Machine Learning in High Dimension
IA317
Dimension Reduction

Thomas Bonald

2023 – 2024

TELECOM Paris
IP PARIS

High dimension

Data $=$ n samples, each with d features

$$X \in \mathbb{R}^{n \times d}$$

High dimension $= d >> 1$ (possibly larger than n)
Typically a **sparse** matrix

Examples

- Textual data (bags of words)
- Medical data
- Customer data

Dimension reduction

Data = n samples, each with d features

$$X \in \mathbb{R}^{n \times d}$$

Dimension reduction

$$X = \begin{bmatrix} \quad\quad\quad\quad \\ \\ \\ \end{bmatrix} \quad \rightarrow \quad Z = \begin{bmatrix} \quad \\ \\ \\ \end{bmatrix}$$

Objective: Find a **dense** representation of data with meaningful **distances** (e.g., Euclidean or cosine similarity). Useful for:

- **classification / regression** → nearest neighbors, SVM
- **clustering** → k-means, Ward
- **visualization** → UMAP, TSNE

Feature selection

Select the k most important features $j_1, \ldots, j_k$ of data X, like

- most **correlated** features
- features of highest **statistical dependence**
- features of highest **mutual information**

with respect to the labels y

Feature selection

$$X = \begin{bmatrix} \qquad\qquad\qquad \end{bmatrix} \quad y = \begin{bmatrix} \; \end{bmatrix} \quad \rightarrow \quad Z = \begin{bmatrix} \quad \end{bmatrix}$$

Random projection

Data = n samples, each with d features

$$X \in \mathbb{R}^{n \times d}$$

Projection over k **random vectors** (usually Gaussian):

$$V = (v_1, \ldots, v_k) \in \mathbb{R}^{d \times k}$$

Random projection

Data $=$ n samples, each with d features

$$X \in \mathbb{R}^{n \times d}$$

Projection over k **random vectors** (usually Gaussian):

$$V = (v_1, \ldots, v_k) \in \mathbb{R}^{d \times k}$$

Random projection

$$X = \begin{bmatrix} \quad\quad\quad\quad \\ \\ \\ \end{bmatrix} \quad \rightarrow \quad Z = XV = \begin{bmatrix} \quad \\ \\ \\ \end{bmatrix}$$

Note: Pairwises Euclidean distances preserved for k large enough
cf. **Johnson-Lindenstrauss** lemma

Matrix factorization

Data = n samples, each with d features

$$X \in \mathbb{R}^{n \times d}$$

Principle

$$X = \begin{bmatrix} \quad & \quad & \quad \end{bmatrix} \approx \begin{bmatrix} \quad \end{bmatrix} \begin{bmatrix} \quad & \quad \end{bmatrix} \quad \rightarrow \quad Z = \begin{bmatrix} \quad \end{bmatrix}$$

Overview

3 main techniques for dimension reduction:

1. **Feature selection**
   $\rightarrow$ Supervised learning X, y
2. **Random projection**
   $\rightarrow$ No learning $\emptyset$
3. **Matrix factorization**
   $\rightarrow$ Unsupervised learning X

Inference

Train set $=$ n samples, each with d features

$$X_{\text{train}} \in \mathbb{R}^{n \times d} \quad \rightarrow \quad Z_{\text{train}} \in \mathbb{R}^{n \times k}$$

Question

How to reduce the dimension of the **test set** $X_{\text{test}} \rightarrow Z_{\text{test}}$ so that distances between Z_{train} and Z_{test} make sense?

Inference

Train set $=$ n samples, each with d features

$$X_{\mathrm{train}} \in \mathbb{R}^{n \times d} \quad \rightarrow \quad Z_{\mathrm{train}} \in \mathbb{R}^{n \times k}$$

Question

How to reduce the dimension of the **test set** $X_{\mathrm{test}} \rightarrow Z_{\mathrm{test}}$ so that distances between Z_{train} and Z_{test} make sense?

1. **Feature selection**
   $\rightarrow$ Same features $\quad j_1, \ldots, j_k \in \{1, \ldots, d\}$

Inference

Train set $=$ n samples, each with d features

$$X_{\text{train}} \in \mathbb{R}^{n \times d} \quad \rightarrow \quad Z_{\text{train}} \in \mathbb{R}^{n \times k}$$

Question

How to reduce the dimension of the **test set** $X_{\text{test}} \rightarrow Z_{\text{test}}$
so that distances between Z_{train} and Z_{test} make sense?

