Lecture 11: Data representations - Linear methods

Rebecka Jörnsten, Mathematical Sciences

MSA220/MVE441 Statistical Learning for Big Data

29 April 2022

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Goals of data representation

Dimension reduction while retaining important aspects of the data

Goals of data representation

Dimension reduction while retaining important aspects of the data

Goals can be

- Visualisation
- Interpretability/Variable selection
- Data compression
- Finding a representation of the data that is more suitable to the posed question

Goals of data representation

Dimension reduction while retaining important aspects of the data

Goals can be

- Visualisation
- Interpretability/Variable selection
- Data compression
- Finding a representation of the data that is more suitable to the posed question

Let us start with **linear dimension reduction**.

Re-cap: SVD

The **singular value decomposition (SVD)** of a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, $n \geq p$, is

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$$

where $\mathbf{U} \in \mathbb{R}^{n \times p}$ and $\mathbf{V} \in \mathbb{R}^{p \times p}$ with

$$\mathbf{U}^\top\mathbf{U} = \mathbf{I}_p \quad \text{and} \quad \mathbf{V}^\top\mathbf{V} = \mathbf{V}\mathbf{V}^\top = \mathbf{I}_p$$

and $\mathbf{D} \in \mathbb{R}^{p \times p}$ is diagonal.

Usually the diagonal elements of $\mathbf{D}$ are sorted such that

$$d_{11} \geq d_{22} \geq \cdots \geq d_{pp}.$$

SVD and best rank-q-approximation (I)

Write $\mathbf{u}_j$ and $\mathbf{v}_j$ for the columns of $\mathbf{U}$ and $\mathbf{V}$, respectively. Then

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top = \sum_{j=1}^{p} d_{jj} \underbrace{\mathbf{u}_j\mathbf{v}_j^\top}_{\text{rank-1-matrix}}$$

Best rank-q-approximation: For $q < p$

$$\mathbf{X}_q = \sum_{j=1}^{q} d_{jj}\mathbf{u}_j\mathbf{v}_j^\top$$

approximates $\mathbf{X}$ as a **sum of layers** with **approximation error**

$$\left\|\mathbf{X} - \mathbf{X}_q\right\|_F^2 = \left\|\sum_{j=q+1}^{p} d_{jj}\mathbf{u}_j\mathbf{v}_j^\top\right\|_F^2 = \sum_{j=q+1}^{p} d_{jj}^2$$

Alternative view of best rank-q-approximation

Using only the first $q < \min(p, n)$ columns of $\mathbf{V}$ and $\mathbf{U}$, and the first q rows and columns of $\mathbf{D}$, leads to

$$\mathbf{X}_q = \mathbf{U}_q \mathbf{D}_q \mathbf{V}_q^\top.$$

According to the **Eckart-Young-Mirsky theorem**, the matrix $\mathbf{X}_q$ is a solution to the following minimization problem (see website for proof)

$$\underset{\text{rank}(\mathbf{M})=q}{\arg\min} \|\mathbf{X} - \mathbf{M}\|_F^2.$$

The solution is unique if the $q + 1$-th singular value is different from the the q-th singular value.

Alternative view of the Eckart-Young-Mirsky problem

For $q < \min(p, n)$, set $\mathbf{L} := \mathbf{U}_q \mathbf{D}_q \in \mathbb{R}^{n \times q}$ and $\mathbf{F} = \mathbf{V}_q^\top \in \mathbb{R}^{q \times p}$.

Alternative view of the Eckart-Young-Mirsky problem

For $q < \min(p, n)$, set $\mathbf{L} := \mathbf{U}_q\mathbf{D}_q \in \mathbb{R}^{n\times q}$ and $\mathbf{F} = \mathbf{V}_q^\top \in \mathbb{R}^{q\times p}$.

Then $\mathbf{X}_q = \mathbf{LF}$ is a solution of

$$\underset{\mathbf{L}\in\mathbb{R}^{n\times q},\mathbf{F}\in\mathbb{R}^{q\times p}}{\arg\min} ||\mathbf{X} - \mathbf{LF}||_F^2$$

Alternative view of the Eckart-Young-Mirsky problem

For $q < \min(p, n)$, set $\mathbf{L} := \mathbf{U}_q\mathbf{D}_q \in \mathbb{R}^{n\times q}$ and $\mathbf{F} = \mathbf{V}_q^\top \in \mathbb{R}^{q\times p}$.

