similar to Greedy-Maximum-Volume, but uses distances
- Iteratively selects pairs of columns

**Algorithm 5**: Fastmap based selection

**Input**: Matrix $X \in \mathbb{R}^{m \times n}$, integer $c$

1. $Y := X$
2. for $i = 1 \ldots c$ (step = 2) do
3.     for $j = 1 \ldots n$ do
4.         choose random column $Y_{*,a} \in Y$;
5.         let $Y_{*,b}$ be column that is farthest apart from $Y_{*,a}$;
6.         let $Y_{*,a}$ be column that is farthest apart from $Y_{*,b}$;
7.         $W_{*,i} = X_{*,a}$;
8.         $W_{*,i+1} = X_{*,b}$;
9.     project $Y$ on line $(Y_{*,a}, Y_{*,b})$
3. Return $W \in \mathbb{R}^{m \times c}$
Simplex Volume Maximization (SIVM) [Thurau et al. (2010)]

- **Problem:** Projections in Greedy, and Fastmap are time consuming
- **Idea:** approximate projections by employing iterative distance computations that maximize the simplex volume
- Similar to
  - Fastmap as it operates on distances
  - Greedy-Maximum-Volume as it directly optimizes the matrix/simplex volume
- but no explicit projections
Let \( C \) be a subset of \( k \) columns from \( X \).

Then \( \Delta(C) \) denotes the \( k \)-dimensional simplex where the volume of the simplex is given by

\[
\text{Vol}(\Delta(C))^2_k = \theta \det(A), \quad \text{with } \theta = \frac{-1^{k+1}}{2^k (k!)^2}, \quad \text{with (1)}
\]

\[
\det(A) = \begin{vmatrix}
0 & 1 & 1 & 1 & \ldots & 1 \\
1 & 0 & d_{1,2}^2 & d_{1,3}^2 & \ldots & d_{1,k}^2 \\
1 & d_{2,1}^2 & 0 & d_{2,3}^2 & \ldots & d_{2,k}^2 \\
1 & d_{3,1}^2 & d_{3,2}^2 & 0 & \ldots & d_{3,k}^2 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
1 & d_{k,1}^2 & d_{k,2}^2 & d_{k,3}^2 & \ldots & 0
\end{vmatrix}
\]

where \( d_{i,j}^2 \) denotes squared distance between column \( i \) and \( j \) and \( \det(A) \) is the \textit{Cayley-Menger} determinant.
If a number of vertices has already been acquired in a sequential manner, the following was recently proven assuming that the vertices are equidistant:

Let $S$ be an $(n - 1)$-simplex. Suppose that its vertices $w_1, \ldots, w_n$ are equidistant and that this distance is $a$. Also, suppose that the distances between vertex $w_{n+1}$ and the other vertices are given by $\{d_{i,n+1}, \ldots, d_{n,n+1}\}$, then the volume of $S$ is determined by

$$
\text{Vol}(S)_n^2 = \frac{a^{2n}}{2^n(n!)^2} \left( \frac{2}{a^4} \sum_{i=1}^{n} d_{i,n+1}^2 d_{j,n+1}^2 + \frac{2}{a^2} \sum_{i=1}^{n} d_{i,n+1}^2 - \frac{n-1}{a^4} \sum_{i=1}^{n} d_{i,n+1}^4 - n + 1 \right).
$$
Equidistant edge lengths $a$ for the first $k$ vertices

For most data this is seldom the case – use of logarithmic scales considerably reduces the introduced error as it maps large distances to similar values and is numerically more stable

From this, an iterative update procedure can be derived:

$$
\mathbf{v}_\pi = \arg \max_k \left( \log(a) \sum_{i=1}^{n} \log(d_{i,k}) + \sum_{j=i+1}^{n} \log(d_{i,k}) \log(d_{j,k}) - \frac{n-1}{2} \sum_{i=1}^{n} \log^2(d_{i,k}) \right).
$$
SIVM example: step 1
SIVM example: step II
SIVM example: step III
SIVM example: step IV
SIVM example: step V
Algorithm 6: Large-scale Deterministic CUR