1. **Feature selection**
   $\rightarrow$ Same features $\quad j_1, \ldots, j_k \in \{1, \ldots, d\}$
2. **Random projection**
   $\rightarrow$ Same vectors $\quad v_1, \ldots, v_k \in \mathbb{R}^d$

Inference

Train set $=$ n samples, each with d features

$$X_{\text{train}} \in \mathbb{R}^{n \times d} \quad \rightarrow \quad Z_{\text{train}} \in \mathbb{R}^{n \times k}$$

Question

How to reduce the dimension of the **test set** $X_{\text{test}} \rightarrow Z_{\text{test}}$ so that distances between Z_{train} and Z_{test} make sense?

1. **Feature selection**
   $\rightarrow$ Same features $\qquad j_1, \ldots, j_k \in \{1, \ldots, d\}$
2. **Random projection**
   $\rightarrow$ Same vectors $\qquad v_1, \ldots, v_k \in \mathbb{R}^d$
3. **Matrix factorization**
   $\rightarrow$? $\qquad X_{\text{train}} \approx \begin{bmatrix} \quad \\ \\ \\ \end{bmatrix} \begin{bmatrix} \qquad\qquad\qquad \end{bmatrix}$

Outline

Focus on 2 **matrix factorization** techniques:

1. Singular Value Decomposition (SVD)
   $\leftrightarrow$ Principal Component Analysis (PCA)
2. Non-negative Matrix Factorization (NMF)

Singular value

Let $X \in \mathbb{R}^{n \times d}$

Definition

We say that $\sigma \geq 0$ is a **singular value** of X if there exist unit vectors $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^d$ such that

$$\begin{aligned} Xv &= \sigma u \\ X^T u &= \sigma v \end{aligned}$$

The vectors u and v are left and right **singular vectors** for σ

Singular value

Let $X \in \mathbb{R}^{n \times d}$

Definition

We say that $\sigma \geq 0$ is a **singular value** of X if there exist unit vectors $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^d$ such that

$$\begin{aligned} Xv &= \sigma u \\ X^T u &= \sigma v \end{aligned}$$

The vectors u and v are left and right **singular vectors** for σ

Note: The vectors u and v are respective **eigenvectors** of XX^T and X^TX for the **eigenvalue** σ^2

Interpretation

Data $=$ n samples, each with d features

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} \in \mathbb{R}^{n \times d}$$

Projection

$$Xv = \sigma u$$

The **projection** of data X over the unit vector v has **norm** σ and **direction** u in $\mathbb{R}^n$

Singular value decomposition

Let $X \in \mathbb{R}^{n \times d}$ of rank r

Theorem

There exist $U = (u_1, \ldots, u_r) \in \mathbb{R}^{n \times r}$, $V = (v_1, \ldots, v_r) \in \mathbb{R}^{d \times r}$ and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ such that

$$X = \left[\quad \right] = \left[\quad \right] \Sigma \left[\quad \right] = U \Sigma V^T$$

with

$$U^T U = I_r \quad V^T V = I_r \quad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$$

The matrices U and V are orthonormal bases of left and right **singular vectors** for the singular values $\sigma_1, \ldots, \sigma_r$.

Proof: Spectral theorem applied to either XX^T or $X^T X$.

Interpretation

Data $=$ n samples, each with d features

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} \in \mathbb{R}^{n \times d}$$

Projection

$$XV = U\Sigma$$

Interpretation

Data $=$ n samples, each with d features

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} \in \mathbb{R}^{n \times d}$$

Projection

$$XV = U\Sigma$$

The **projection** of data X over the unit vectors $v_1, \ldots, v_r$ gives vectors of **norms** $\sigma_1, \ldots, \sigma_r$ and **orthogonal directions** $u_1, \ldots, u_r$