Then $\mathbf{X}_q = \mathbf{L}\mathbf{F}$ is a solution of

$$\underset{\mathbf{L}\in\mathbb{R}^{n\times q},\mathbf{F}\in\mathbb{R}^{q\times p}}{\arg\min} \|\mathbf{X} - \mathbf{L}\mathbf{F}\|_F^2$$

Notes:

- Whereas $\mathbf{X}_q$ can be the unique minimizer for the original minimisation problem, the matrices $\mathbf{F}$ and $\mathbf{L}$ are not unique.
- **This is just PCA:** When using SVD to compute the PCA of $\mathbf{X}$, then the columns of $\mathbf{V}$ contain the PC directions and the rows of $\mathbf{F}$ the first q of them. Projecting the data onto the PCs but then reconstructing it means to compute $(\mathbf{X}\mathbf{V}_q)\mathbf{V}_q^\top = (\mathbf{U}\mathbf{D}\mathbf{V}^\top\mathbf{V}_q)\mathbf{V}_q^\top = (\mathbf{U}\mathbf{D}\mathbf{I}_{p\times q})\mathbf{V}_q^\top = (\mathbf{U}_q\mathbf{D}_q)\mathbf{V}_q^\top = \mathbf{L}\mathbf{F}$.

Low-rank matrix factorisation

Let $q < \min(p, n)$

$$\underset{\mathbf{L}\in\mathbb{R}^{n\times q},\mathbf{F}\in\mathbb{R}^{q\times p}}{\arg\min} ||\mathbf{X} - \mathbf{L}\mathbf{F}||_F^2$$

Interpretation

- The rows of $\mathbf{F}$ can be seen as **basis vectors** or **coordinates** of a subspace in feature space

Low-rank matrix factorisation

Let $q < \min(p, n)$

$$\underset{\mathbf{L}\in\mathbb{R}^{n\times q},\mathbf{F}\in\mathbb{R}^{q\times p}}{\arg\min} ||\mathbf{X} - \mathbf{LF}||_F^2$$

Interpretation

- The rows of $\mathbf{F}$ can be seen as **basis vectors** or **coordinates** of a subspace in feature space
- The rows of $\mathbf{L}$ provide **coefficients** that combine the basis vectors in $\mathbf{F}$ to the closest q-dimensional approximation of the respective observation

Low-rank matrix factorisation

Let $q < \min(p, n)$

$$\underset{\mathbf{L}\in\mathbb{R}^{n\times q},\mathbf{F}\in\mathbb{R}^{q\times p}}{\arg\min} ||\mathbf{X} - \mathbf{L}\mathbf{F}||_F^2$$

Interpretation

- The rows of $\mathbf{F}$ can be seen as **basis vectors** or **coordinates** of a subspace in feature space
- The rows of $\mathbf{L}$ provide **coefficients** that combine the basis vectors in $\mathbf{F}$ to the closest q-dimensional approximation of the respective observation
- In the framework of **factor analysis** the rows of $\mathbf{F}$ are called **factors** and the rows of $\mathbf{L}$ are called **(latent) loadings**

Notes on factor analysis

- Originated in psychometrics with the idea that factors could describe unobservable (latent) properties (e.g. intelligence)

Notes on factor analysis

- Originated in psychometrics with the idea that factors could describe unobservable (latent) properties (e.g. intelligence)
- A typical assumption is that the rows of $\mathbf{F}$ are orthogonal, i.e. $\mathbf{F}\mathbf{F}^\top = \mathbf{I}_q$

Notes on factor analysis

- Originated in psychometrics with the idea that factors could describe unobservable (latent) properties (e.g. intelligence)
- A typical assumption is that the rows of $\mathbf{F}$ are orthogonal, i.e. $\mathbf{F}\mathbf{F}^\top = \mathbf{I}_q$
- But even row orthogonality of $\mathbf{F}$ does not ensure **identifiability** (uniqueness of the solution) since for a orthogonal matrix $\mathbf{R} \in \mathbb{R}^{q \times q}$

$$\mathbf{L}'\mathbf{F}' := (\mathbf{L}\mathbf{R})(\mathbf{R}^\top\mathbf{F}) = \mathbf{L}\mathbf{F}$$

  and $\mathbf{F}'$ is orthogonal if $\mathbf{F}$ is

Notes on factor analysis

- Originated in psychometrics with the idea that factors could describe unobservable (latent) properties (e.g. intelligence)
- A typical assumption is that the rows of $\mathbf{F}$ are orthogonal, i.e. $\mathbf{F}\mathbf{F}^\top = \mathbf{I}_q$
- But even row orthogonality of $\mathbf{F}$ does not ensure **identifiability** (uniqueness of the solution) since for a orthogonal matrix $\mathbf{R} \in \mathbb{R}^{q \times q}$

  $$\mathbf{L}'\mathbf{F}' := (\mathbf{L}\mathbf{R})(\mathbf{R}^\top\mathbf{F}) = \mathbf{L}\mathbf{F}$$

  and $\mathbf{F}'$ is orthogonal if $\mathbf{F}$ is
- Every orthogonal matrix describes a rotation and when applied to factors and loadings it is called a **factor rotation**