**Input:** Matrix $X \in \mathbb{R}^{m \times n}$, integer $k$

1. for $j = 1 \ldots n$ do
   2. $n_j \leftarrow \log(||X_{*,j}||)$;
   3. $\Phi_{0,j} \leftarrow n_j$;
   4. $\Lambda_{0,j} \leftarrow n_j^2$;
   5. $\Psi_{0,j} \leftarrow 0$;
   6. select $= \arg \max_j (n_j)$;
   7. $a = n_{\text{select}}$;
   8. $w_1 = X_{*, \text{select}}$;
   9. for $i = 2 \ldots k$ do
      10. for $j = 1 \ldots n$ do
         11. $p_j \leftarrow \log(d(w_{i-1}, X_{*,j}))$;
         12. $\Phi_{i,j} \leftarrow \Phi_{i-1,j} + p_j$;
         13. $\Lambda_{i,j} \leftarrow \Lambda_{i-1,j} + p_j^2$;
         14. $\Psi_{i,j} \leftarrow \Psi_{i-1,j} + p_j \Phi_{i-1}$;
         15. select $= \arg \max_j \left( a \Phi_{i,j} + \Psi_{i,j} - \frac{(i-1)}{2} \Lambda_{i,j} \right)$;
      16. $w_i = X_{*, \text{select}}$;
      17. $W_{*,i} = X_{*, \text{select}}$;
   18. Return $W \in \mathbb{R}^{m \times k}$
Constrained NMF-like factorizations of gigantic matrices

- Most constrained factorizations do not scale well (e.g. Archetypal Analysis, Convex-NMF, ...)
- Possible solution: maximum volume criterion
Assume a data matrix $V$

$$V = [v_1, v_2, \ldots, v_n] \in \mathbb{R}^{m \times n}$$

and consider a constrained optimization

$$RSS = \min_{A,B} \|V - VGH\|^2$$

where

$$H \in \mathbb{R}^{k \times n}, H \succeq 0, \sum_{p=1}^{k} h_{pl} = 1, \quad l = 1, \ldots n$$

$$G \in \mathbb{R}^{n \times k}, G \succeq 0, \sum_{j=1}^{n} g_{jl} = 1, \quad l = 1, \ldots k$$
Convexity constrained matrix factorization II

Why is it interesting?

- Coefficients $\in H$ are stochastic vectors
- (Soft)clustering, ranking, classification, . . .
- Shines a new light on data compared to standard mean-based clustering

convexity constrained

$k$-means
Convexity constrained matrix factorization III

\[ \text{RSS}(p) = \left\| V - VH \right\|^2, \text{ s.t. } h_{k,j} \geq 0, \sum_j h_{k,j} = 1, g_{i,k} \geq 0, \sum_k g_{i,k} = 1 \]  \hspace{1cm} (3)

- Eq. 3 solved for optimal coefficients \(\rightarrow\) alternating least sq. optimization
- **Problem:** solving Eq. 3 computationally demanding
- **Not** applicable for more than \(\approx 5000\) data-points
- **But:** modern data can contain millions/billions of samples

\(\rightarrow\) How to make it scale?
Convexity constrained matrix factorization IV

\[
\text{RSS}(p) = \min_{G,H} \| V - VGH \|^2
\]  

• **Idea:** optimize \( G \) and \( H \) independently

Optimizing \( H \):

• Given \( G \) computation of \( H \) can be done in parallel ✓

Optimizing \( G \):

• \( VG \): basis vectors reside on the data convex hull [Cutler et al.(1994)]

• Compute the convex hull of \( V \) and select appropriate basis vectors ✓

→ **Problem:** computing the convex hull can be an even more difficult problem [2, 1]!
Convex hull subsampling

• Worst case complexity for computing the convex hull is $\Theta(n^{m/2})$ \(^1\)

• Joint estimation of $G/H$ is non convex problem

$\Rightarrow$ Subsampling the convex hull only narrows choice of solutions

---

\(^1\) $n$ points in $m$ dimensions
Convex-hull NMF [Thurau et al.(2009)]

**Observation:** Data points on the convex hull of a *lower dimensional* affine projection also reside on the convex hull in the original space\(^2\)

Selecting Archetype candidates:

1. Compute top-\(k\) eigenvectors
2. Map data to all pairwise projections of the top \(k\)-eigenvectors
3. For each resulting 2D-mapping compute convex hull
4. Perform AA on all found convex hull data points in the original data space

\(^2\)This directly follows the main theorem of polytope theory [Ziegler,“Lectures on Polytopes”. 1995]
SIVM for AA
Maximum volume equivalent

- SIVM also delivers a fast approximation of the convex hull
- Solves convexity constrained factorization in linear time $O(kn)$
- Tested for matrices with billions of entries
- Faster than AA or CH-NMF but less accurate
Summary

