Nonnegative Matrix Factorization

Theory and Application

Presentation for Matrix Decompositions
PhD in Statistical Sciences
Academic year 2016/2017
University of Bologna

elena.geminiani4@unibo.it

30th March 2017

Table of contents

1. Nonnegative Matrix Factorization
   - Introduction
   - The method
   - Matrix factorization
   - An ill-posed problems
   - Extensions
2. Application to Text Mining
   - Introduction
   - Analysis
3. Conclusion

Introduction

- Recent developments in technology result in increasing quantities of data, rapidly overwhelming many of classical analysis tools

Introduction

- Recent developments in technology result in increasing quantities of data, rapidly overwhelming many of classical analysis tools
  ➜ Powerful methods for data representation and dimension reduction

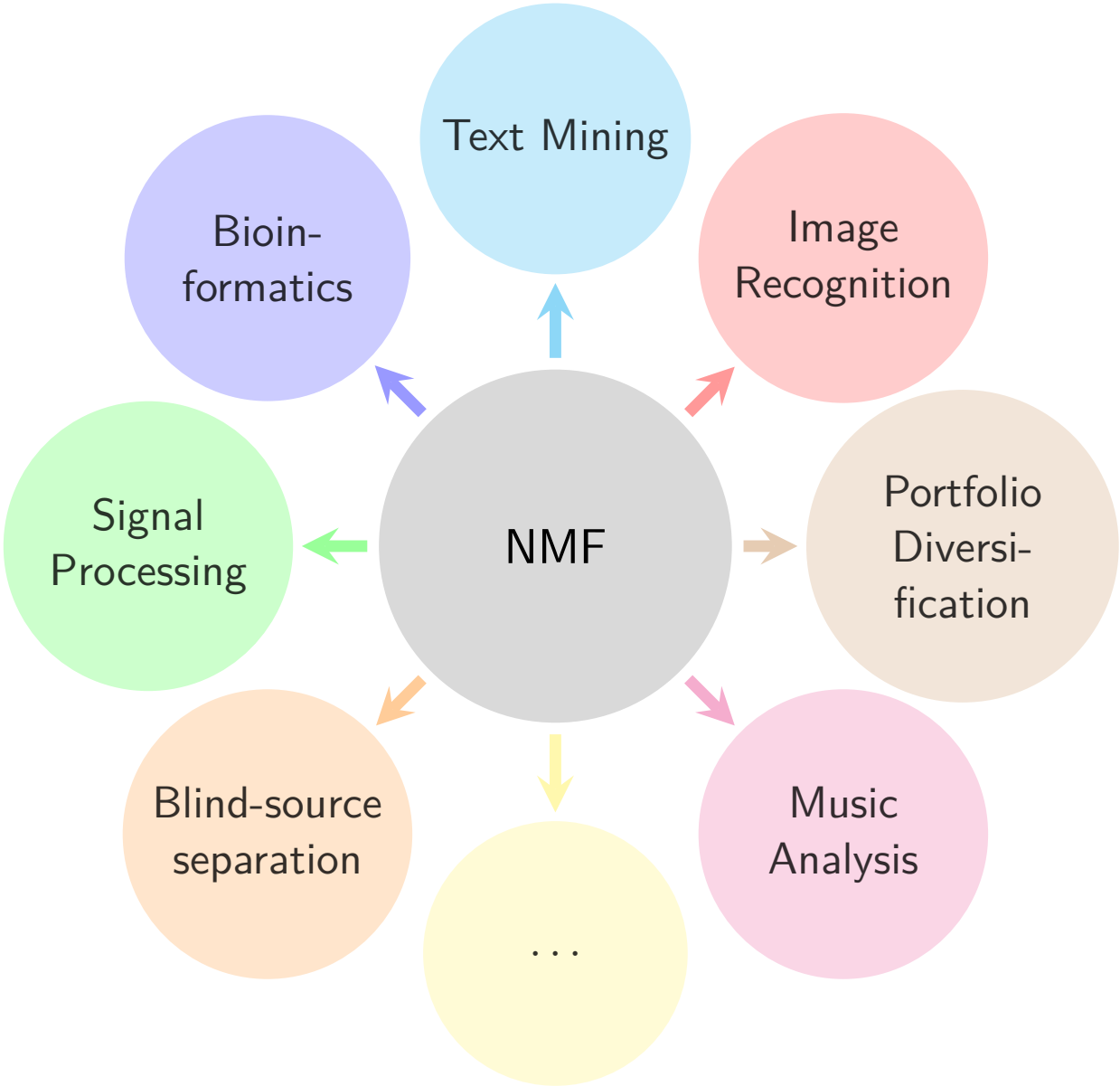
Introduction

- Recent developments in technology result in increasing quantities of data, rapidly overwhelming many of classical analysis tools
  ➜ Powerful methods for data representation and dimension reduction
- *Nonnegative data* (e.g. pixel intensities, users scores) cannot be analyzed by means of well-known techniques since they cannot guarantee to preserve nonnegativity

Introduction

- Recent developments in technology result in increasing quantities of data, rapidly overwhelming many of classical analysis tools
  ➜ Powerful methods for data representation and dimension reduction
- *Nonnegative data* (e.g. pixel intensities, users scores) cannot be analyzed by means of well-known techniques since they cannot guarantee to preserve nonnegativity
- **Nonnegative Matrix Factorization** (NMF; Paatero and Tapper 1994; Lee and Seung 1999) is one of the most popular low-rank approximations obtained via subspace approximation, used for compression, visualization, feature selection and noise filtering.

Fields of application



The method

NMF allows to decompose multivariate data and search the latent features responsible for the generation of observable variables.

The method

NMF allows to decompose multivariate data and search the latent features responsible for the generation of observable variables.

Target matrix $\boldsymbol{V}$ $(n \times m)$

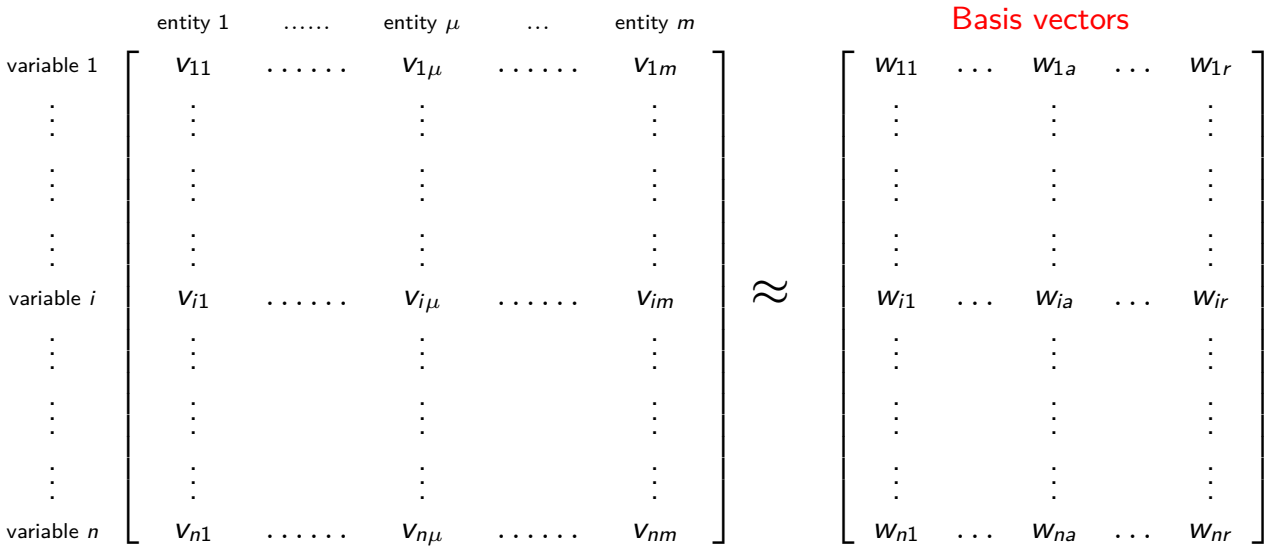
	entity 1	.....	entity μ	...	entity m
variable 1	v_{11}	. . . . .	$v_{1\mu}$	. . . . .	v_{1m}
variable i	v_{i1}	. . . . .	$v_{i\mu}$	. . . . .	v_{im}
variable n	v_{n1}	. . . . .	$v_{n\mu}$	. . . . .	v_{nm}

$\boldsymbol{V}$

The method

NMF allows to decompose multivariate data and search the latent features responsible for the generation of observable variables.

Target matrix $\boldsymbol{V}$ $(n \times m)$ is approximately factorized into the product of two nonnegative factor matrices, a $(n \times r)$ *basis matrix* $\boldsymbol{W}$

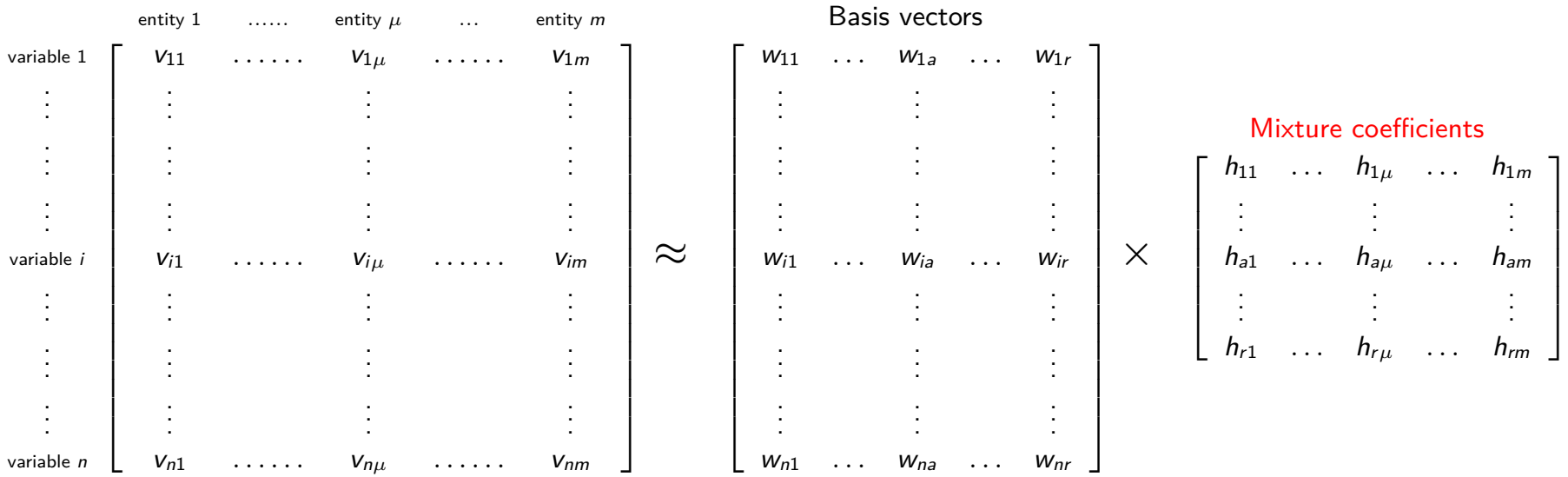


$$\boldsymbol{V} \approx \boldsymbol{W}$$

The method

NMF allows to decompose multivariate data and search the latent features responsible for the generation of observable variables.

Target matrix $\boldsymbol{V}$ $(n \times m)$ is approximately factorized into the product of two nonnegative factor matrices, a $(n \times r)$ *basis matrix* $\boldsymbol{W}$ and a $(r \times m)$ *mixture coefficients matrix* $\boldsymbol{H}$

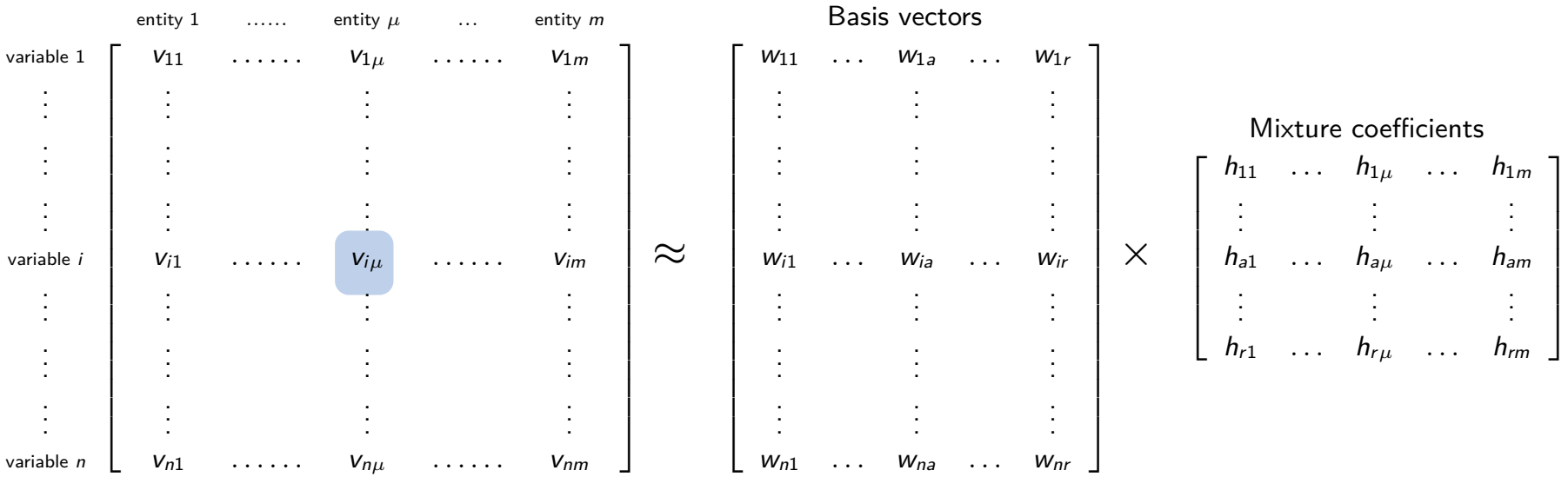


$$\boldsymbol{V} \approx \boldsymbol{W}\boldsymbol{H} \qquad \text{with } (\boldsymbol{W}, \boldsymbol{H}) \geq 0$$

The method

NMF allows to decompose multivariate data and search the latent features responsible for the generation of observable variables.