Top right singular vector

Let $X \in \mathbb{R}^{n \times d}$

Property

The top right singular vector is the direction of **highest inertia**:

$$v_1 = \arg \max_{v:||v||=1} ||Xv||^2$$

Top right singular vector

Let $X \in \mathbb{R}^{n \times d}$

Property

The top right singular vector is the direction of **highest inertia**:

$$v_1 = \arg\max_{v:||v||=1} ||Xv||^2$$

Note: If X is centered, in the sense that

$$1^T X = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} \left[\quad \quad \quad \right] = 0$$

v_1 is the direction of **highest variance** $\rightarrow$ Principal Component

Top right singular vectors

Let $X \in \mathbb{R}^{n \times d}$

Property

The top-k right singular vectors are the **orthogonal** directions of **highest inertia**:

$$v_1, \dots, v_k = \arg \max_{\substack{V \in \mathbb{R}^{d \times k} \\ V^T V = I}} ||XV||^2$$

Top right singular vectors

Let $X \in \mathbb{R}^{n \times d}$

Property

The top-k right singular vectors are the **orthogonal** directions of **highest inertia**:

$$v_1, \dots, v_k = \arg \max_{\substack{V \in \mathbb{R}^{d \times k} \\ V^T V = I}} ||XV||^2$$

Note: If X is centered, $v_1, \dots, v_k$ are the directions of **highest variance** $\rightarrow$ Principal Components

Principal Component Analysis

PCA = SVD **after** centering

$$X \quad \rightarrow \quad X - \frac{11^T}{n} X$$

The directions (= principal components) can be interpreted as the directions of **highest variance**

Warning

If X is a **sparse** matrix, its centered version is no longer sparse!

Dimension reduction by SVD

Data $=$ n samples, each with d features

$$X \in \mathbb{R}^{n \times d}$$

1. SVD

$$X = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \Sigma \begin{bmatrix} & & & & \\ & & & & \end{bmatrix} = U \Sigma V^T$$

Dimension reduction by SVD

Data $=$ n samples, each with d features

$$X \in \mathbb{R}^{n \times d}$$

1. SVD

$$X = \begin{bmatrix} \quad & \quad \\ \\ \\ \end{bmatrix} \Sigma \begin{bmatrix} \quad & \quad & \quad & \quad \\ \\ \end{bmatrix} = U \Sigma V^T$$

2. Projection

Projection on the **top-k right singular** vectors

$$Z = X V_k$$

Inference

Train set = n samples, each with d features

$$X_{\text{train}} \in \mathbb{R}^{n \times d}$$

1. SVD → learning

$$X_{\text{train}} = \begin{bmatrix} & \\ & \\ & \end{bmatrix} \Sigma \begin{bmatrix} & & & & \end{bmatrix} = U \Sigma V^T$$

2. Projection → inference

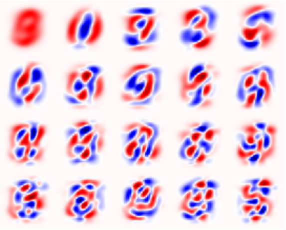
Projection on the **top-k right singular vectors** (of the **train set**)

$$Z_{\text{train}} = X_{\text{train}} V_k$$
$$Z_{\text{test}} = X_{\text{test}} V_k$$

Example: MNIST

$X \in \{0, \dots, 255\}^{n \times d}$
$n = 10,000$ samples
$d = 28 \times 28 = 784$

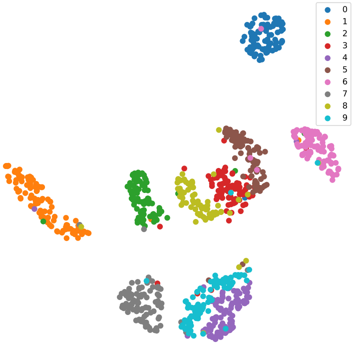
Samples



Singular vectors
$v_1, \dots, v_{20}$

Example: MNIST

Projection on the first 20 **right singular vectors**
Visualization of 1,000 samples



Train set

Test set

Low-rank approximation

Let $X \in \mathbb{R}^{n \times d}$

Definition

We say that $\hat{X}$ is the **best rank-k approximation** of X if

$$\hat{X} = \arg \min_{M:\text{rank}(M)=k} ||X - M||^2$$

with $|| \cdot ||$ the Frobenius norm (= Euclidean norm for matrices)