• Simple algorithms: don’t be scared by MF in the wild
• All methods available via the Python Matrix Factorization toolbox: pymf.googlecode.com
References


EVALUATION & APPLICATIONS
Rorschach Test

[Hermann Rorschach (* Nov 8, 1884; † April 2 1922)]
Etzioni's Rorschach Test for Computer Scientists
Moore’s Law?
Storage Capacity?
Number of Web Pages?
Number of Facebook Users?
Number of Scientific Publications?
Number of Sensor Readings?
Computing 2020: Science in an Exponential World

“The amount of scientific data is doubling every year.”
[Szalay, Gray; Nature 440, 413-414 (23 March 2006)]

“Real-time data on the whereabouts and behaviors of much of humanity advance behavioral science and offer practical benefits [...]”

“Emerging sensor-equipped computing devices are overcoming longstanding temporal and spatial boundaries to human perception.”

Data is the new oil
Computing 2020: Science in an Exponential World

“The amount of scientific data is doubling every year.”
[Szalay, Gray; *Nature* 440, 413-414 (23 March 2006)]

“Real-time data on the whereabouts and behaviors of much of humanity advance behavioral science and offer practical benefits [...]”

“Emerging sensor-equipped computing devices are overcoming longstanding temporal and spatial boundaries to human perception.”

Data is the new soil
How to make use of millions of shared users’ experiences?
Analysis of Bibliographic Data

- Cumulative publication histograms of 757,368 authors, i.e.,
  - cumulated numbers of publications listed in DBLP in author’s first year, second year, and so on
- To account for histogram nature, use log-transformed data, see also Aitchison (1982)

Good descriptor for her activity and also to some extend for her success (but no implication on impact/quality)

- Senior researcher: likely to have contributed over several years
- PhD students: may not have published many papers.
DBLP (using hierarchical version of CH-NMF)

S.M. Reddy

C.H. Papadimitriou

3%

97%

extremely prolific for more than 40 years

extremely prolific for more than 30 years

senior researcher

senior researcher; constantly publishing over several years

junior researcher; several publication over some years

early career; few publications
Social Media Analysis: Digital Forensics

In a micro-blog, is a single person or a team of authors blogging under a single pseudonym?

- 1.5 million twitter tweets of more than 300 popular twitter users (including tweets from Ashton Kutcher, Barack Obama, Britney Spears, among others)
- From each tweet, compute a set of 30 stylistic features.
  - Ratios between adverbs, verbs, signs, own words, or punctuation signs, … and the length of the tweet
Britney Spears has a “multiple personality” twittering!

short, personal, and emotional vs. marketing and ads
World of WarCraft Activity Profiles

- Online appearance of characters (www.warcraftrealms.com)
- Membership in a guild is exclusive:
  - The selection of a guild influences with whom players frequently interact and how successful players are
- Player level distribution among a guild is a good descriptor
  - a guild of very experienced level 80 characters has a higher chance for achievements than a guild of level 10 players.
  - 150 million votes of 18 million characters belonging to 1.4 million guilds

How do guilds develop over time?

Do game upgrades have impact on guilds?
World-Of-Warcraft

- Constant activity
- Formed before 2nd update, then very active
- Active for the first few levels
- Formed early then disbanded
- Improving, boost after 1st update, no activity after 2nd update
- Seldom active
- Constantly improving till 1st update
World-Of-Warcraft

Game Mining, i.e., using data mining for behavior analytics and in-game marketing in order to maximize online game revenue is considered a multi-million if not billion market.
Because this follows a single Gaussian distribution, Kmeans produces very similar cluster representatives centered around the mean of the Gaussian.
In contrast, cluster representatives computed with max vol constraint describe the diversity well and accommodates human cognition.
80 Million Tiny Google Images (30 billion entries)
80 Million Tiny Google Images (30 billion entries)

Geometric similarity to Walsh filters that are found among the principal components of natural images (Heidemann, 2006)

Time to compute a single basis vector:
- Single core: ~4 hours
- Hadoop: ~6 minutes
Less is more

- Applications to anomaly detection and time-evolving monitoring
  - Essentially, monitor the CUR compression factor over time.

Figure 16: Network flow over time: we can detect anomalies by monitoring the approximation accuracy (b), while traditional method based on traffic volume cannot do (a).