Notes on factor analysis

- Originated in psychometrics with the idea that factors could describe unobservable (latent) properties (e.g. intelligence)
- A typical assumption is that the rows of $\mathbf{F}$ are orthogonal, i.e. $\mathbf{F}\mathbf{F}^\top = \mathbf{I}_q$
- But even row orthogonality of $\mathbf{F}$ does not ensure **identifiability** (uniqueness of the solution) since for a orthogonal matrix $\mathbf{R} \in \mathbb{R}^{q \times q}$

  $$\mathbf{L}'\mathbf{F}' := (\mathbf{L}\mathbf{R})(\mathbf{R}^\top\mathbf{F}) = \mathbf{L}\mathbf{F}$$

  and $\mathbf{F}'$ is orthogonal if $\mathbf{F}$ is
- Every orthogonal matrix describes a rotation and when applied to factors and loadings it is called a **factor rotation**
- Through optimization of $\mathbf{R}$, we can make either factors (**varimax rotation**) or loadings (**quartimax rotation**) sparse

Conclusions from Factor Analysis/SVD-based approach

- The SVD-based approach is provably best in the Frobenius norm

Conclusions from Factor Analysis/SVD-based approach

- The SVD-based approach is provably best in the Frobenius norm
- Best q can be easily chosen by observing the approximation error

Conclusions from Factor Analysis/SVD-based approach

- The SVD-based approach is provably best in the Frobenius norm
- Best q can be easily chosen by observing the approximation error

However:

- Interpretation is difficult since layers both add and subtract information

$$(d_{ii}\mathbf{u}_i\mathbf{v}_i^\top)^{(r,s)} = d_{ii}\mathbf{u}_i^{(r)}\mathbf{v}_i^{(s)}$$

Conclusions from Factor Analysis/SVD-based approach

- The SVD-based approach is provably best in the Frobenius norm
- Best q can be easily chosen by observing the approximation error

However:

- Interpretation is difficult since layers both add and subtract information

$$(d_{ii}\mathbf{u}_i\mathbf{v}_i^\top)^{(r,s)} = d_{ii}\mathbf{u}_i^{(r)}\mathbf{v}_i^{(s)}$$

- $\mathbf{U}$ and $\mathbf{V}$, respectively $\mathbf{L}$ and $\mathbf{F}$, are not unique and usually dense (no zero entries)

Non-negative Matrix Factorization (NMF)

Idea: We can add constraints to the low-rank matrix factorisation problem.

Non-negative matrix factorisation (NMF): Let $q < \min(p, n)$

$$\underset{\mathbf{L}\in\mathbb{R}^{n\times q},\mathbf{F}\in\mathbb{R}^{q\times p}}{\arg\min} ||\mathbf{X}-\mathbf{LF}||_F^2 \quad \text{such that} \quad \mathbf{L}\geq 0,\ \mathbf{F}\geq 0$$

- **Sum of positive layers:** $\mathbf{X} \approx \sum_{j=1}^{q} \mathbf{L}^{(:,j)}\mathbf{F}^{(j,:)}$

Non-negative Matrix Factorization (NMF)

Idea: We can add constraints to the low-rank matrix factorisation problem.

Non-negative matrix factorisation (NMF): Let $q < \min(p, n)$

$$\underset{\mathbf{L}\in\mathbb{R}^{n\times q},\mathbf{F}\in\mathbb{R}^{q\times p}}{\arg\min} ||\mathbf{X} - \mathbf{L}\mathbf{F}||_F^2 \quad \text{such that} \quad \mathbf{L} \geq 0,\ \mathbf{F} \geq 0$$

- **Sum of positive layers:** $\mathbf{X} \approx \sum_{j=1}^{q} \mathbf{L}^{(:,j)}\mathbf{F}^{(j,:)}$

Non-negative Matrix Factorization (NMF)

Idea: We can add constraints to the low-rank matrix factorisation problem.