Target matrix $\boldsymbol{V}$ $(n \times m)$ is approximately factorized into the product of two nonnegative factor matrices, a $(n \times r)$ *basis matrix* $\boldsymbol{W}$ and a $(r \times m)$ *mixture coefficients matrix* $\boldsymbol{H}$

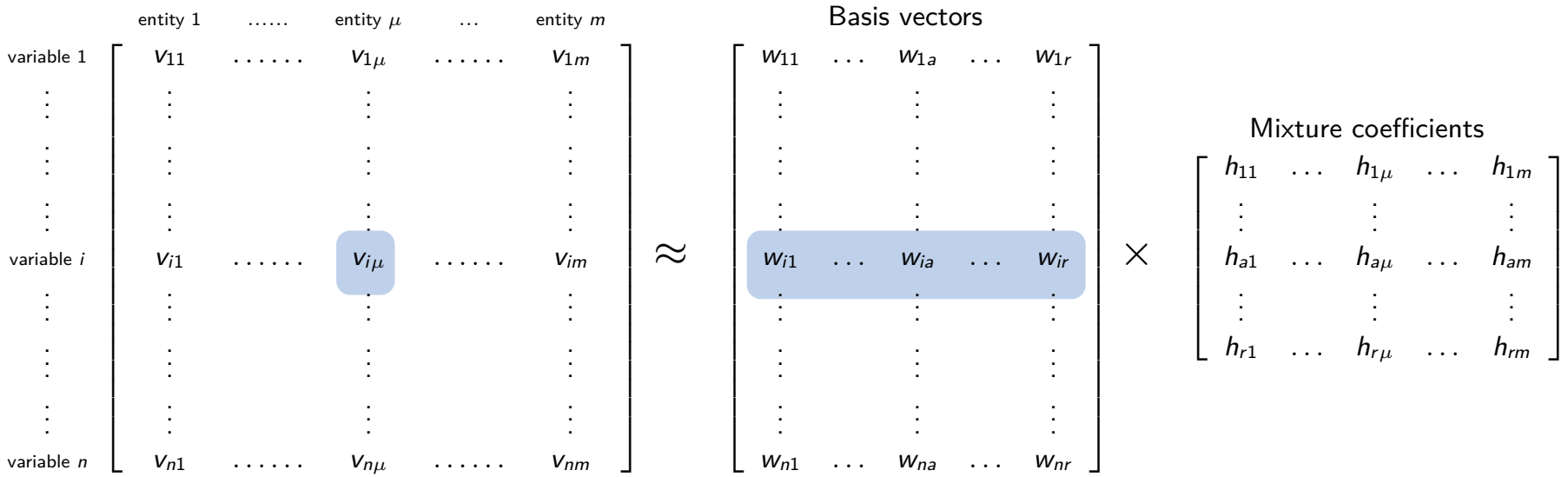


$$v_{i\mu} \approx (\boldsymbol{WH})_{i\mu}$$

The method

NMF allows to decompose multivariate data and search the latent features responsible for the generation of observable variables.

Target matrix $\boldsymbol{V}$ $(n \times m)$ is approximately factorized into the product of two nonnegative factor matrices, a $(n \times r)$ *basis matrix* $\boldsymbol{W}$ and a $(r \times m)$ *mixture coefficients matrix* $\boldsymbol{H}$

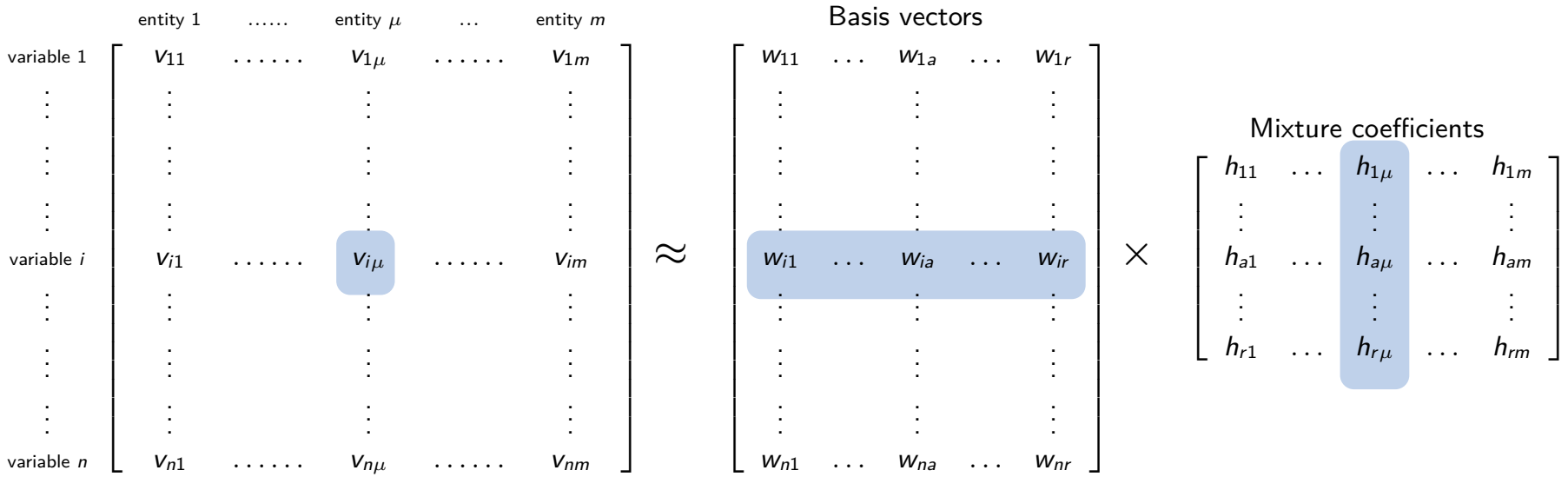


$$v_{i\mu} \approx (\boldsymbol{WH})_{i\mu} = \sum_{a=1}^{r} w_{ia}$$

The method

NMF allows to decompose multivariate data and search the latent features responsible for the generation of observable variables.

Target matrix $\boldsymbol{V}$ $(n \times m)$ is approximately factorized into the product of two nonnegative factor matrices, a $(n \times r)$ *basis matrix* $\boldsymbol{W}$ and a $(r \times m)$ *mixture coefficients matrix* $\boldsymbol{H}$

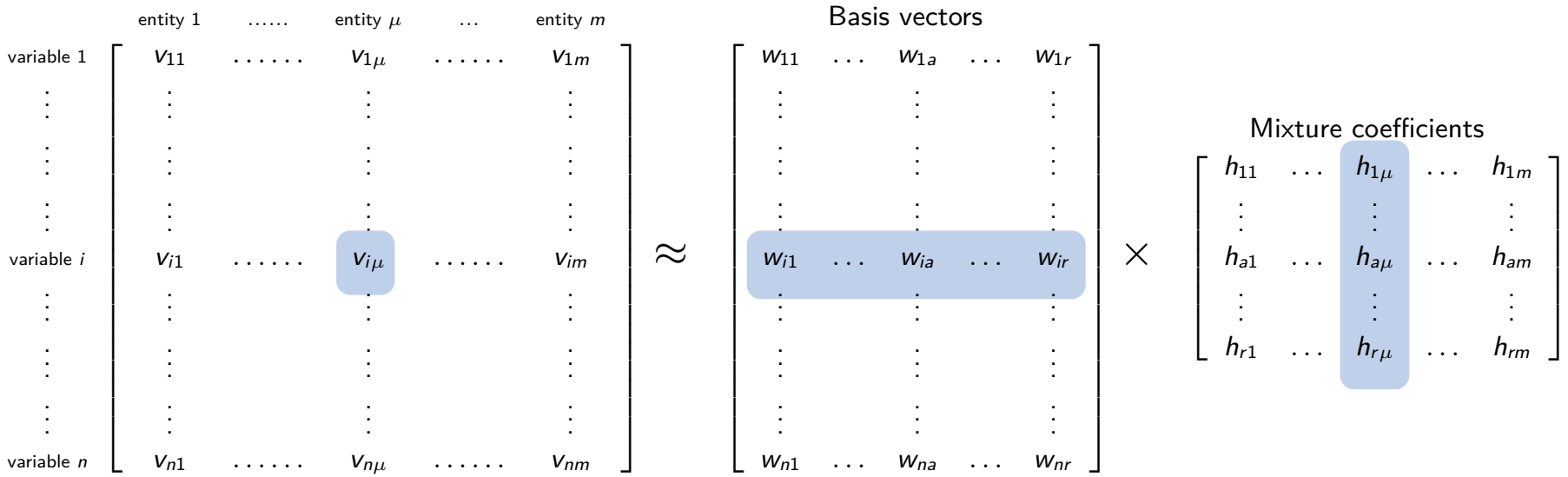


$$v_{i\mu} \approx (\boldsymbol{WH})_{i\mu} = \sum_{a=1}^{r} w_{ia} h_{a\mu}$$

The method

NMF allows to decompose multivariate data and search the latent features responsible for the generation of observable variables.

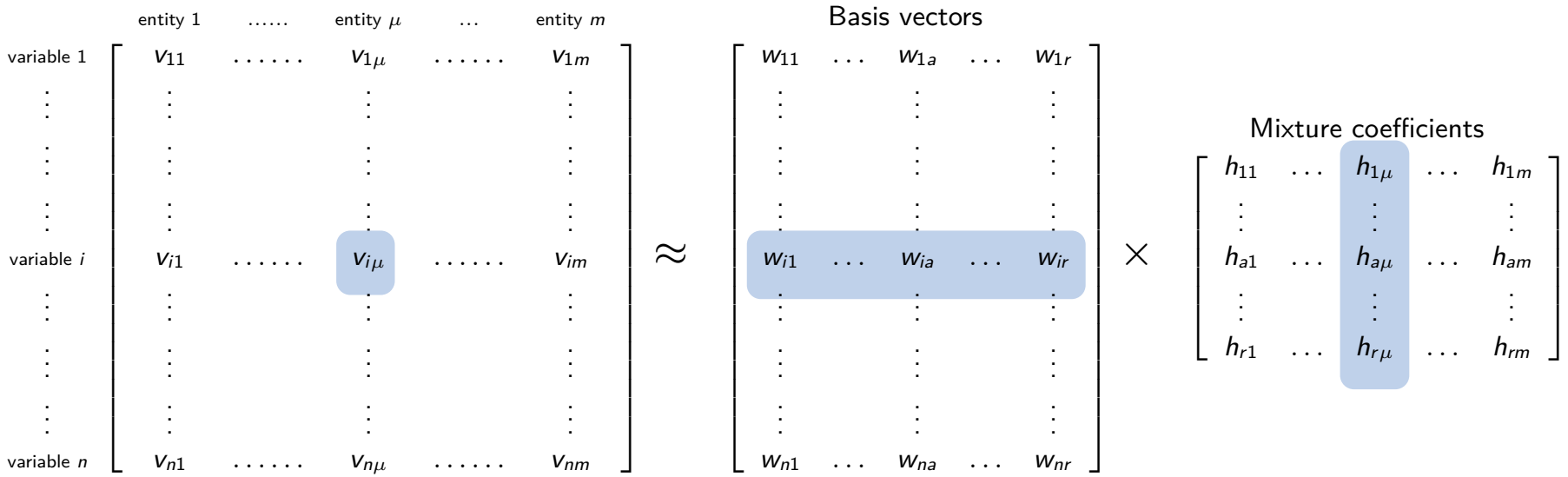
Target matrix $\boldsymbol{V}$ $(n \times m)$ is approximately factorized into the product of two nonnegative factor matrices, a $(n \times r)$ *basis matrix* $\boldsymbol{W}$ and a $(r \times m)$ *mixture coefficients matrix* $\boldsymbol{H}$



$$v_{i\mu} \approx (\boldsymbol{WH})_{i\mu} = \sum_{a=1}^{r} w_{ia} h_{a\mu}, \qquad \text{for } i = 1, \ldots, n, \text{ and } \mu = 1, \ldots, m$$

The method

NMF allows to decompose multivariate data and search the latent features responsible for the generation of observable variables.
Target matrix $\boldsymbol{V}$ $(n \times m)$ is approximately factorized into the product of two nonnegative factor matrices, a $(n \times r)$ *basis matrix* $\boldsymbol{W}$ and a $(r \times m)$ *mixture coefficients matrix* $\boldsymbol{H}$



$$v_{i\mu} \approx (\boldsymbol{WH})_{i\mu} = \sum_{a=1}^{r} w_{ia} h_{a\mu}, \qquad \text{for } i = 1, \ldots, n, \text{ and } \mu = 1, \ldots, m$$

A crucial role is played by the factorization rank $r << min(n, m)$ such that $(n+m)r < nm$.

Matrix Factorization

The factorization of the matrix $\boldsymbol{V}$ can be achieved in several ways using numerical algorithms (block coordinate descendent scheme).

Matrix Factorization

The factorization of the matrix $\boldsymbol{V}$ can be achieved in several ways using numerical algorithms (block coordinate descendent scheme).