Figure 17: DBLP over time: The approximation accuracy drops slowly as the graphs grow denser.
Randomized vs. deterministic approaches

Results validates on several real-world graphs from [http://snap.stanford.edu/data/index.html](http://snap.stanford.edu/data/index.html)

Deterministic approaches tend to produce lower relative errors

Running times comparable (for a fixed k, we may have to run randomized approach several time!)
CUR for Document Analysis (Term x Docs Matrices)

SVD (top 2 eigenets) + Kmeans

Singular vectors are dense, contain negative entries, and are not easily interpretable

CUR: Projection on the best rank 2 approx. To the supspace spanned by the top five “highest-leverage“ terms
In translational science projects, comprehensible results are essentialy
How to increasing the efficiency and effectiveness of the way we manage and allocate our natural resources?
Historical Climatography Series (HCS) temperature normals for the period between January 1961 and December 2000 for U.S. States. The data consists of monthly averaged temperatures for all U.S. States. Triggered by the four seasons, we computed factorizations using 4 basis vectors.

**Temperature**

- **NMF**: No actual data points. Difficult to interpret. Does it group together US states with (A) hot summers and (B) cold winters?

- **MaxVol**: Actual data points so we can assign month and year. Cover well the four seasons.

**Figure 1:** Anatomy of *daphnia magna* from B. De Samber, L. Vincze et al.

Several organs clearly visible: Eggs, intestine, …

Decomposing biological samples into parts:

Cryofixed *daphnia magna*
Cryofixed Daphnia Magna: Results

Only 6 factors required:

1st: **carapace** facing detector
   → Ca + Sr

2nd: **egg shells / int. structure**
   → Rb + Zn + Fe

3rd: **interior of eggs**
   → Zn + Fe + Br + Rb

4th: **digestive gland + heart + more intestinal tract**
   → Fe + …

5th: **intestinal tract**
   → Zn + Rb + Cu + …

6th: **carapace** facing away from detector
   → Sr
Only 6 factors required:

1st: **carapace** facing detector
   $\Rightarrow$ Ca + Sr

2nd: **egg shells** / **int. structure**
   $\Rightarrow$ Rb + Zn + Fe

3rd: **interior of eggs**
   $\Rightarrow$ Zn + Fe + Br + Rb

4th: **digestive gland** + heart + more intestinal tract
   $\Rightarrow$ Fe + …

5th: **intestinal tract**
   $\Rightarrow$ Zn + Rb + Cu + …

6th: **carapace** facing away from detector

However, approaches have to scale well, too.
Hyperspectral Imaging

Matrices with millions of entries
## Hyperspectral Images

<table>
<thead>
<tr>
<th>Name of AVIRIS Spectral Image</th>
<th>Size</th>
<th># Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indian Pines</td>
<td>220 × (145 × 145)</td>
<td>220 × 21.025</td>
</tr>
<tr>
<td>Cuprite, Nevada</td>
<td>50 × (400 × 350)</td>
<td>50 × 140.000</td>
</tr>
<tr>
<td>Moffett Field, California</td>
<td>56 × (500 × 350)</td>
<td>56 × 175.000</td>
</tr>
<tr>
<td>Jasper Ridge area, California</td>
<td>60 × (600 × 512)</td>
<td>60 × 307.200</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name of Natural Scene Spectral Image</th>
<th>Size</th>
<th># Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sameiro area (Braga)</td>
<td>33 × (1018 × 1339)</td>
<td>33 × 1,363.102</td>
</tr>
<tr>
<td>Ruives (Vieira do Minho)</td>
<td>33 × (1017 × 1338)</td>
<td>33 × 1,360.746</td>
</tr>
<tr>
<td>Museum of the Monastery (Mire de Tribes)</td>
<td>33 × (1018 × 1267)</td>
<td>33 × 1,289.806</td>
</tr>
<tr>
<td>Gualtar campus (University of Minho)</td>
<td>33 × (1019 × 1337)</td>
<td>33 × 1,362.403</td>
</tr>
<tr>
<td>Terras de Bouro (Minho region)</td>
<td>32 × (1020 × 1339)</td>
<td>33 × 1,365.780</td>
</tr>
<tr>
<td>Picoto area (Braga)</td>
<td>33 × (1021 × 1338)</td>
<td>33 × 1,366.098</td>
</tr>
<tr>
<td>Ribeira area (Porto)</td>
<td>33 × (1017 × 1340)</td>
<td>33 × 1,362.780</td>
</tr>
<tr>
<td>Soutu (Minho region)</td>
<td>33 × (1018 × 1340)</td>
<td>33 × 1,364.120</td>
</tr>
</tbody>
</table>
End-Member Detection