Non-negative matrix factorisation (NMF): Let $q < \min(p, n)$

$$\operatorname*{arg\,min}_{\mathbf{L}\in\mathbb{R}^{n\times q}, \mathbf{F}\in\mathbb{R}^{q\times p}} ||\mathbf{X} - \mathbf{L}\mathbf{F}||_F^2 \quad \text{such that} \quad \mathbf{L} \geq 0,\ \mathbf{F} \geq 0$$

- **Sum of positive layers:** $\mathbf{X} \approx \sum_{j=1}^{q} \mathbf{L}^{(:,j)}\mathbf{F}^{(j,:)}$
- No fast specialised algorithm or analytic solution exists (NP-hard problem)

Non-negative Matrix Factorization (NMF)

Idea: We can add constraints to the low-rank matrix factorisation problem.

Non-negative matrix factorisation (NMF): Let $q < \min(p, n)$

$$\underset{\mathbf{L}\in\mathbb{R}^{n\times q},\mathbf{F}\in\mathbb{R}^{q\times p}}{\arg\min} ||\mathbf{X}-\mathbf{LF}||_F^2 \quad \text{such that} \quad \mathbf{L}\geq 0,\ \mathbf{F}\geq 0$$

- **Sum of positive layers:** $\mathbf{X} \approx \sum_{j=1}^{q} \mathbf{L}^{(:,j)}\mathbf{F}^{(j,:)}$
- No fast specialised algorithm or analytic solution exists (NP-hard problem)
- Requires that the data $\mathbf{X}$ has to be non-negative

Non-negative Matrix Factorization (NMF)

Idea: We can add constraints to the low-rank matrix factorisation problem.

Non-negative matrix factorisation (NMF): Let $q < \min(p, n)$

$$\underset{\mathbf{L}\in\mathbb{R}^{n\times q},\mathbf{F}\in\mathbb{R}^{q\times p}}{\arg\min} \|\mathbf{X}-\mathbf{LF}\|_F^2 \quad \text{such that} \quad \mathbf{L}\geq 0,\ \mathbf{F}\geq 0$$

- **Sum of positive layers:** $\mathbf{X} \approx \sum_{j=1}^{q} \mathbf{L}^{(:,j)}\mathbf{F}^{(j,:)}$
- No fast specialised algorithm or analytic solution exists (NP-hard problem)
- Requires that the data $\mathbf{X}$ has to be non-negative
- $\mathbf{L}$ and $\mathbf{F}$ are again not uniquely identifiable.

Non-negative Matrix Factorization (NMF)

Idea: We can add constraints to the low-rank matrix factorisation problem.

Non-negative matrix factorisation (NMF): Let $q < \min(p, n)$

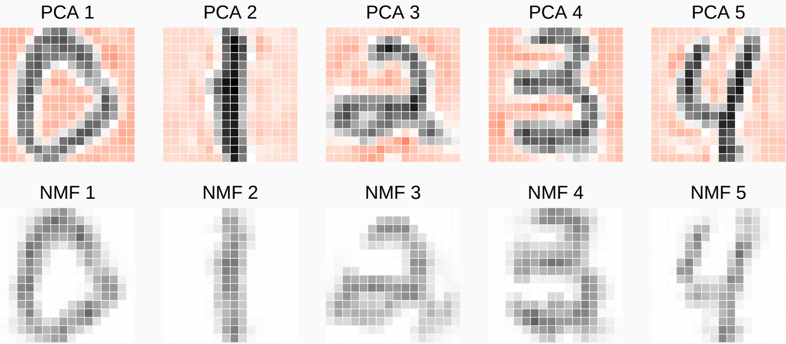
$$\operatorname*{arg\,min}_{\mathbf{L}\in\mathbb{R}^{n\times q},\mathbf{F}\in\mathbb{R}^{q\times p}} ||\mathbf{X} - \mathbf{LF}||_F^2 \quad \text{such that} \quad \mathbf{L} \geq 0,\ \mathbf{F} \geq 0$$

- **Sum of positive layers:** $\mathbf{X} \approx \sum_{j=1}^{q} \mathbf{L}^{(:,j)}\mathbf{F}^{(j,:)}$
- No fast specialised algorithm or analytic solution exists (NP-hard problem)
- Requires that the data $\mathbf{X}$ has to be non-negative
- $\mathbf{L}$ and $\mathbf{F}$ are again not uniquely identifiable.
- Choice of q not as straight-forward as for SVD

SVD vs NMF – Example: Reconstruction

MNIST-derived zip code digits ($n = 1000$, $p = 256$)

100 samples are drawn randomly from each class to keep the problem balanced.