General framework of NMF algorithms

Input: A nonnegative matrix $\boldsymbol{V} \in \mathbb{R}_{+}^{n \times m}$ and the factorization rank r

- Generate some initial matrices $\boldsymbol{W}^{(0)} \geq 0, \boldsymbol{H}^{(0)} \geq 0$
- **for** $t = 1, 2, \ldots,$ stopping time
  $\boldsymbol{W}^{(t)} = \text{update } (\boldsymbol{V}, \boldsymbol{H}^{(t-1)}, \boldsymbol{W}^{(t-1)})$
  $\boldsymbol{H}^{(t)^T} = \text{update } (\boldsymbol{V}^T, \boldsymbol{W}^{(t)^T}, \boldsymbol{H}^{(t-1)^T})$
  end for

Output: A rank-r NMF of $\boldsymbol{V} \approx \boldsymbol{W}\boldsymbol{H}$, for $(\boldsymbol{W}, \boldsymbol{H}) \geq 0$

Matrix Factorization

The factorization of the matrix $\boldsymbol{V}$ can be achieved in several ways using numerical algorithms (block coordinate descendent scheme).

General framework of NMF algorithms

Input: A nonnegative matrix $\boldsymbol{V} \in \mathbb{R}_{+}^{n \times m}$ and the factorization rank r

- Generate some initial matrices $\boldsymbol{W}^{(0)} \geq 0, \boldsymbol{H}^{(0)} \geq 0$
- **for** $t = 1, 2, \ldots,$ stopping time
  $\boldsymbol{W}^{(t)} = \text{update } (\boldsymbol{V}, \boldsymbol{H}^{(t-1)}, \boldsymbol{W}^{(t-1)})$
  $\boldsymbol{H}^{(t)^T} = \text{update } (\boldsymbol{V}^T, \boldsymbol{W}^{(t)^T}, \boldsymbol{H}^{(t-1)^T})$
  end for

Output: A rank-r NMF of $\boldsymbol{V} \approx \boldsymbol{W}\boldsymbol{H}$, for $(\boldsymbol{W}, \boldsymbol{H}) \geq 0$

- Objective function
- Numerical algorithm

Objective function

To find an approximate factorization $\boldsymbol{V} \approx \boldsymbol{WH}$, one needs to define loss functions $L(\boldsymbol{V}, \boldsymbol{WH})$ that quantify the quality of the approximation.

Objective function

To find an approximate factorization $\boldsymbol{V} \approx \boldsymbol{WH}$, one needs to define loss functions $L(\boldsymbol{V}, \boldsymbol{WH})$ that quantify the quality of the approximation.

- Square Euclidean distance:

$$L(\boldsymbol{V}, \boldsymbol{WH}) = \|\boldsymbol{V} - \boldsymbol{WH}\|^2 = \sum_{i=1}^{n} \sum_{\mu=1}^{m} \left(v_{i\mu} - (\boldsymbol{WH})_{i\mu}\right)^2$$

Objective function

To find an approximate factorization $\boldsymbol{V} \approx \boldsymbol{WH}$, one needs to define loss functions $L(\boldsymbol{V}, \boldsymbol{WH})$ that quantify the quality of the approximation.

- Square Euclidean distance:

$$L(\boldsymbol{V}, \boldsymbol{WH}) = \|\boldsymbol{V} - \boldsymbol{WH}\|^2 = \sum_{i=1}^{n}\sum_{\mu=1}^{m}\left(v_{i\mu} - (\boldsymbol{WH})_{i\mu}\right)^2$$

- Generalized Kullback-Leibler divergence

$$L(\boldsymbol{V}, \boldsymbol{WH}) = D(\boldsymbol{V}||\boldsymbol{WH}) = \sum_{i=1}^{n}\sum_{\mu=1}^{m}\left(v_{i\mu}\log\frac{v_{i\mu}}{(\boldsymbol{WH})_{i\mu}} - v_{i\mu} + (\boldsymbol{WH})_{i\mu}\right)$$

Objective function

To find an approximate factorization $\boldsymbol{V} \approx \boldsymbol{WH}$, one needs to define loss functions $L(\boldsymbol{V}, \boldsymbol{WH})$ that quantify the quality of the approximation.

- Square Euclidean distance:

$$L(\boldsymbol{V}, \boldsymbol{WH}) = \|\boldsymbol{V} - \boldsymbol{WH}\|^2 = \sum_{i=1}^{n}\sum_{\mu=1}^{m}\left(v_{i\mu} - (\boldsymbol{WH})_{i\mu}\right)^2$$

- Generalized Kullback-Leibler divergence

$$L(\boldsymbol{V}, \boldsymbol{WH}) = D(\boldsymbol{V}||\boldsymbol{WH}) = \sum_{i=1}^{n}\sum_{\mu=1}^{m}\left(v_{i\mu}\log\frac{v_{i\mu}}{(\boldsymbol{WH})_{i\mu}} - v_{i\mu} + (\boldsymbol{WH})_{i\mu}\right)$$

Nonlinear optimization problem: minimize the objective function with respect to matrix factors, subject to nonnegativity constraint:

$$\min_{\boldsymbol{W},\boldsymbol{H}\geq 0}\Big[\underbrace{L(\boldsymbol{V}, \boldsymbol{WH})}_{\textit{Loss function}} + \underbrace{R(\boldsymbol{V}, \boldsymbol{WH})}_{\textit{Regularization term}}\Big]$$

Matrix Factorization

The factorization of the matrix $\boldsymbol{V}$ can be achieved in several ways using numerical algorithms (block coordinate descendent scheme).

General framework of NMF algorithms

Input: A nonnegative matrix $\boldsymbol{V} \in \mathbb{R}_{+}^{n \times m}$ and the factorization rank r

- Generate some initial matrices $\boldsymbol{W}^{(0)} \geq 0, \boldsymbol{H}^{(0)} \geq 0$
- **for** $t = 1, 2, \ldots,$ stopping time
  $\boldsymbol{W}^{(t)} = \text{update } (\boldsymbol{V}, \boldsymbol{H}^{(t-1)}, \boldsymbol{W}^{(t-1)})$
  $\boldsymbol{H}^{(t)^T} = \text{update } (\boldsymbol{V}^T, \boldsymbol{W}^{(t)^T}, \boldsymbol{H}^{(t-1)^T})$
  end for

Output: A rank-r NMF of $\boldsymbol{V} \approx \boldsymbol{W}\boldsymbol{H}$, for $(\boldsymbol{W}, \boldsymbol{H}) \geq 0$

Matrix Factorization

The factorization of the matrix $\boldsymbol{V}$ can be achieved in several ways using numerical algorithms (block coordinate descendent scheme).

General framework of NMF algorithms

Input: A nonnegative matrix $\boldsymbol{V} \in \mathbb{R}_{+}^{n \times m}$ and the factorization rank r

- Generate some initial matrices $\boldsymbol{W}^{(0)} \geq 0, \boldsymbol{H}^{(0)} \geq 0$
- **for** $t = 1, 2, \ldots,$ stopping time
  $\boldsymbol{W}^{(t)} = \text{update } (\boldsymbol{V}, \boldsymbol{H}^{(t-1)}, \boldsymbol{W}^{(t-1)})$
  $\boldsymbol{H}^{(t)^T} = \text{update } (\boldsymbol{V}^T, \boldsymbol{W}^{(t)^T}, \boldsymbol{H}^{(t-1)^T})$
  end for

Output: A rank-r NMF of $\boldsymbol{V} \approx \boldsymbol{W}\boldsymbol{H}$, for $(\boldsymbol{W}, \boldsymbol{H}) \geq 0$

- Objective function
- Numerical algorithm

Numerical algorithms

Cost functions are not simultaneously convex in both $\boldsymbol{W}$ and $\boldsymbol{H}$

Numerical algorithms

Cost functions are not simultaneously convex in both $\boldsymbol{W}$ and $\boldsymbol{H}$ ➜ local minima

Numerical algorithms

Cost functions are not simultaneously convex in both $\boldsymbol{W}$ and $\boldsymbol{H}$ ➜ local minima

Numerical techniques can be classified in three categories:

Numerical algorithms

Cost functions are not simultaneously convex in both $\boldsymbol{W}$ and $\boldsymbol{H}$ ➜ local minima

Numerical techniques can be classified in three categories:

1. **Gradient descent algorithms**: simplest to implement, but very slow to converge (if at all); sensitive to the step size choice and lack a convergence supporting theory;

Numerical algorithms

Cost functions are not simultaneously convex in both $\boldsymbol{W}$ and $\boldsymbol{H}$ ➜ local minima

Numerical techniques can be classified in three categories:

1. **Gradient descent algorithms**: simplest to implement, but very slow to converge (if at all); sensitive to the step size choice and lack a convergence supporting theory;
2. **Multiplicative update rules**: extensively used due to their simplicity, applicable to sparse matrices, proposed as numerical algorithm for solving NMF by Lee and Seung 2001;

Numerical algorithms

Cost functions are not simultaneously convex in both $\boldsymbol{W}$ and $\boldsymbol{H}$ ➜ local minima

Numerical techniques can be classified in three categories:

1. **Gradient descent algorithms**: simplest to implement, but very slow to converge (if at all); sensitive to the step size choice and lack a convergence supporting theory;
2. **Multiplicative update rules**: extensively used due to their simplicity, applicable to sparse matrices, proposed as numerical algorithm for solving NMF by Lee and Seung 2001;
3. **Alternating least squares algorithms** (Kim and Park 2007): a least squares step is followed by another least squares step in an alternating fashion; very fast, quick to converge, and give accurate factors.

Numerical algorithms

Cost functions are not simultaneously convex in both $\boldsymbol{W}$ and $\boldsymbol{H}$ ➜ local minima

Numerical techniques can be classified in three categories:

- ✘ **Gradient descent algorithms**: simplest to implement, but very slow to converge (if at all); sensitive to the step size choice and lack a convergence supporting theory;
- ✓ **Multiplicative update rules**: extensively used due to their simplicity, applicable to sparse matrices, proposed as numerical algorithm for solving NMF by Lee and Seung 2001;
- ✓ **Alternating least squares algorithms** (Kim and Park 2007): a least squares step is followed by another least squares step in an alternating fashion; very fast, quick to converge, and give accurate factors.

Numerical algorithms

Cost functions are not simultaneously convex in both $\boldsymbol{W}$ and $\boldsymbol{H}$ ➜ local minima

Numerical techniques can be classified in three categories:

1. **Gradient descent algorithms**: simplest to implement, but very slow to converge (if at all); sensitive to the step size choice and lack a convergence supporting theory;

➥ **Multiplicative update rules**: extensively used due to their simplicity, applicable to sparse matrices, proposed as numerical algorithm for solving NMF by Lee and Seung 2001;

2. **Alternating least squares algorithms** (Kim and Park 2007): a least squares step is followed by another least squares step in an alternating fashion; very fast, quick to converge, and give accurate factors.

Multiplicative rules

They can be divided according to the choice of the objective function.
For $a = 1, \ldots, r \quad \mu = 1, \ldots, m \quad i = 1, \ldots, n$:

Multiplicative rules

They can be divided according to the choice of the objective function.
For $a = 1, \ldots, r \quad \mu = 1, \ldots, m \quad i = 1, \ldots, n$:

Divergence measure

$$h_{a\mu} \leftarrow h_{a\mu} \frac{\sum_{i=1}^{n} w_{ia} \frac{v_{i\mu}}{(\boldsymbol{WH})_{i\mu}}}{\sum_{i=1}^{n} w_{ia}} \qquad w_{ia} \leftarrow w_{ia} \frac{\sum_{\mu=1}^{m} h_{a\mu} \frac{v_{i\mu}}{(\boldsymbol{WH})_{i\mu}}}{\sum_{\mu=1}^{m} h_{a\mu}}$$

Multiplicative rules

They can be divided according to the choice of the objective function.
For $a = 1, \ldots, r \quad \mu = 1, \ldots, m \quad i = 1, \ldots, n$:

Divergence measure

$$h_{a\mu} \leftarrow h_{a\mu} \frac{\sum_{i=1}^{n} w_{ia} \frac{v_{i\mu}}{(\boldsymbol{WH})_{i\mu}}}{\sum_{i=1}^{n} w_{ia}} \qquad w_{ia} \leftarrow w_{ia} \frac{\sum_{\mu=1}^{m} h_{a\mu} \frac{v_{i\mu}}{(\boldsymbol{WH})_{i\mu}}}{\sum_{\mu=1}^{m} h_{a\mu}}$$

Square Euclidean distance

$$h_{a\mu} \leftarrow h_{a\mu} \frac{\left(\boldsymbol{W}^T \boldsymbol{V}\right)_{a\mu}}{\left(\boldsymbol{W}^T \boldsymbol{WH}\right)_{a\mu}} \qquad w_{ia} \leftarrow w_{ia} \frac{\left(\boldsymbol{VH}^T\right)_{ia}}{\left(\boldsymbol{WHH}^T\right)_{ia}}$$

Multiplicative rules

They can be divided according to the choice of the objective function.
For $a = 1, \ldots, r \quad \mu = 1, \ldots, m \quad i = 1, \ldots, n$:

Divergence measure

$$h_{a\mu} \leftarrow h_{a\mu} \frac{\sum_{i=1}^{n} w_{ia} \frac{v_{i\mu}}{(\boldsymbol{WH})_{i\mu}}}{\sum_{i=1}^{n} w_{ia}} \qquad w_{ia} \leftarrow w_{ia} \frac{\sum_{\mu=1}^{m} h_{a\mu} \frac{v_{i\mu}}{(\boldsymbol{WH})_{i\mu}}}{\sum_{\mu=1}^{m} h_{a\mu}}$$

Square Euclidean distance

$$h_{a\mu} \leftarrow h_{a\mu} \frac{\left(\boldsymbol{W}^T \boldsymbol{V}\right)_{a\mu}}{\left(\boldsymbol{W}^T \boldsymbol{WH}\right)_{a\mu}} \qquad w_{ia} \leftarrow w_{ia} \frac{\left(\boldsymbol{VH}^T\right)_{ia}}{\left(\boldsymbol{WHH}^T\right)_{ia}}$$

Objective function guaranteed to monotonically decrease, but not necessarily converges to a local minimum.