Low-rank approximation

Let $X \in \mathbb{R}^{n \times d}$

Definition

We say that $\hat{X}$ is the **best rank-k approximation** of X if

$$\hat{X} = \arg \min_{M:\text{rank}(M)=k} ||X - M||^2$$

with $|| \cdot ||$ the Frobenius norm (= Euclidean norm for matrices)

Theorem

For any $k \leq r$, the **best rank-k approximation** of X is

$$\hat{X} = U_k \Sigma_k {V_k}^T$$

with U_k, V_k, Σ_k the **restrictions** to the top k singular values

Approximation error

Let $X \in \mathbb{R}^{n \times d}$

Corollary

For any $k \leq r$, the minimum **square error** of a rank-k approximation of X is

$$||X - \hat{X}||^2 = \sum_{k < l \leq r} \sigma_l^2$$

Approximation error

Let $X \in \mathbb{R}^{n \times d}$

Corollary

For any $k \leq r$, the minimum **square error** of a rank-k approximation of X is

$$||X - \hat{X}||^2 = \sum_{k < l \leq r} \sigma_l^2$$

Notes:

- If $k = 0$ then $\hat{X} = 0$ and $||X||^2 = \sum_{l=1}^{r} \sigma_l^2$

Approximation error

Let $X \in \mathbb{R}^{n \times d}$

Corollary

For any $k \leq r$, the minimum **square error** of a rank-k approximation of X is

$$||X - \hat{X}||^2 = \sum_{k<l\leq r} \sigma_l^2$$

Notes:

- If $k = 0$ then $\hat{X} = 0$ and $||X||^2 = \sum_{l=1}^{r} \sigma_l^2$
- If $k = r$ then $\hat{X} = X$

Approximation error

Let $X \in \mathbb{R}^{n \times d}$

Corollary

For any $k \leq r$, the minimum **square error** of a rank-k approximation of X is

$$||X - \hat{X}||^2 = \sum_{k < l \leq r} \sigma_l^2$$

Notes:

- If $k = 0$ then $\hat{X} = 0$ and $||X||^2 = \sum_{l=1}^{r} \sigma_l^2$
- If $k = r$ then $\hat{X} = X$
- If $0 < k < r$ then $||X - \hat{X}||^2 = ||X||^2 - \sum_{l=1}^{k} \sigma_l^2$

Outline

Focus on 2 matrix factorization techniques:

1. Singular Value Decomposition (SVD)
   $\leftrightarrow$ Principal Component Analysis (PCA)
2. **Non-negative Matrix Factorization** (NMF)

Non-negative matrix factorization

Data $=$ n samples, each with d **non-negative** features

$$X \in \mathbb{R}^{n\times d} \quad X \geq 0$$

NMF

$$X \approx WH = \begin{bmatrix} \quad \end{bmatrix} \begin{bmatrix} \qquad \end{bmatrix} \quad \rightarrow \quad Z = W = \begin{bmatrix} \quad \end{bmatrix}$$

with $W, H \geq 0$

Non-negative matrix factorization

Data = n samples, each with d **non-negative** features

$$X \in \mathbb{R}^{n \times d} \quad X \geq 0$$

NMF

$$X \approx WH = \begin{bmatrix} \quad \\ \\ \\ \end{bmatrix} \begin{bmatrix} \qquad\qquad\qquad \\ \\ \end{bmatrix} \quad \rightarrow \quad Z = W = \begin{bmatrix} \quad \\ \\ \\ \end{bmatrix}$$

with $W, H \geq 0$

Note: Not a projection!

Interpretation

Data = n samples, each with d **non-negative** features

$$X \in \mathbb{R}^{n \times d} \quad X \geq 0$$

Let $W, H \geq 0$ such that

$$X \approx WH \quad \text{with} \quad H = \begin{bmatrix} h_1^T \\ \vdots \\ h_k^T \end{bmatrix}$$

Interpretation

Data $=$ n samples, each with d **non-negative** features

$$X \in \mathbb{R}^{n \times d} \quad X \geq 0$$

Let $W, H \geq 0$ such that

$$X \approx WH \quad \text{with} \quad H = \begin{bmatrix} h_1^T \\ \vdots \\ h_k^T \end{bmatrix}$$