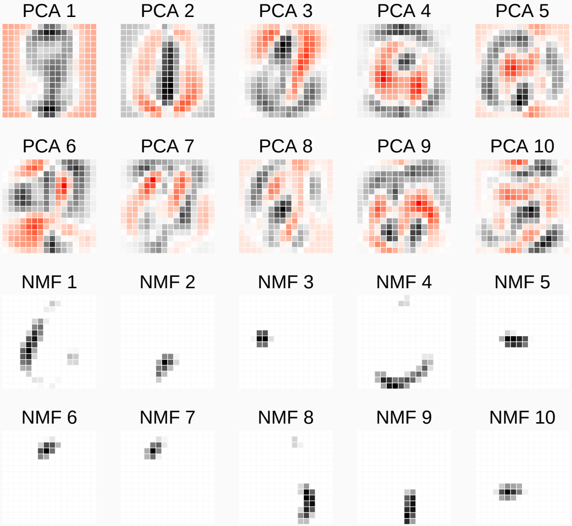
Red-ish colours are for negative values, white is around zero and dark stands for positive values. Reconstructions are done using 50 first PCs / $q = 50$.

SVD vs NMF – Example: Basis Components

Large difference between principal components (columns of $\mathbf{V}$) and NMF basis components (rows of $\mathbf{F}$)

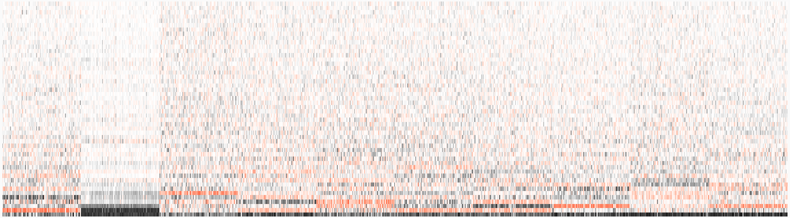
The non-negativity constraint leads to **sparsity** in the **basis** (in $\mathbf{F}$) and **coefficients** (in $\mathbf{L}$, next slide).

Therefore, NMF captures **sparse characteristic parts** while PCA components capture more global features.

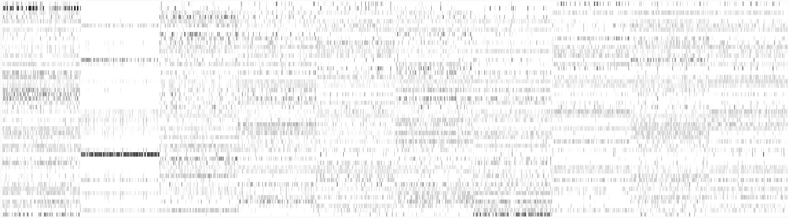


SVD vs NMF – Example: Coefficients ()

SVD coefficients



NMF coefficients



Note the additional **sparsity** in the NMF coefficients.

How to solve the NMF problem?

The NMF problem is

$$\underset{\mathbf{L}\in\mathbb{R}^{n\times q},\mathbf{F}\in\mathbb{R}^{q\times p}}{\arg\min} ||\mathbf{X}-\mathbf{LF}||_F^2 \quad \text{such that} \quad \mathbf{L}\geq 0, \mathbf{F}\geq 0$$

How to solve the NMF problem?

The NMF problem is

$$\underset{\mathbf{L}\in\mathbb{R}^{n\times q},\mathbf{F}\in\mathbb{R}^{q\times p}}{\arg\min} ||\mathbf{X}-\mathbf{LF}||_F^2 \quad \text{such that} \quad \mathbf{L}\geq 0, \mathbf{F}\geq 0$$

Most algorithms use **two-block coordinate descent** and solve

$$\mathbf{L}^{[t]} = \underset{\mathbf{L}\geq 0}{\arg\min} ||\mathbf{X}-\mathbf{L}\mathbf{F}^{[t-1]}||_F^2 \quad \text{and} \quad \mathbf{F}^{[t]} = \underset{\mathbf{F}\geq 0}{\arg\min} ||\mathbf{X}-\mathbf{L}^{[t]}\mathbf{F}||_F^2$$

iteratively.

How to solve the NMF problem?