Numerical algorithms

Cost functions are not simultaneously convex in both $\boldsymbol{W}$ and $\boldsymbol{H}$ ➜ local minima

Numerical techniques can be classified in three categories:

Numerical algorithms

Cost functions are not simultaneously convex in both $\boldsymbol{W}$ and $\boldsymbol{H}$ ➜ local minima

Numerical techniques can be classified in three categories:

1. **Gradient descent algorithms**: simplest to implement, but very slow to converge (if at all). Their convergence is very sensitive to the step size choice and a convergence supporting theory is missing;
2. **Multiplicative update rules**: extensively used since they are simple to implement, applicable to sparse matrices, and they were proposed as numerical algorithm for solving NMF by Lee and Seung 2001;

➥ **Alternating least squares algorithms** (Kim and Park 2007) : a least squares step is followed by another least squares step in an alternating fashion; very fast, quick to convergence, and give accurate factors.

Alternating least squares

Developed to solve the issues affecting multiplicative rules:

- Slow convergence ➜ alternating strategy

Alternating least squares

Developed to solve the issues affecting multiplicative rules:

- Slow convergence ➜ alternating strategy
- Locking property ➜ escaping from poor local minima

Alternating least squares

Developed to solve the issues affecting multiplicative rules:

- Slow convergence ➜ alternating strategy
- Locking property ➜ escaping from poor local minima

Basic ALS

1. Take $\boldsymbol{W}$ as fixed
2. Solve for $\boldsymbol{H}$ in the equation $\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H} = \boldsymbol{W}^T\boldsymbol{V}$
3. Set all negative elements in $\boldsymbol{H}$ to zero
4. Then, take $\boldsymbol{H}$ as fixed
5. Solves for $\boldsymbol{W}$ in the equation $\boldsymbol{H}\boldsymbol{H}^T\boldsymbol{W}^T = \boldsymbol{H}\boldsymbol{V}^T$
6. Set all the negative elements in $\boldsymbol{W}$ to zero.

Alternating least squares

Developed to solve the issues affecting multiplicative rules:

- Slow convergence ➜ alternating strategy
- Locking property ➜ escaping from poor local minima

Basic ALS

1. Take $\boldsymbol{W}$ as fixed
2. Solve for $\boldsymbol{H}$ in the equation $\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H} = \boldsymbol{W}^T\boldsymbol{V}$
3. Set all negative elements in $\boldsymbol{H}$ to zero
4. Then, take $\boldsymbol{H}$ as fixed
5. Solves for $\boldsymbol{W}$ in the equation $\boldsymbol{H}\boldsymbol{H}^T\boldsymbol{W}^T = \boldsymbol{H}\boldsymbol{V}^T$
6. Set all the negative elements in $\boldsymbol{W}$ to zero.

Computationally demanding due to additional nonnegativity constraint on least squares.

Initialization

NMF algorithms need to be initialized with values for $\boldsymbol{W}^{(0)}$ and/or $\boldsymbol{H}^{(0)}$.

Initialization

NMF algorithms need to be initialized with values for $\boldsymbol{W}^{(0)}$ and/or $\boldsymbol{H}^{(0)}$.

Standard NMF relies on **random initialization** ➜ very simple, but requires multiple runs with different starting points (high computation times).

Initialization

NMF algorithms need to be initialized with values for $\boldsymbol{W}^{(0)}$ and/or $\boldsymbol{H}^{(0)}$.

Standard NMF relies on **random initialization** ➜ very simple, but requires multiple runs with different starting points (high computation times).

Several seeding methods have been proposed, based on:

- **Independent component analysis**
- **Nonnegative double singular value decomposition**
- **Clustering techniques**

In practice, one may use several different initializations via a Monte Carlo approach, and keep the best solution obtained.

Factorization rank

- Too high values ➜ potential risk of overfitting
- Too low values ➜ potential risk of misinterpretation

Factorization rank

- Too high values ➜ potential risk of overfitting
- Too low values ➜ potential risk of misinterpretation

Consider the following application to image recognition:

Factorization rank

- Too high values ➜ potential risk of overfitting
- Too low values ➜ potential risk of misinterpretation

Consider the following application to image recognition:

Figure 1: *Madonna del Cardellino* in grey-scale format; Target matrix rank = 400

Factorization rank

Approximated paintings with different factorization ranks: **5**, 20, 40, 80.

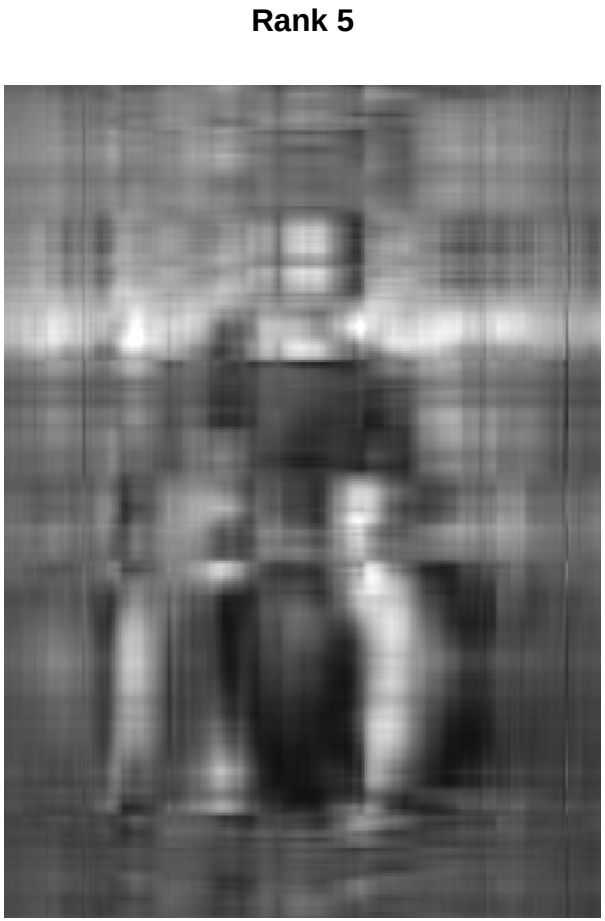


Figure 2: *Factorization rank=5*

Factorization rank

Approximated paintings with different factorization ranks: 5, **20**, 40, 80.

Figure 3: *Factorization rank=20*

Factorization rank

Approximated paintings with different factorization ranks: 5, 20, **40**, 80.

Figure 4: *Factorization rank=40*

Factorization rank

Approximated paintings with different factorization ranks: 5, 20, 40, **80**.

Figure 5: *Factorization rank=80*

Factorization rank

Unless one has prior knowledge on domain theory, a common way to choose r is through **trial and error**.

Factorization rank

Unless one has prior knowledge on domain theory, a common way to choose r is through **trial and error**.

However, several approaches have been proposed:

- Apply **SVD** and look at the decay of singular values of the input data matrix;

Factorization rank

Unless one has prior knowledge on domain theory, a common way to choose r is through **trial and error**.

However, several approaches have been proposed:

- Apply **SVD** and look at the decay of singular values of the input data matrix;
- Take the first value of r for which the **cophenetic coefficient** starts decreasing;

Factorization rank

Unless one has prior knowledge on domain theory, a common way to choose r is through **trial and error**.

However, several approaches have been proposed:

- Apply **SVD** and look at the decay of singular values of the input data matrix;
- Take the first value of r for which the **cophenetic coefficient** starts decreasing;
- Choose the first value where the **RSS curve** presents an inflection point;

Factorization rank

Unless one has prior knowledge on domain theory, a common way to choose r is through **trial and error**.

However, several approaches have been proposed:

- Apply **SVD** and look at the decay of singular values of the input data matrix;
- Take the first value of r for which the **cophenetic coefficient** starts decreasing;
- Choose the first value where the **RSS curve** presents an inflection point;
- Consider smallest value at which the decrease in the **RSS** is lower than the decrease of analogous curve from **random data**.

A unique decomposition?

- *Find a simplicial convex cone* $C_{\boldsymbol{W}} = \left\{ \sum_{a=1}^{r} \lambda_a \boldsymbol{w}_a, \lambda_a \geq 0 \right\}$ *contained in the positive orthant and containing data points.*

A unique decomposition?

❖ *Find a simplicial convex cone* $C_{\boldsymbol{W}} = \left\{ \sum_{a=1}^{r} \lambda_a \boldsymbol{w}_a, \lambda_a \geq 0 \right\}$ *contained in the positive orthant and containing data points.*

If data values are strictly positive:

$$v_{i\mu} \geq \epsilon \quad \forall \epsilon > 0$$

the column vectors of $\boldsymbol{V}$ are well inside the interior of positive orthant of $\mathbb{R}^n$, and there are many simplicial cones containing the data (Donoho and Stodden 2004).

A unique decomposition?

❖ *Find a simplicial convex cone* $C_{\boldsymbol{W}} = \left\{ \sum_{a=1}^{r} \lambda_a \boldsymbol{w}_a, \lambda_a \geq 0 \right\}$ *contained in the positive orthant and containing data points.*

If data values are strictly positive:

$$v_{i\mu} \geq \epsilon \quad \forall \epsilon > 0$$

the column vectors of $\boldsymbol{V}$ are well inside the interior of positive orthant of $\mathbb{R}^n$, and there are many simplicial cones containing the data (Donoho and Stodden 2004).

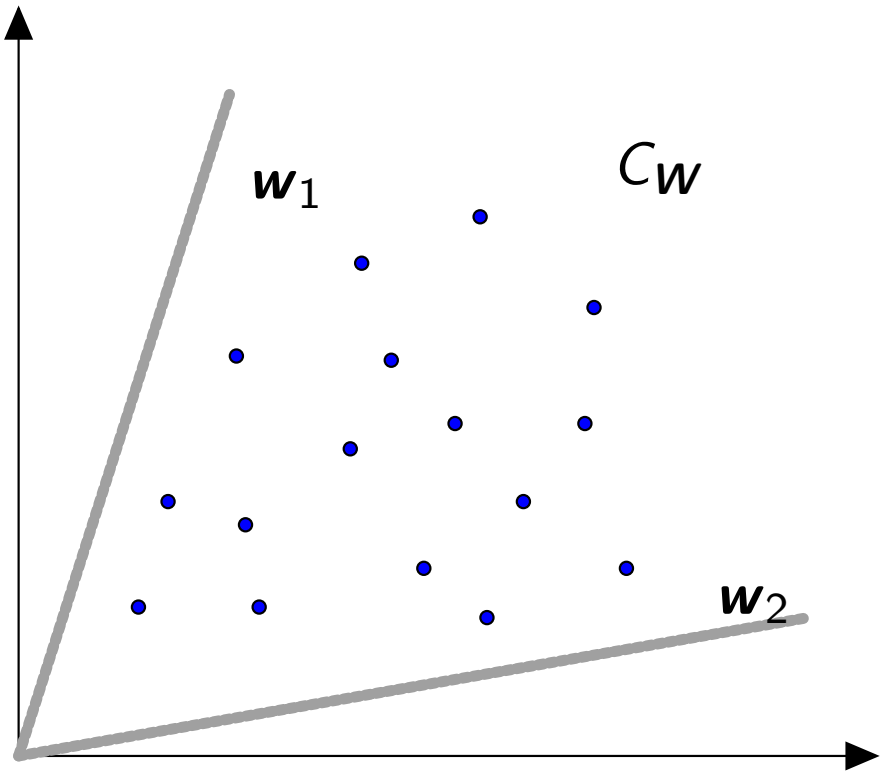


Figure 6: Non-uniqueness of nonnegative matrix factorization

A unique decomposition?

❖ *Find a simplicial convex cone* $C_{\boldsymbol{W}} = \left\{ \sum_{a=1}^{r} \lambda_a \boldsymbol{w}_a, \lambda_a \geq 0 \right\}$ *contained in the positive orthant and containing data points.*

If data values are strictly positive:

$$v_{i\mu} \geq \epsilon \quad \forall \epsilon > 0$$

the column vectors of $\boldsymbol{V}$ are well inside the interior of positive orthant of $\mathbb{R}^n$, and there are many simplicial cones containing the data (Donoho and Stodden 2004).

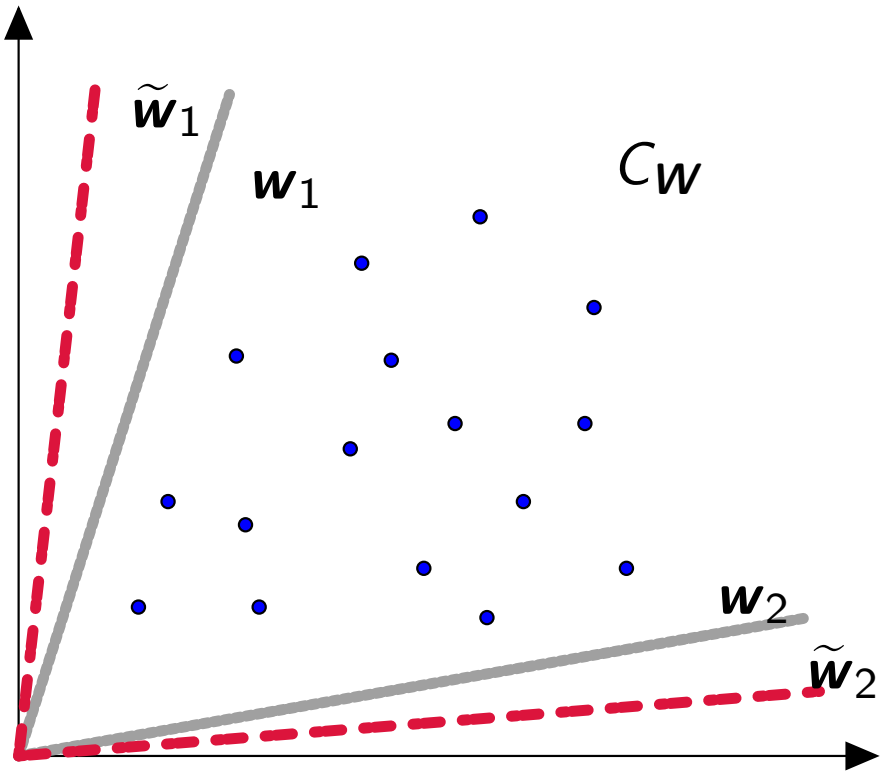


Figure 6: Non-uniqueness of nonnegative matrix factorization

A unique decomposition?