- NMF: no actual data points. Quite noisy
- MaxVol: Actual data points. Less noisy
Abundance Maps per End-Member

(a) AVIRIS Indian Pines

(b) Ground Truth

(c) Spectral Band

(a) NMF BV 1

(b) SVM-NMF BV 3

(c) SVM-NMF BV 10

(a) Colored picture of “Terras de Bouro (Minho region)”

(b)

(c)
Hyperspectral Images

Hyperspectral Image 3

Hyperspectral Image 3

Hyperspectral Image 3

Christian Bauckhage, Kristian Kersting, Christian Thurau
Factorizing Gigantic Matrices
The world's population expected to grow from 6.8 billion today to 9.1 billion by 2050.

How to expand agricultural output massively without increasing by much the amount of land used?

Early Detection of Drought Stress using Sensor Technology
Phenotyping: Who is stressed and why?

- Large-scale phenotyping is the natural complement to genome sequencing as a route to rapid advances in biology.
- Ultimately, it links genomics with the performance of plants in the interaction with environmental cues.

Can we detect it earlier using sensor technology?

Barley A

Barley B

Measurement Day (MD) 1  MD 2  MD 3  MD 4  MD 5  MD 6  MD 7

Visual
Hyperspectral Imaging to the Rescue

- Hyperspectral images were taken every 2-3 days
- Each image is a data cube: 640 x 640 x 120 \( \sim \) 50*10^7 data points
- Turning the cube into a matrix of spectra: 4.1*10^6 x 120
- Thus, 10 plants over 7 measurement days: 290*10^6 x 120 \( \sim \) 3.4 billion data points
Hyperspectral Imaging to the Rescue

- Hyperspectral images were taken every 2-3 days
- Each image is a data cube: $640 \times 640 \times 120 \sim 50 \times 10^7$ data points
- Turning the cube into a matrix of spectra: $4.1 \times 10^6 \times 120$
- Thus, 10 plants over 7 measurement days: $290 \times 10^6 \times 120 \sim 3.4 \text{ billion data points}$

Machine Learning and Data Mining to the Rescue

Phenotyping by means of sensor technologies involves the identification of relevant patterns in massive data sets of high dimensional sensor readings with a demanding signal-to-noise ratio.
Machine Learning = Data + Model

Machine’s performance at some task improves with experience

- **Learning**
  - Decision trees, PCA, NMF, k-means, SVMs, …

- **Probability**
  - Bayesian networks, Markov networks, Gaussian Processes …

- **Logic + Learning**
  - Inductive Logic Programming (ILP)

- **Learning + Probability**
  - (Structural) EM, Dynamic Programming, Variational, …

- **Logic + Probability + Learning**
  - SRL, StarAI, …
Machine Learning = Data + Model

Machine’s performance at some task improves with experience

- Decision trees, PCA, NMF, k-means, SVMs, …
- Bayesian networks, Markov networks, Gaussian Processes …
- Inductive Logic Programming (ILP)
- Structural EM, Dynamic Programming, Variational, …
- SRL, StarAI, …

Mission:
Identify patterns of drought stress

Constraints:
- No drought/control labels given
- Non-negative data
- Interpretable patterns
- Scales well to gigantic matrices
The Challenges of Phenotyping

Due to its temporal nature — large data matrices over time — and the demand of physical meaning of the results, plant phenotyping presents unique computational problems in scale and interpretability.

Data-Drive Matrix Factorization to the Rescue!
The Power Of Convex Recombinations

We can estimate distributions!
Dirichlet Distribution

$$\text{Dir}(h_i | \alpha) = B(\alpha) \prod_{j=1}^{C} h_i^j \alpha_j - 1$$

Paves the way to statistical machine learning at massive scale!
Exploratory Plant Phenotyping

1. Compute extreme points $E$ over all plants and day
2. Eliminate background basis vectors by visual inspection
3. Reconstruct each plant for each day in terms of $E$
4. Estimate Dirichlets per plant and day
5. Embed Dirichlets into space and time
Exploratory Plant Phenotyping

1. Compute extreme points \( E \) over all plants and day
2. Eliminate background basis vectors by visual inspection
3. Reconstruct each plant for each day in terms of \( E \)
4. Estimate Dirichlets per plant and day
5. Embed Dirichlets into space and time
Exploratory Plant Phenotyping