The NMF problem is

$$\underset{\mathbf{L}\in\mathbb{R}^{n\times q},\mathbf{F}\in\mathbb{R}^{q\times p}}{\arg\min} \|\mathbf{X}-\mathbf{LF}\|_F^2 \quad \text{such that} \quad \mathbf{L}\geq 0, \mathbf{F}\geq 0$$

Most algorithms use **two-block coordinate descent** and solve

$$\mathbf{L}^{[t]} = \underset{\mathbf{L}\geq 0}{\arg\min} \|\mathbf{X}-\mathbf{LF}^{[t-1]}\|_F^2 \quad \text{and} \quad \mathbf{F}^{[t]} = \underset{\mathbf{F}\geq 0}{\arg\min} \|\mathbf{X}-\mathbf{L}^{[t]}\mathbf{F}\|_F^2$$

iteratively.

Note that the problem is **symmetric** in $\mathbf{L}$ and $\mathbf{F}$ since

$$\|\mathbf{X}-\mathbf{LF}\|_F^2 = \|\mathbf{X}^\top - \mathbf{F}^\top\mathbf{L}^\top\|_F^2.$$

No separate algorithms needed for $\mathbf{L}$ and $\mathbf{F}$.

Short note on cost functions

Our derviation was based on Frobenius norm and inspired by the SVD-based approach of the best rank-q approximation. However, other cost functions are possible.

Short note on cost functions

Our derviation was based on Frobenius norm and inspired by the SVD-based approach of the best rank-q approximation. However, other cost functions are possible.

- **Note:** Cost functions determine the distribution of noise

Short note on cost functions

Our derviation was based on Frobenius norm and inspired by the SVD-based approach of the best rank-q approximation. However, other cost functions are possible.

- **Note:** Cost functions determine the distribution of noise
- Frobenius norm implies Gaussian distribution

Short note on cost functions

Our derviation was based on Frobenius norm and inspired by the SVD-based approach of the best rank-q approximation. However, other cost functions are possible.

- **Note:** Cost functions determine the distribution of noise
- Frobenius norm implies Gaussian distribution
- An alternative for Poisson distributed data (count data)

$$D(\mathbf{X}||\mathbf{LF}) = \sum_{i=1}^{p}\sum_{j=1}^{n}\left(\mathbf{X}^{(i,j)}\log\frac{\mathbf{X}^{(i,j)}}{(\mathbf{LF})^{(i,j)}} - \mathbf{X}^{(i,j)} + (\mathbf{LF})^{(i,j)}\right)$$

Resembles the **Kullback-Leibler divergence** and the **log-likelihood of Poisson-distributed data** with mean $(\mathbf{LF})^{(i,j)}$ for $\mathbf{X}^{(i,j)}$.

Alternating least squares updates for NMF

A simple update rule is **alternating least squares (ALS)**: Solve the unconstrained least squares problem

$$\mathbf{Z}^{[t]} = \arg\min_{\mathbf{Z}\in\mathbb{R}^{q\times p}} ||\mathbf{X} - \mathbf{L}^{[t-1]}\mathbf{Z}||_F^2$$

and set elementwise $\mathbf{F}^{[t]} = \max(\mathbf{Z}^{[t]}, 0)$. Analogous for $\mathbf{L}^{[t]}$.

Alternating least squares updates for NMF

A simple update rule is **alternating least squares (ALS)**: Solve the unconstrained least squares problem

$$\mathbf{Z}^{[t]} = \arg\min_{\mathbf{Z}\in\mathbb{R}^{q\times p}} ||\mathbf{X} - \mathbf{L}^{[t-1]}\mathbf{Z}||_F^2$$

and set elementwise $\mathbf{F}^{[t]} = \max(\mathbf{Z}^{[t]}, 0)$. Analogous for $\mathbf{L}^{[t]}$.

- The method is cheap but can have convergence issues.

Alternating least squares updates for NMF

A simple update rule is **alternating least squares (ALS)**: Solve the unconstrained least squares problem

$$\mathbf{Z}^{[t]} = \arg\min_{\mathbf{Z}\in\mathbb{R}^{q\times p}} ||\mathbf{X} - \mathbf{L}^{[t-1]}\mathbf{Z}||_F^2$$

and set elementwise $\mathbf{F}^{[t]} = \max(\mathbf{Z}^{[t]}, 0)$. Analogous for $\mathbf{L}^{[t]}$.