- *Find a simplicial convex cone* $C_{\boldsymbol{W}} = \left\{ \sum_{a=1}^{r} \lambda_a \boldsymbol{w}_a, \lambda_a \geq 0 \right\}$ *contained in the positive orthant and containing data points.*

Under such strict positivity condition, there are many distinct factorizations.

A unique decomposition?

- *Find a simplicial convex cone* $C_{\boldsymbol{W}} = \left\{ \sum_{a=1}^{r} \lambda_a \boldsymbol{w}_a, \lambda_a \geq 0 \right\}$ *contained in the positive orthant and containing data points.*

Under such strict positivity condition, there are many distinct factorizations.

NMF is an ill-posed problem

Any nonnegative invertible matrix $\boldsymbol{Q}$ satisfying $\begin{cases} \widetilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{Q} \geq 0 \\ \widetilde{\boldsymbol{H}} = \boldsymbol{Q}^{-1}\boldsymbol{H} \geq 0 \end{cases}$

generates an equivalent factorization:

$$\widetilde{\boldsymbol{W}}\widetilde{\boldsymbol{H}} = \boldsymbol{W}\boldsymbol{H}$$

A unique decomposition?

❖ *Find a simplicial convex cone* $C_{\boldsymbol{W}} = \left\{ \sum_{a=1}^{r} \lambda_a \boldsymbol{w}_a, \lambda_a \geq 0 \right\}$ *contained in the positive orthant and containing data points.*

Under such strict positivity condition, there are many distinct factorizations.

NMF is an ill-posed problem

Any nonnegative invertible matrix $\boldsymbol{Q}$ satisfying $\begin{cases} \widetilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{Q} \geq 0 \\ \widetilde{\boldsymbol{H}} = \boldsymbol{Q}^{-1}\boldsymbol{H} \geq 0 \end{cases}$

generates an equivalent factorization:

$$\widetilde{\boldsymbol{W}}\widetilde{\boldsymbol{H}} = \boldsymbol{W}\boldsymbol{H}$$

➜ Impose additional constraints to confine the feasible solution set.

Extensions

Over the years, the standard NMF model has been extended to explicitly include auxiliary constraints (incorporated in the objective function) on $\boldsymbol{W}$ and/or $\boldsymbol{H}$ to improve the factorization.

Extensions

Over the years, the standard NMF model has been extended to explicitly include auxiliary constraints (incorporated in the objective function) on $\boldsymbol{W}$ and/or $\boldsymbol{H}$ to improve the factorization.

- **Smoothness**: regularizes computed solutions in presence of noise in data;

Extensions

Over the years, the standard NMF model has been extended to explicitly include auxiliary constraints (incorporated in the objective function) on $\boldsymbol{W}$ and/or $\boldsymbol{H}$ to improve the factorization.

- **Smoothness**: regularizes computed solutions in presence of noise in data;
- **Sparseness**: provides a clearer representation of data. Sparseness is defined as (Hoyer 2004):

$$\text{sparseness}(\boldsymbol{x}) = \frac{\sqrt{n} - \frac{\|\boldsymbol{x}\|_1}{\|\boldsymbol{x}\|_2}}{\sqrt{n} - 1}$$

Extensions

Over the years, the standard NMF model has been extended to explicitly include auxiliary constraints (incorporated in the objective function) on $\boldsymbol{W}$ and/or $\boldsymbol{H}$ to improve the factorization.

- **Smoothness**: regularizes computed solutions in presence of noise in data;
- **Sparseness**: provides a clearer representation of data. Sparseness is defined as (Hoyer 2004):

$$\text{sparseness}(\boldsymbol{x}) = \frac{\sqrt{n} - \frac{\|\boldsymbol{x}\|_1}{\|\boldsymbol{x}\|_2}}{\sqrt{n} - 1}$$

Some other extensions of standard NMF include:

1. *Non-smooth NMF (nsNMF);*
2. *NMF with offset;*
3. *Pattern-expression NMF*

Application to Text Mining

Text analysis learns from collections of text data to recover topics and classify documents.

Application to Text Mining

Text analysis learns from collections of text data to recover topics and classify documents.

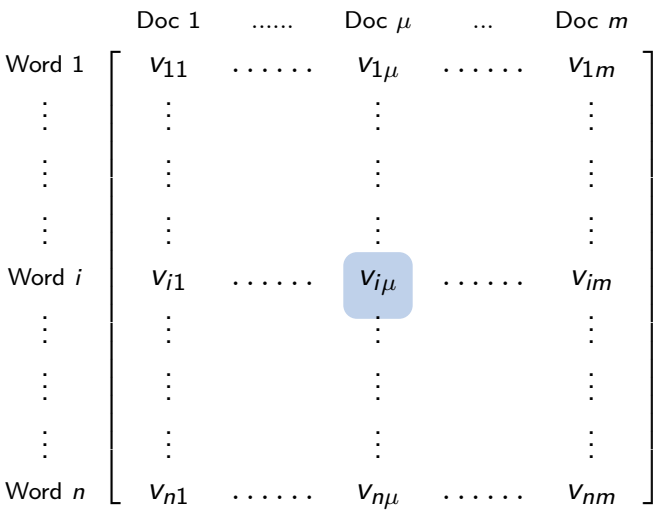
Term-document matrix

	Doc 1	.....	Doc μ	...	Doc m
Word 1	v_{11}	.....	$v_{1\mu}$	.....	v_{1m}
Word i	v_{i1}	.....	$v_{i\mu}$	.....	v_{im}
Word n	v_{n1}	.....	$v_{n\mu}$	.....	v_{nm}

Application to Text Mining

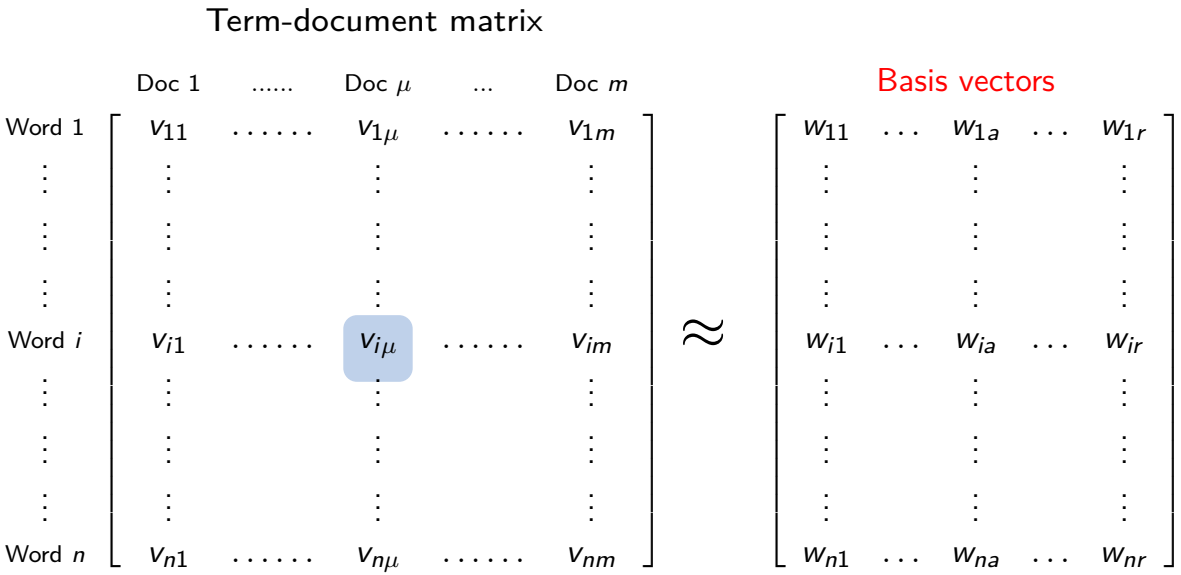
Text analysis learns from collections of text data to recover topics and classify documents.

Term-document matrix



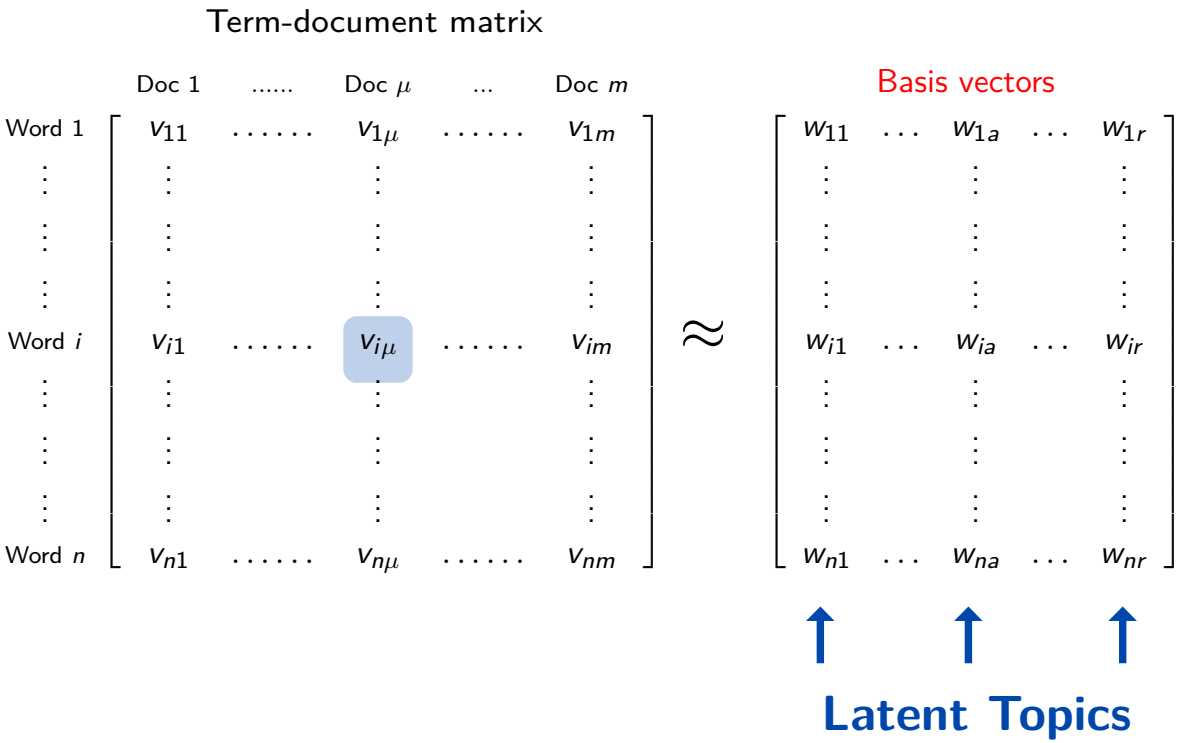
Application to Text Mining

Text analysis learns from collections of text data to recover topics and classify documents.



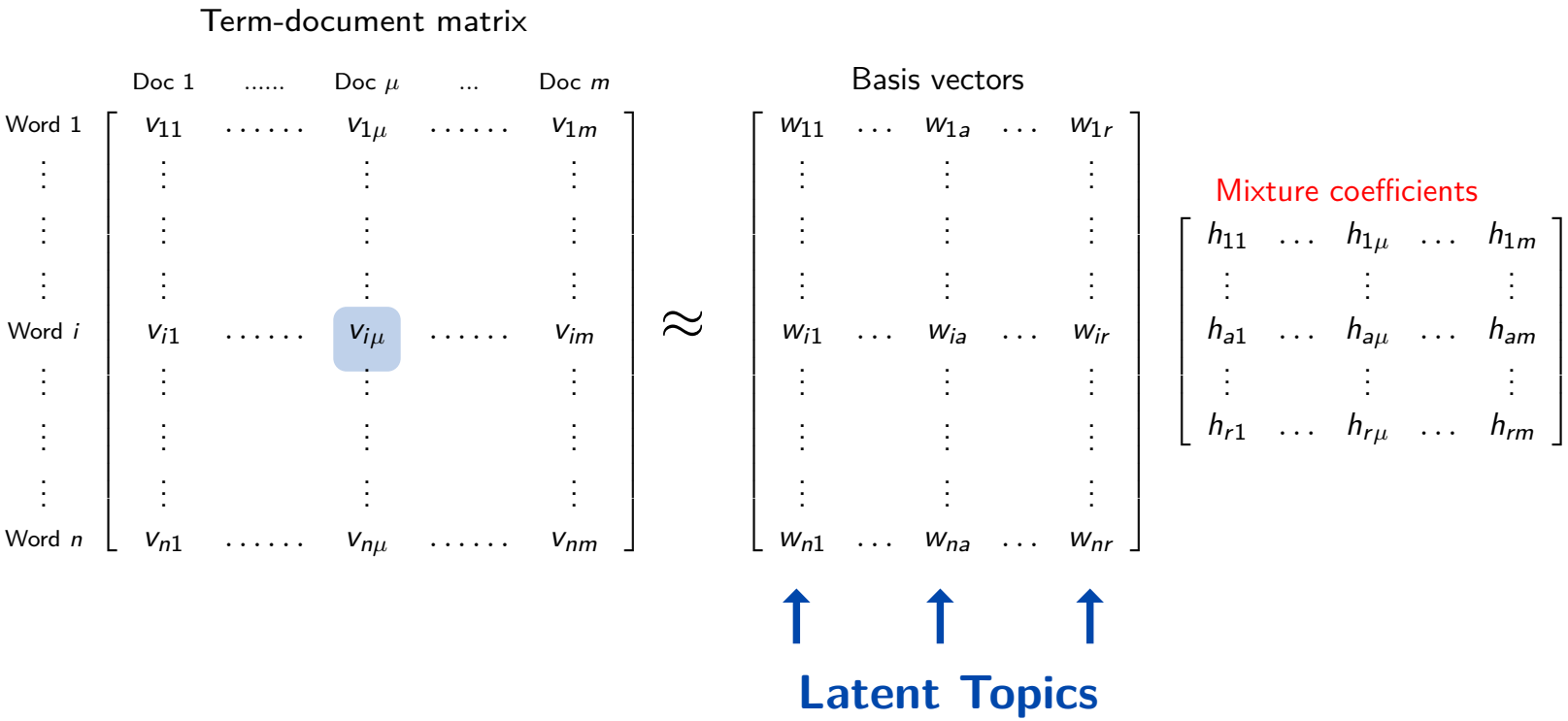
Application to Text Mining

Text analysis learns from collections of text data to recover topics and classify documents.