Each data sample $x \in \mathbb{R}^d$ (row of X) can be seen as the weighted **superposition** of the components (or patterns) $h_1, \ldots, h_k \in \mathbb{R}^d$:

$$x \approx w_1 h_1 + \ldots + w_k h_k \quad w_1, \ldots, w_k \geq 0$$

A probabilistic view

After normalization, each data sample can be seen as a **probability distribution** over the features:

$$X \in \mathbb{R}^{n \times d} \quad X \geq 0 \quad \rightarrow \quad P \in \mathbb{R}^{n \times d} \quad P \geq 0, P1 = 1$$

Let $W, H \geq 0$ such that

$$P \approx WH \quad \text{with} \quad H = \begin{bmatrix} h_1^T \\ \vdots \\ h_k^T \end{bmatrix} \quad H1 = 1$$

A probabilistic view

After normalization, each data sample can be seen as a **probability distribution** over the features:

$$X \in \mathbb{R}^{n \times d} \quad X \geq 0 \quad \rightarrow \quad P \in \mathbb{R}^{n \times d} \quad P \geq 0, P1 = 1$$

Let $W, H \geq 0$ such that

$$P \approx WH \quad \text{with} \quad H = \begin{bmatrix} h_1^T \\ \vdots \\ h_k^T \end{bmatrix} \quad H1 = 1$$

Each data sample $p \in \mathbb{R}^d$ (row of P), seen as a probability distribution over the features, is a **mixture** of the probability distributions $h_1, \ldots, h_k \in \mathbb{R}^d$:

$$p \approx w_1 h_1 + \ldots + w_k h_k \quad w_1, \ldots, w_k \geq 0$$

Non-negative matrix factorization

Let $X \in \mathbb{R}^{n \times d}$ with $X \geq 0$

We seek to solve:

$$\boxed{\min_{W,H \geq 0} \|X - WH\|^2}$$

This optimization problem is **convex** in W or H but not in both

Non-negative matrix factorization

Let $X \in \mathbb{R}^{n \times d}$ with $X \geq 0$
We seek to solve:

$$\boxed{\min_{W,H \geq 0} \|X - WH\|^2}$$

This optimization problem is **convex** in W or H but not in both

Lee-Seung's algorithm (2000)

Alternate updates

$$H \leftarrow H \times \frac{W^T X}{W^T W H} \quad W \leftarrow W \times \frac{X H^T}{W H H^T}$$

with **component-wise** matrix multiplications and divisions

Theorem

The approximation error $\|X - WH\|^2$ is **non-increasing**

Inference

Train set $=$ n samples, each with d non-negative features

$$X_{\text{train}} \in \mathbb{R}^{n \times d} \quad X_{\text{train}} \geq 0$$

1. NMF $\rightarrow$ learning

$$X_{\text{train}} = \begin{bmatrix} \quad \\ \\ \\ \end{bmatrix} \begin{bmatrix} \qquad\qquad \end{bmatrix} \approx WH \quad \rightarrow \quad Z_{\text{train}} = W = \begin{bmatrix} \quad \\ \\ \\ \end{bmatrix}$$

Inference

Train set $=$ n samples, each with d non-negative features

$$X_{\text{train}} \in \mathbb{R}^{n \times d} \quad X_{\text{train}} \geq 0$$

1. NMF $\rightarrow$ learning

$$X_{\text{train}} = \begin{bmatrix} \quad \end{bmatrix} \begin{bmatrix} \qquad\qquad \end{bmatrix} \approx WH \quad \rightarrow \quad Z_{\text{train}} = W = \begin{bmatrix} \quad \end{bmatrix}$$

2. Constrained NMF $\rightarrow$ partial learning

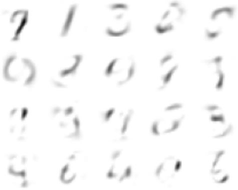
For the **test set**, apply Lee-Seung's algorithm with H fixed:

$$X_{\text{test}} = \begin{bmatrix} \quad \end{bmatrix} \begin{bmatrix} \qquad\qquad \end{bmatrix} \approx W'H \quad \rightarrow \quad Z_{\text{test}} = W' = \begin{bmatrix} \quad \end{bmatrix}$$

Example: MNIST

$X \in \{0, \ldots, 255\}^{n \times d}$
$n = 10,000$ samples
$d = 28 \times 28 = 784$