- The method is cheap but can have convergence issues.
- Can be useful for initialisation (some steps of ALS first, then another algorithm)

Alternating non-negative least squares updates for NMF

It holds that

$$\begin{aligned}\|\mathbf{X}-\mathbf{L}\mathbf{F}\|_F^2 &= \sum_{i=1}^{p} \|\mathbf{X}^{(:,i)} - \mathbf{L}\mathbf{F}^{(:,i)}\|_2^2 \\ &= \sum_{i=1}^{p} \mathbf{F}^{(:,i)^\top} (\underbrace{\mathbf{L}^\top\mathbf{L}}_{=\mathbf{Q}})\mathbf{F}^{(:,i)} + (\underbrace{-\mathbf{L}^\top\mathbf{X}^{(:,i)}}_{=\mathbf{c}})^\top \mathbf{F}^{(:,i)} + \|\mathbf{X}^{(:,i)}\|_2^2\end{aligned}$$

Minimizing over $\mathbf{F}^{(:,i)} \geq 0$, this is a sum of p independent **non-negative least squares (NNLS)** problems. The resulting update rule is called **alternating NNLS**.

NNLS problems are equivalent to quadratic programming problems of the form

$$\arg\min_{\mathbf{x}\geq 0} \frac{1}{2}\mathbf{x}^\top\mathbf{Q}\mathbf{x} + \mathbf{c}^\top\mathbf{x}$$

for positive semi-definite $\mathbf{Q}$.

Multiplicative updates for NMF

Multiplicative updates (MU) have been popularized by Lee and Seung (1999). Their form depends on the cost function. In the following $\mathbf{A} \circ \mathbf{B}$ denotes elementwise multiplication of matrices and division is also meant elementwise.

1. Frobenius norm:

$$\mathbf{L} \leftarrow \mathbf{L} \circ \frac{\mathbf{X}\mathbf{F}^\top}{\mathbf{L}\mathbf{F}\mathbf{F}^\top} \quad \text{and} \quad \mathbf{F} \leftarrow \mathbf{F} \circ \frac{\mathbf{L}^\top\mathbf{X}}{\mathbf{L}^\top\mathbf{L}\mathbf{F}}$$

2. KL divergence:

$$\mathbf{L}^{(l,k)} \leftarrow \mathbf{L}^{(l,k)} \frac{\sum_{i=1}^{p} \mathbf{F}^{(k,i)}\mathbf{X}^{(l,i)} / (\mathbf{L}\mathbf{F})^{(l,i)}}{\sum_{i=1}^{p} \mathbf{F}^{(k,i)}} \quad \text{and}$$

$$\mathbf{F}^{(k,i)} \leftarrow \mathbf{F}^{(k,i)} \frac{\sum_{l=1}^{n} \mathbf{L}^{(l,k)}\mathbf{X}^{(l,i)} / (\mathbf{L}\mathbf{F})^{(l,i)}}{\sum_{l=1}^{n} \mathbf{L}^{(l,k)}}$$

Multiplicative updates for NMF and gradient descent

Multiplicative updates are a special case of **gradient descent**. Let $J(\mathbf{L}, \mathbf{F}) = \frac{1}{2}||\mathbf{X} - \mathbf{L}\mathbf{F}||_F^2$ then

$$\nabla_{\mathbf{L}} J = \mathbf{L}\mathbf{F}\mathbf{F}^\top - \mathbf{X}\mathbf{F}^\top$$

$$\nabla_{\mathbf{F}} J = \mathbf{L}^\top\mathbf{L}\mathbf{F} - \mathbf{L}^\top\mathbf{X}$$

Multiplicative updates for NMF and gradient descent

Multiplicative updates are a special case of **gradient descent**. Let $J(\mathbf{L}, \mathbf{F}) = \frac{1}{2}||\mathbf{X} - \mathbf{L}\mathbf{F}||_F^2$ then

$$\nabla_{\mathbf{L}} J = \mathbf{L}\mathbf{F}\mathbf{F}^\top - \mathbf{X}\mathbf{F}^\top$$

$$\nabla_{\mathbf{F}} J = \mathbf{L}^\top\mathbf{L}\mathbf{F} - \mathbf{L}^\top\mathbf{X}$$

Gradient descent in $\mathbf{L}$ for step-length α performs

$$\mathbf{L} \leftarrow \mathbf{L} - \alpha\nabla_{\mathbf{L}} J$$

Multiplicative updates for NMF and gradient descent

Multiplicative updates are a special case of **gradient descent**. Let $J(\mathbf{L}, \mathbf{F}) = \frac{1}{2}||\mathbf{X} - \mathbf{LF}||_F^2$ then

$$\nabla_{\mathbf{L}} J = \mathbf{LFF}^\top - \mathbf{XF}^\top$$

$$\nabla_{\mathbf{F}} J = \mathbf{L}^\top\mathbf{LF} - \mathbf{L}^\top\mathbf{X}$$