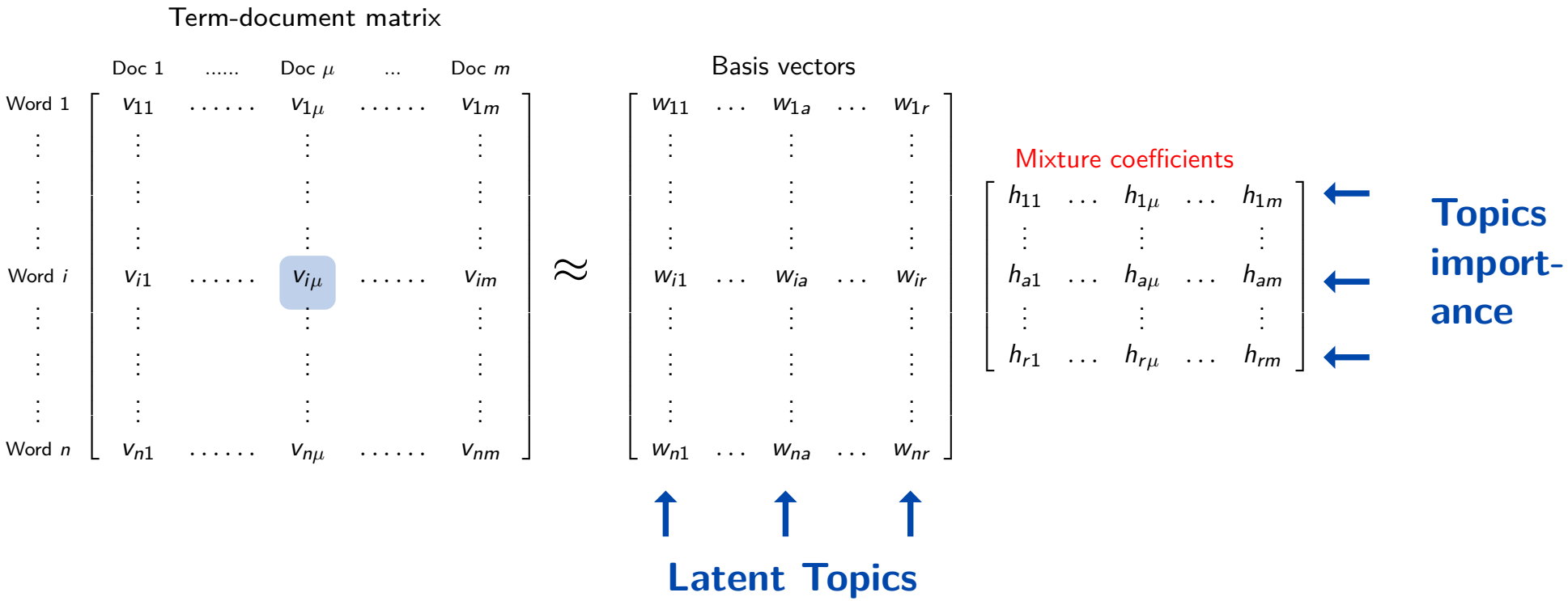
Application to Text Mining

Text analysis learns from collections of text data to recover topics and classify documents.



Application to Text Mining

Text analysis learns from collections of text data to recover topics and classify documents.



Analysis

1. *Analyzed texts*: complete published works by William Shakespeare (corpus composed by 182 docs);

Analysis

1. *Analyzed texts*: complete published works by William Shakespeare (corpus composed by 182 docs);
2. Variety of preprocessing operations;

Analysis

1. *Analyzed texts*: complete published works by William Shakespeare (corpus composed by 182 docs);
2. Variety of preprocessing operations;
3. Term-document matrix of ≈19000 words; sparseness degree ≈95%.

Analysis

1. *Analyzed texts*: complete published works by William Shakespeare (corpus composed by 182 docs);
2. Variety of preprocessing operations;
3. Term-document matrix of ≈19000 words; sparseness degree ≈95%.

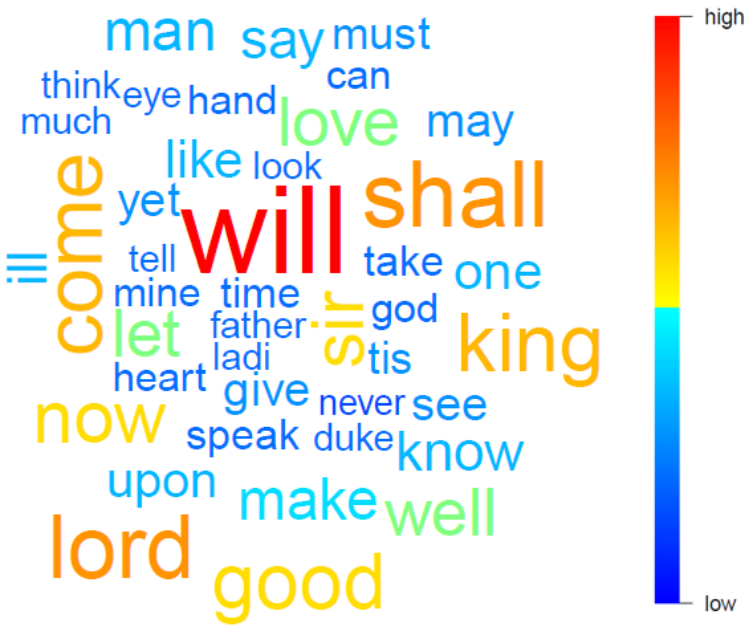


Figure 7: Word cloud (words occurring more than 1000 times)

Analysis

1. *Analyzed texts*: complete published works by William Shakespeare (corpus composed by 182 docs);
2. Variety of preprocessing operations;
3. Term-document matrix of ≈19000 words; sparseness degree ≈95%.
4. Removal of unfrequent terms ➜ 63×182 term-document matrix;

Analysis

1. *Analyzed texts*: complete published works by William Shakespeare (corpus composed by 182 docs);
2. Variety of preprocessing operations;
3. Term-document matrix of ≈19000 words; sparseness degree ≈95%.
4. Removal of unfrequent terms ➜ 63×182 term-document matrix;
5. A rank-3 NMF model fitted to the data using multiplicative update minimizing Kullback-Leibler divergence (model estimated 200 times, each time starting from different set of initial values);

Analysis

1. *Analyzed texts*: complete published works by William Shakespeare (corpus composed by 182 docs);
2. Variety of preprocessing operations;
3. Term-document matrix of ≈19000 words; sparseness degree ≈95%.
4. Removal of unfrequent terms ➜ 63×182 term-document matrix;
5. A rank-3 NMF model fitted to the data using multiplicative update minimizing Kullback-Leibler divergence (model estimated 200 times, each time starting from different set of initial values);
6. Factor matrices explored by means of heatmaps.

Topic recovery

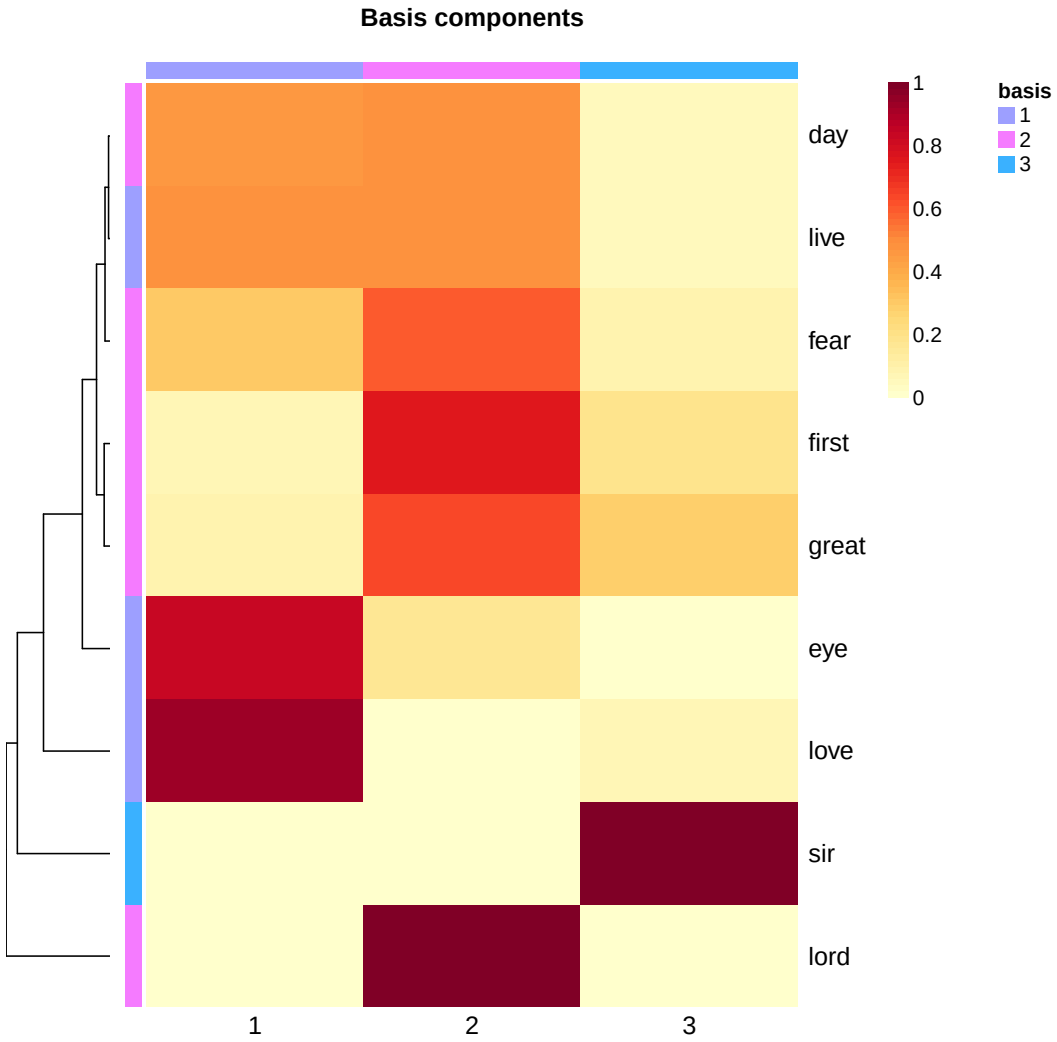


Figure 7: Heatmap of basis matrix (only relevant rows selected)

Topic recovery

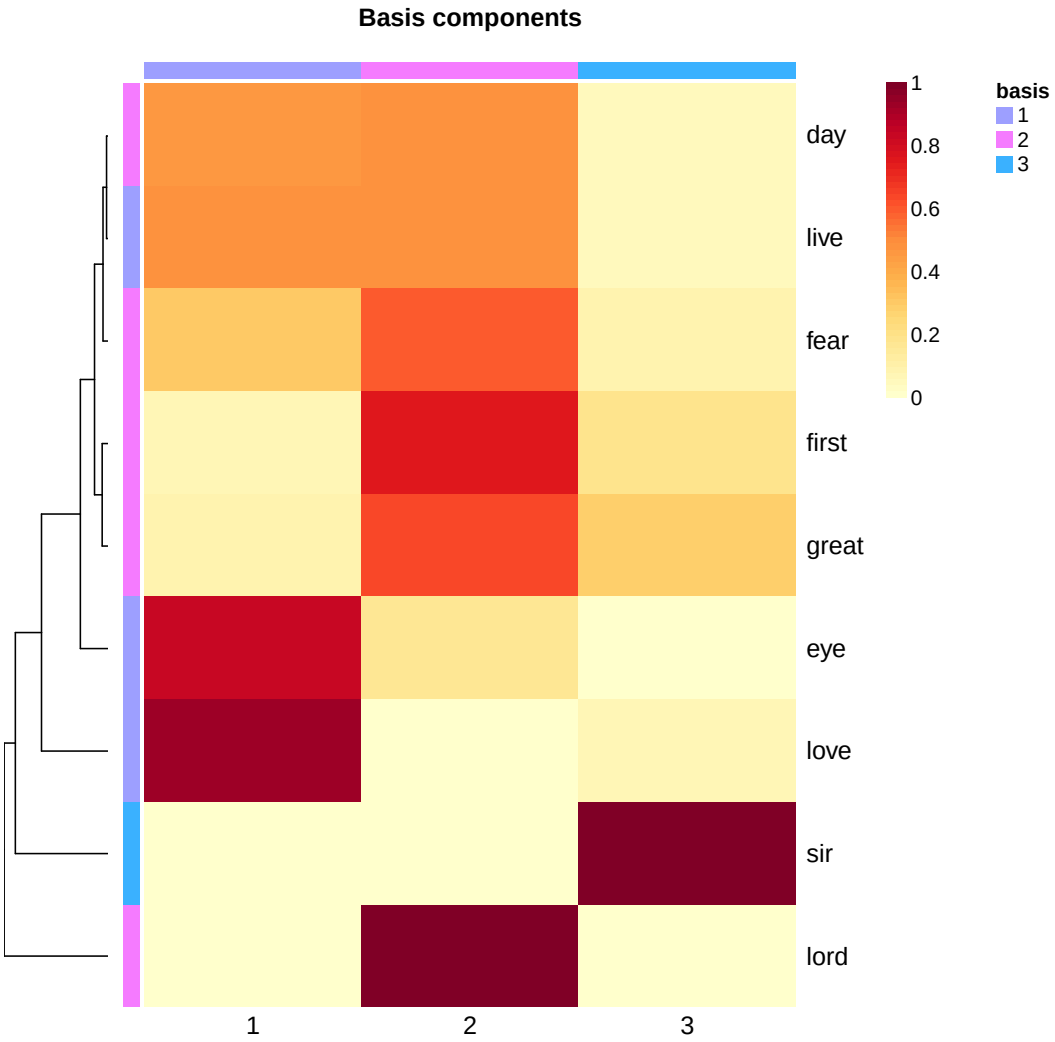


Figure 7: Heatmap of basis matrix (only relevant rows selected)