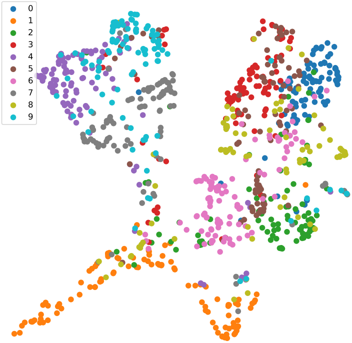
Samples



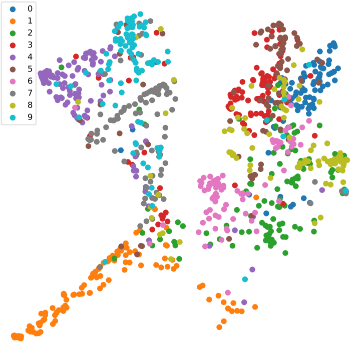
Components
$h_1, \ldots, h_{20}$

Example: MNIST

NMF in dimension 20
Visualization of 1,000 samples



Train set



Test set

Loss function

Let $X \in \mathbb{R}^{n \times d}$ with $X \geq 0$
We have seen the NMF for the **square error**:

$$\boxed{\min_{W,H \geq 0} \|X - WH\|^2}$$

What about other **loss functions**?

Bregman divergence

Let $F : \Omega \to \mathbb{R}$ be a **strictly convex** function of class C^1

Definition

The Bregman divergence associated with F is:

$$\forall x, y \in \Omega, \quad D_F(x, y) = F(x) - F(y) - \nabla F(y).(x - y)$$

Bregman divergence

Let $F : \Omega \to \mathbb{R}$ be a **strictly convex** function of class C^1

Definition

The Bregman divergence associated with F is:

$$\forall x, y \in \Omega, \quad D_F(x, y) = F(x) - F(y) - \nabla F(y).(x - y)$$

Proposition

We have $D_F(x, y) \geq 0$ and $D_F(x, y) = 0$ if and only if $x = y$

Note: In general, **not** a metric!

- Not symmetric
- No triangle inequality

Bregman divergence

Let $F : \Omega \to \mathbb{R}$ be a **strictly convex** function of class C^1

Definition

$$\forall x, y \in \Omega, \quad D_F(x, y) = F(x) - F(y) - \nabla F(y).(x - y)$$

Bregman divergence

Let $F : \Omega \to \mathbb{R}$ be a **strictly convex** function of class C^1

Definition

$$\forall x, y \in \Omega, \quad D_F(x,y) = F(x) - F(y) - \nabla F(y).(x-y)$$

Examples

- $\Omega = \mathbb{R}^d, F(x) = ||x||^2$

$$\boxed{D_F(x,y) = ||x-y||^2}$$

- $\Omega = \mathbb{R}_+^d, F(x) = \sum_{i=1}^d x_i \log x_i$

$$\boxed{D_F(x,y) = \sum_{i=1}^{d} \left(x_i \log \frac{x_i}{y_i} + x_i - y_i \right)}$$

$\to$ Generalized Kullback-Leibler divergence

NMF for the Kullback-Leibler divergence

Let $X \in \mathbb{R}^{n \times d}$ with $X \geq 0$
We seek to solve:

$$\boxed{\min_{W,H \geq 0} D(X||WH)}$$

This optimization problem is **convex** in W or H but not in both

NMF for the Kullback-Leibler divergence

Let $X \in \mathbb{R}^{n \times d}$ with $X \geq 0$

We seek to solve:

$$\boxed{\min_{W,H \geq 0} D(X||WH)}$$

This optimization problem is **convex** in W or H but not in both

Lee-Seung's algorithm (2000)

Alternate updates

$$H \leftarrow H \times \frac{W^T \frac{X}{WH}}{W^T 1 1^T} \quad W \leftarrow W \times \frac{\frac{X}{WH} H^T}{1 1^T H^T}$$

with **component-wise** matrix multiplications and divisions

Theorem

The divergence $D(X||WH)$ is **non-increasing**

Summary

Dimension reduction

- **Feature selection** $\rightarrow$ supervised learning
- **Random projection** $\rightarrow$ no learning
- **Matrix factorization** $\rightarrow$ unsupervised learning
  SVD $\leftrightarrow$ projection
  NMF $\leftrightarrow$ superposition