Gradient descent in $\mathbf{L}$ for step-length α performs

$$\mathbf{L} \leftarrow \mathbf{L} - \alpha \nabla_{\mathbf{L}} J$$

It can be shown that

$$\boldsymbol{\alpha} = \frac{\mathbf{L}}{\mathbf{LFF}^\top} \in \mathbb{R}^{n \times q},$$

where division is element-wise, is an **admissible step length**. Element-wise multiplication of $\boldsymbol{\alpha}$ and $\nabla_{\mathbf{L}} J$ yields the MU for $\mathbf{L}$. Analogously for $\mathbf{F}$.

Multiplicative updates for NMF and gradient descent

Multiplicative updates are a special case of **gradient descent**. Let $J(\mathbf{L}, \mathbf{F}) = \frac{1}{2}||\mathbf{X} - \mathbf{L}\mathbf{F}||_F^2$ then

$$\nabla_{\mathbf{L}} J = \mathbf{L}\mathbf{F}\mathbf{F}^\top - \mathbf{X}\mathbf{F}^\top$$

$$\nabla_{\mathbf{F}} J = \mathbf{L}^\top\mathbf{L}\mathbf{F} - \mathbf{L}^\top\mathbf{X}$$

Gradient descent in $\mathbf{L}$ for step-length α performs

$$\mathbf{L} \leftarrow \mathbf{L} - \alpha \nabla_{\mathbf{L}} J$$

It can be shown that

$$\boldsymbol{\alpha} = \frac{\mathbf{L}}{\mathbf{L}\mathbf{F}\mathbf{F}^\top} \in \mathbb{R}^{n \times q},$$

where division is element-wise, is an **admissible step length**. Element-wise multiplication of $\boldsymbol{\alpha}$ and $\nabla_{\mathbf{L}} J$ yields the MU for $\mathbf{L}$. Analogously for $\mathbf{F}$.

Note: Analogous results hold for the KL divergence.

Advantages of NMF

- **Interpretability:** As in the case of truncated SVD we are adding layers, but now all layers are positive and each layer adds information

Advantages of NMF

- **Interpretability:** As in the case of truncated SVD we are adding layers, but now all layers are positive and each layer adds information
- **Clustering interpretation:**
  - The rows of $\mathbf{F}$ can be interpreted as cluster centroids
  - Cluster membership of each observation is determined by the rows of $\mathbf{L}$
  - Observation j is assigned to the cluster k if $\mathbf{L}^{(j,k)} > \mathbf{L}^{(j,i)}$ for all $i \neq k$

Initialising NMF

NMF can be initialised in multiple ways

- **Random initialisation:** Uniformly distributed entries in $[0, 1]$ for $\mathbf{L}$ and $\mathbf{F}$
- **Clustering techniques:** Run k-means with q clusters on data, store cluster centroids in rows of $\mathbf{F}$ and $\mathbf{L}^{(l,k)} \neq 0 \Leftrightarrow \mathbf{X}^{(l,:)}$ belongs to cluster k
- **SVD**: Determine best rank-q-approximation $\sum_{i=1}^{q} d_{ii}\mathbf{v}_i\mathbf{u}_i^\top$, note that

$$\begin{aligned} d_{ii}\mathbf{u}_i\mathbf{v}_i^\top &= ([+d_{ii}\mathbf{u}_i]_+[+\mathbf{v}_i^\top]_+ + [-d_{ii}\mathbf{u}_i]_+[-\mathbf{v}_i^\top]_+) \\ &\quad - ([+d_{ii}\mathbf{u}_i]_+[-\mathbf{v}_i^\top]_+ + [-d_{ii}\mathbf{u}_i]_+[+\mathbf{v}_i^\top]_+) \end{aligned}$$

  and initialize NMF by summing only the positive parts or the larger of the positive parts.

Take-home message

- Linear dimension reduction approximates matrices through additive layers (hence linear).
- The SVD-based approach leads to factor analysis, built on the intuition that there are underlying factors describing the data and the intensity of their presence in a sample is quantified in the loadings
- By adding non-negativity constraints to the matrix factorisation problem, NMF creates more interpretable results and can be used for clustering at the same time