- First basis vector: entails to common topic of **love, beauty**, and **eternity**;

Topic recovery

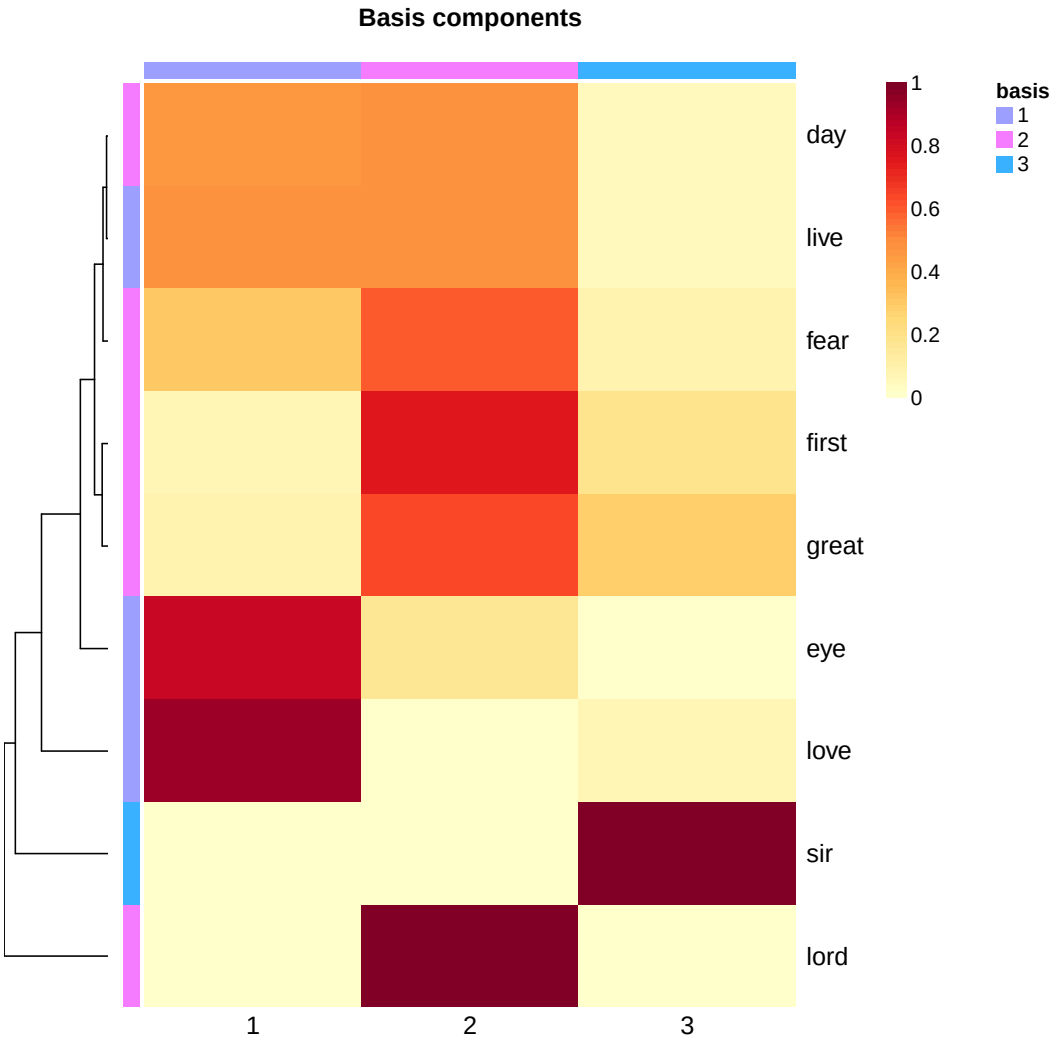


Figure 7: Heatmap of basis matrix (only relevant rows selected)

- First basis vector: entails to common topic of **love, beauty**, and **eternity**;
- Second basis vector: activated by words standing for various themes: **obeisance**, **grandeur**, but also **power** and **impending death**;

Topic recovery

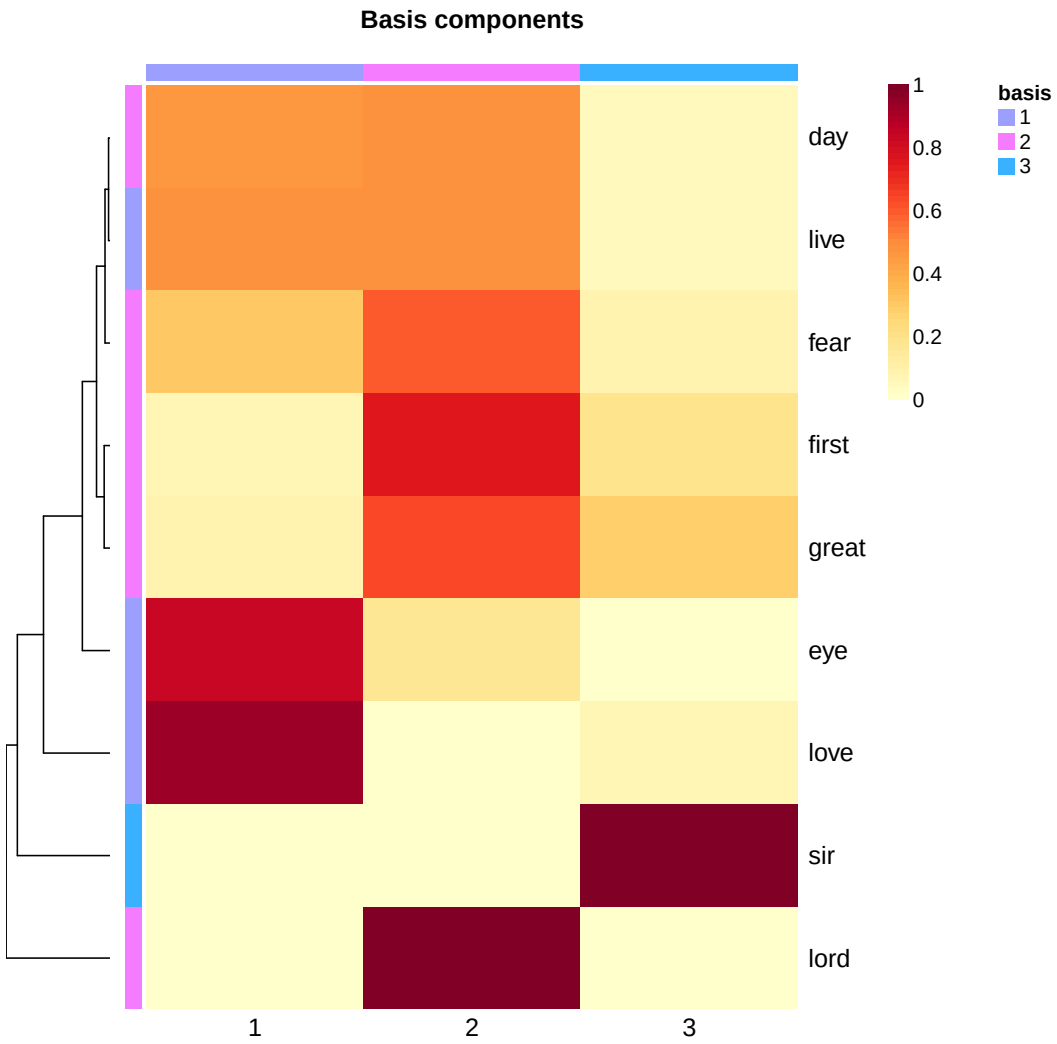


Figure 7: Heatmap of basis matrix (only relevant rows selected)

- First basis vector: entails to common topic of **love, beauty**, and **eternity**;
- Second basis vector: activated by words standing for various themes: **obeisance**, **grandeur**, but also **power** and **impending death**;
- Third basis component: related to comicality, **sarcasm** and **irony**.

Document classification

Allocate docs to basis component for which they have the highest coefficient.

Document classification

Allocate docs to basis component for which they have the highest coefficient.

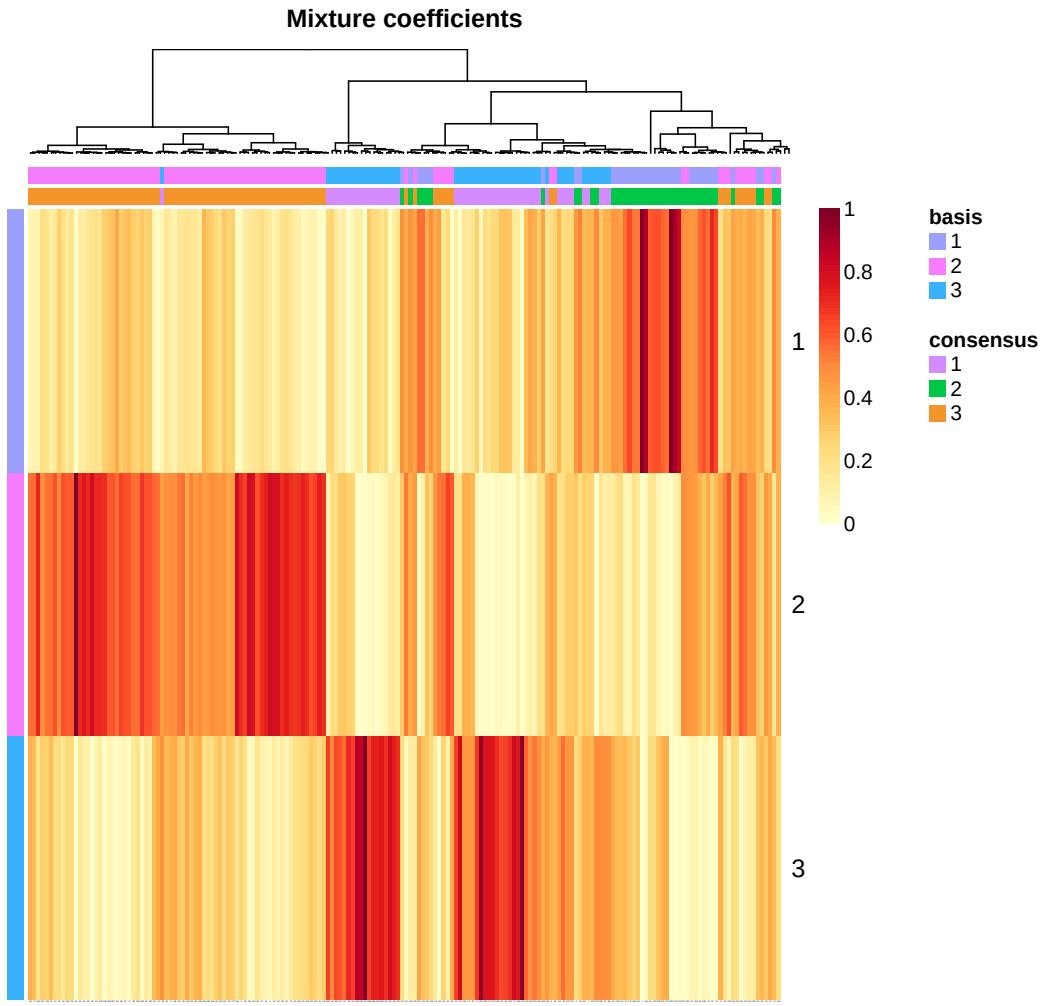


Figure 8: Mixture coefficients heatmap

Document classification

Allocate docs to basis component for which they have the highest coefficient.

Sonnets
Romeo and Juliet
Love's Labour's Lost
A Midsummer Night's Dream
The Two Gentlemen of Verona
A Lover's Complaint
The Tempest
. . .