1. Compute extreme points \( E \) over all plants and day

2. Eliminate background basis vectors by visual inspection

3. Reconstruct each plant for each day in terms of \( E \)

4. Estimate Dirichlets per plant and day

5. Embed Dirichlets into space and time
Exploratory Plant Phenotyping

1. Compute extreme points $E$ over all plants and day

2. Eliminate background basis vectors by visual inspection

3. Reconstruct each plant for each day in terms of $E$

4. Estimate Dirichlets per plant and day

5. Embed Dirichlets into space and time
Exploratory Plant Phenotyping

1. Compute extreme points $E$ over all plants and day
2. Eliminate background basis vectors by visual inspection
3. Reconstruct each plant for each day in terms of $E$
4. Estimate Dirichlets per plant and day
5. Embed Dirichlets into space and time
Exploratory Plant Phenotyping

1. Compute extreme points $E$ over all plants and day

2. Eliminate background basis vectors by visual inspection

3. Reconstruct each plant for each day in terms of $E$

4. Estimate Dirichlets per plant and day

5. Embed Dirichlets into space and time
Exploratory Plant Phenotyping

1. Compute extreme points $E$ over all plants and day
2. Eliminate background basis vectors by visual inspection
3. Reconstruct each plant for each day in terms of $E$
4. Estimate Dirichlets per plant and day
5. Embed Dirichlets into space and time

Weighted by some distance between Dirichlets

Plant 1 / MD1

Plant 2 / MD1

Plant 1 / MD2

Plant 2 / MD2

Embedding using MDS using the graph distances (similar to ISOMAP)
Exploratory Plant Phenotyping

1. Compute extreme points E over all plants and day

2. Eliminate background basis vectors by visual inspection

3. Reconstruct each plant for each day in terms of E

4. Estimate Dirichlets per plant and day

5. Embed Dirichlets into space and time

No assumption on the data distribution!

Takes about 1.5 days on a single core. Parallelized down to 0.5 hour (80 cores).
Drought Gap

We gain about one week

Interpretable results!

Mainly pigments
Financial Traces

- 1 Basic Materials
- 2 Conglomerates
- 3 Consumer Goods
- 4 Financial
- 5 Healthcare
- 6 Industrial Goods
- 7 Services
- 8 Technology
- 9 Utilities

Iraq War

Financial Crises

2000 - 2011
Lessons learned

- Many modern applications present unique challenges in scaling and interpretability
  - Social networks, fraud detection, biology, computational sustainability, …

- (Data-driven) constrained matrix factorization techniques meet the challenges

Although we have focused on data-driven approaches, the techniques presented may also have an impact on stochastic optimization approaches.
Example: LDA $\approx$ Matrix Factorization

- LDA $\approx$ PLSA $\approx$ NMF $\approx$ Matrix Factorization

So, can we improve LDA by adapting techniques developed for Matrix Factorization?
Latent Dirichlet Allocation

- Fully generative statistical language models
- Posterior approximation based on
  - Sampling
  - Variational parameters
- Variational Bayes inference for LDA
Scaling LDA to massive, even growing datasets

\[ \mathcal{L}(n, \lambda) \triangleq \sum_d \ell(n_d, \gamma(n_d, \lambda), \phi(n_d, \lambda), \lambda) \]

- Online LDA turns the variational approach into a stochastic natural gradient algorithm with per document (mini-batch) update

Idea: Schedule documents for update according to max volume ranking, statistical leverage, norm, … ranking

Schedules (mini-batches of) documents uniformly at random for updates
Influence Scheduled LDA Variants

- Perplexity on 3M Wikipedia corpus (the lower the better)


Influence Scheduled LDA Variants

- Perplexity on 3M Wikipedia corpus (the lower the better)


**Probably the fastest single core LDA approaches**
Faster Topic Convergence

Christian Bauckhage, Kristian Kersting, Christian Thurau
Factorizing Gigantic Matrices
Conclusions

- Most effort in recent years has gone into the modeling part. Massive data sets allow one to ask the data again.
- Particularly appealing given that nowadays (often cheap) sensors produce massive amounts of data.
- Data-driven constrained matrix factorization scales well and yields easy-to-interpret results in a single run (deterministic).
  - Is already fast on a single core machine. The more cores, however, the faster.
- It even enables distribution-"free" embedding of massive (temporal) data.
- Techniques also relevant for stochastic optimization approaches.

Thanks for your attention !!!