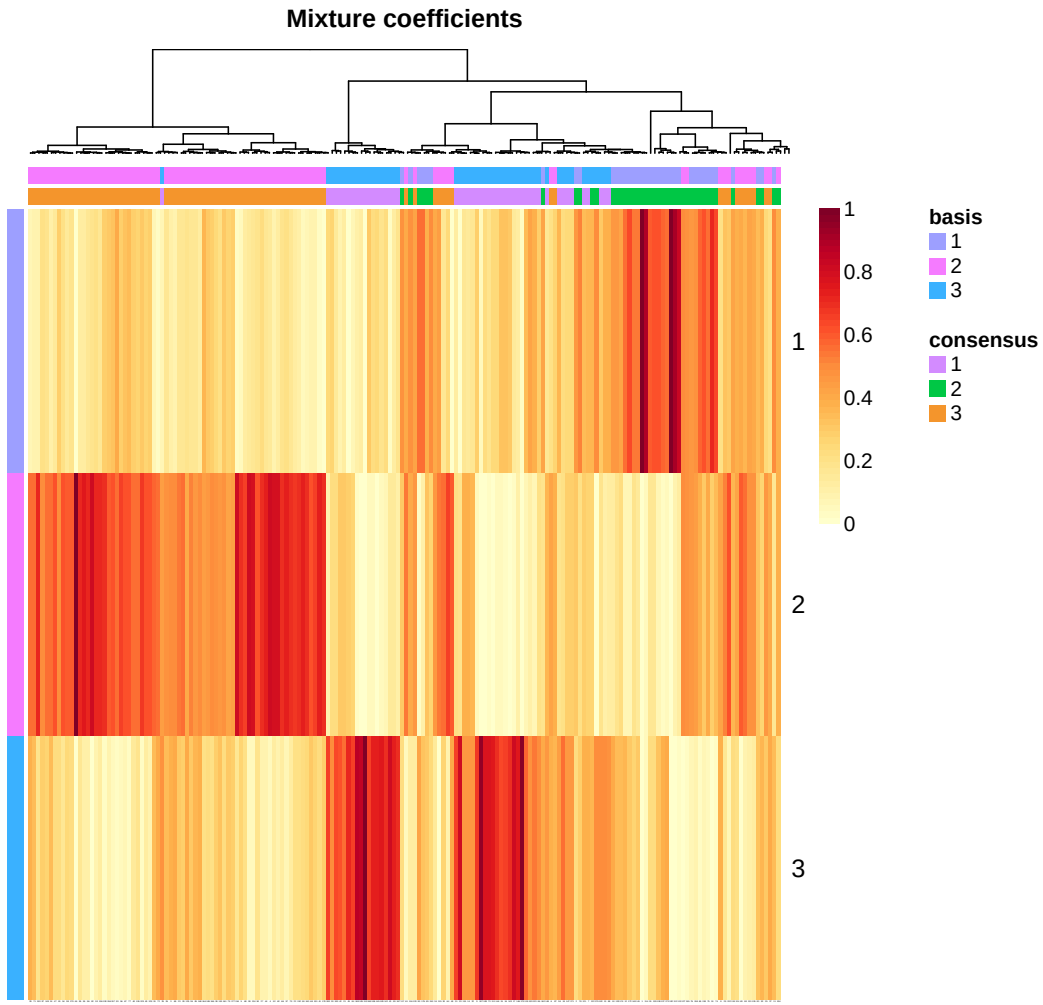


Figure 8: Mixture coefficients heatmap

Document classification

Allocate docs to basis component for which they have the highest coefficient.

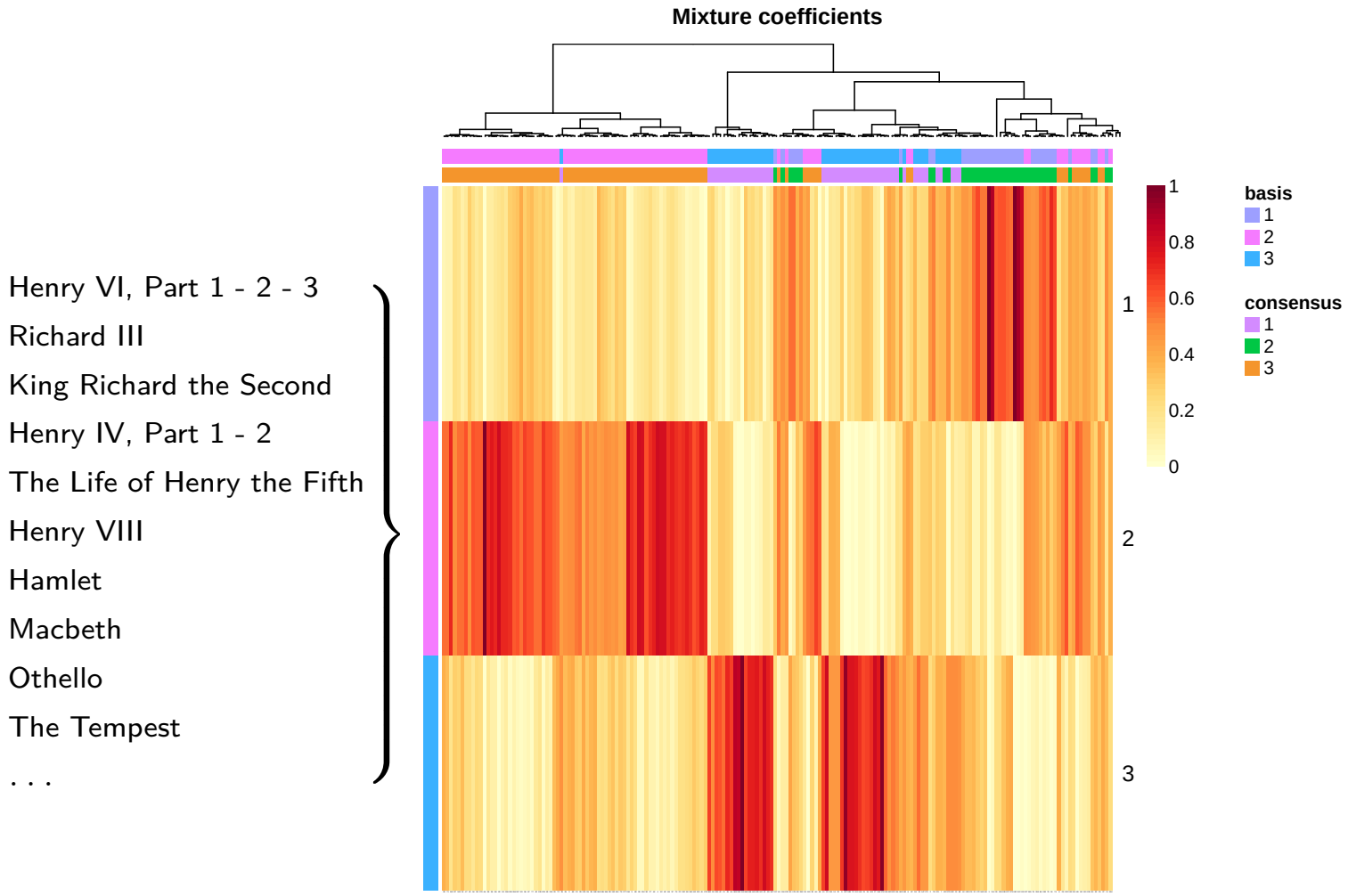


Figure 8: Mixture coefficients heatmap

Document classification

Allocate docs to basis component for which they have the highest coefficient.

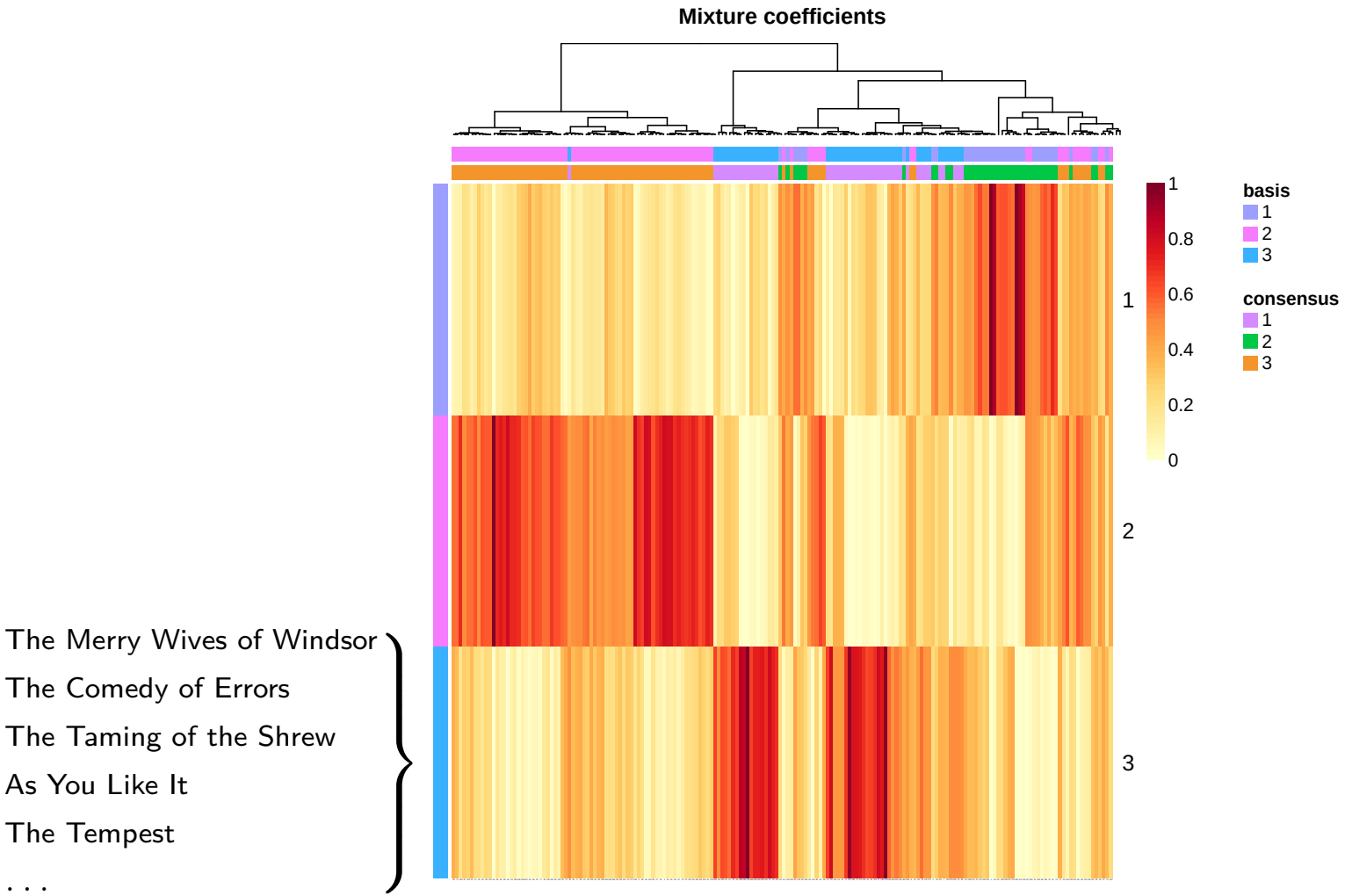


Figure 8: Mixture coefficients heatmap

Conclusion

NMF has made great strides and huge impact in the last few years and, by looking at the number of yearly citations received by founding paper of Lee and Seung 1999, it's very likely it will have a promising and bright future.

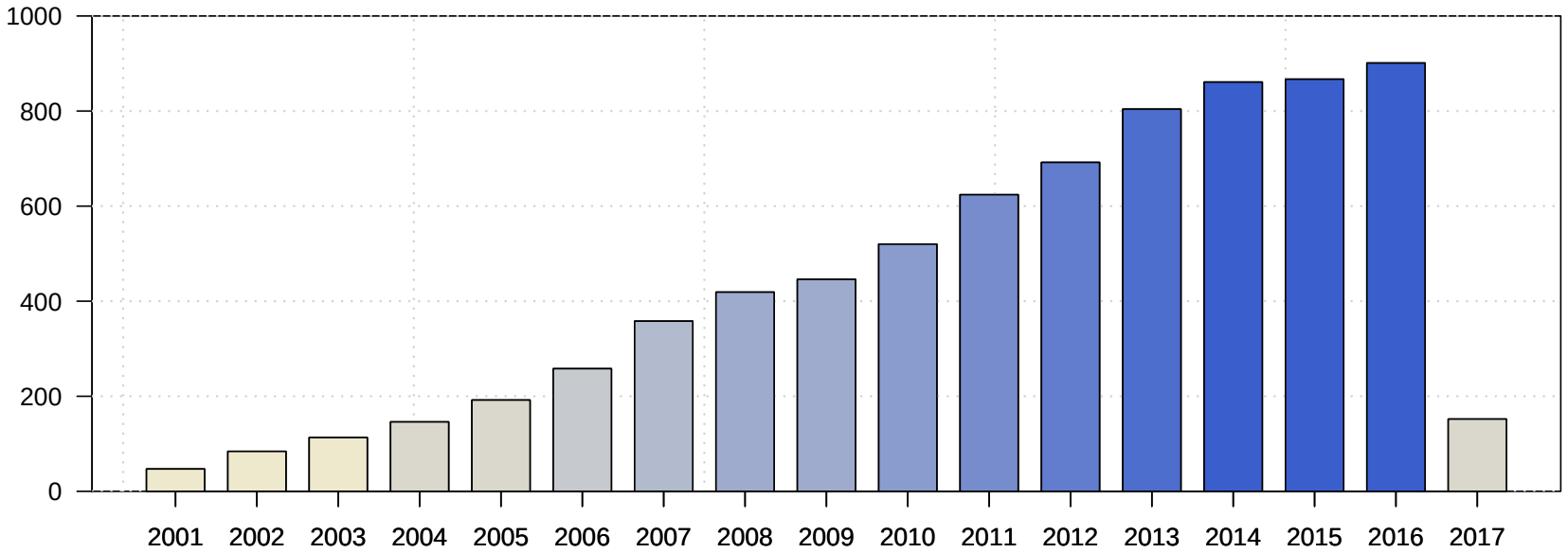


Figure 9: Citations per year of Lee and Seung 1999's paper that launched NMF

References

- Donoho, David L and Stodden, Victoria C (2004). 'When does non-negative matrix factorization give a correct decomposition into parts?' In: *Advances in neural information processing systems 16: proceedings of the 2003 conference*. Ed. by S Thrun, L Saul and B Schölkopf.
- Hoyer, Patrik O (2004). 'Non-negative matrix factorization with sparseness constraints'. In: *Journal of machine learning research* 5.Nov, pp. 1457–1469.
- Kim, Hyunsoo and Park, Haesun (2007). 'Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis'. In: *Bioinformatics* 23.12, pp. 1495–1502.
- Lee, Daniel D and Seung, H Sebastian (1999). 'Learning the parts of objects by non-negative matrix factorization'. In: *Nature* 401.6755, pp. 788–791.
- – (2001). 'Algorithms for non-negative matrix factorization'. In: *Advances in neural information processing systems*, pp. 556–562.
- Paatero, Pentti and Tapper, Unto (1994). 'Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values'. In: *Environmetrics* 5.2, pp. 111–126.