Machine Learning & Pattern Recognition

SONG Xuemeng

songxuemeng@sdu.edu.cn

http://xuemeng.bitcron.com/

Unsupervised Feature Extraction

- Principal Component Analysis (PCA)
- Nonnegative Matrix Factorization (NMF)

Unsupervised Feature Extraction

- Principal Component Analysis (PCA)
- Nonnegative Matrix Factorization (NMF)

What is feature extraction?

- Feature extraction (dimensionality reduction/feature reduction) refers to the mapping of the original **high-dimensional** data into a **low-dimensional** space.
- Criterion for feature reduction can be different based on different problem setting
  - ✓ Unsupervised setting: minimize the information loss
  - ✓ Supervised setting: maximize the class discrimination

Feature Extraction VS. Feature Selection

- Feature extraction
  - All original features are used.
  - The transformed features are linear combinations of the original features

- Feature selection
  - Only a subset of the original features are used.

Why Feature Extraction?

- Machine learning techniques may not be effective for high-dimensional data
  - Curse of Dimensionality
  - Accuracy and efficiency degrade rapidly as the dimension increases

- The **intrinsic** dimension may be small
  - For example, the number of genes responsible for a certain type of disease may be small

Why Feature Extraction?

- **Visualization**: projection of high-dimensional data onto 2D or 3D

- **Data compression**: efficient storage and retrieval

- **Noise removal**: positive effect on query accuracy

Feature Extraction Algorithms

- **Unsupervised**
  - Principal Component Analysis (PCA)
  - Nonnegative Matrix Factorization (NMF)
  - Independent Component Analysis (ICA) [Reading]
- **Supervised**
  - Linear Discriminant Analysis (LDA)
  - General Graph Embedding (GE) [Reading]
  - Canonical Correlation Analysis (CCA) [Reading, encouraged]
- **Semi-supervised**
  - Research topic [Further study, encouraged]

Principal Component Analysis (PCA)

- PCA represents the high-dimensional data in a more tractable, lower-dimensional form, without losing too much information.
- Reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables.
- Capture big (principal) variability in the data and ignore small variability.

- The new variables, called principal components (PCs), are uncorrelated, and are ordered by the fraction of the total information each retains.

Algebraic Derivation of PCs

Given a sample set of m observations on a vector of d variables

$$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m\} \in \mathbb{R}^d$$

Define the first PC of the samples by the linear projection $\boldsymbol{w}_1 \in \mathbb{R}^d$

$$z_{1i} = \boldsymbol{w}_1{}^{\boldsymbol{T}} \boldsymbol{x}_i = \sum_{k=1}^{d} w_{1k}\, x_{ik}$$

where $\boldsymbol{w}_1 = (w_{11}, w_{12}, \dots, w_{1d})^{\boldsymbol{T}}$ $\quad \boldsymbol{x}_i = (x_{11}, x_{12}, \dots, x_{1d}))^{\boldsymbol{T}}$

$$z_1 = \{z_{11}, z_{12}, \dots, z_{1m}\}$$

$\boldsymbol{w}_1$is chosen such that $var[z_1]$ is maximum.

Algebraic Derivation of PCs

To find $\boldsymbol{w}_1$, first note that

$$var[z_1] = E\left((z_1 - \bar{z}_1)^2\right) = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}_1{}^T\boldsymbol{x}_i - \boldsymbol{w}_1{}^T\overline{\boldsymbol{x}})^2$$

$$= \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{w}_1{}^T(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T\boldsymbol{w}_1 = \boldsymbol{w}_1{}^T\boldsymbol{S}\boldsymbol{w}_1$$

where $\boldsymbol{S} = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T$ is the covariance matrix.

$\overline{\boldsymbol{x}} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i$ is the mean.

Algebraic Derivation of PCs

To find $\boldsymbol{w}_1$, first note that

$$var[z_1] = E\left((z_1 - \bar{z}_1)^2\right) = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}_1{}^T\boldsymbol{x}_i - \boldsymbol{w}_1{}^T\overline{\boldsymbol{x}})^2$$

$$= \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{w}_1{}^T(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T\boldsymbol{w}_1 = \boldsymbol{w}_1{}^T\boldsymbol{S}\boldsymbol{w}_1$$

where $\boldsymbol{S} = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T$ is the covariance matrix.

$\overline{\boldsymbol{x}} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i$ is the mean.

The covariance matrix $\boldsymbol{S}$ is symmetric.

- The eigenvectors must be orthogonal to one another.
- The eigenvalues of $\boldsymbol{S}$ must all be ≥ 0

Algebraic Derivation of PCs

To find $\boldsymbol{w}_1$ that maximizes $var[z_1]$ subject to ${\boldsymbol{w}_1}^{\boldsymbol{T}}\boldsymbol{w}_1$=1

Let λ be a Lagrange multiplier

$$L = {\boldsymbol{w}_1}^{\boldsymbol{T}}\boldsymbol{S}\boldsymbol{w}_1 + \lambda({\boldsymbol{w}_1}^{\boldsymbol{T}}\boldsymbol{w}_1 - 1)$$

$$\Rightarrow \qquad \frac{\partial L}{\partial \boldsymbol{w}_1} = \boldsymbol{S}\boldsymbol{w}_1 - \lambda\boldsymbol{w}_1 = 0 \qquad \Rightarrow \qquad (\boldsymbol{S} - \lambda\boldsymbol{I}_d)\boldsymbol{w}_1 = 0$$

Therefore $\boldsymbol{w}_1$ is an eigenvector of $\boldsymbol{S}$.

The corresponding to the largest eigenvalue λ=λ_1

Algebraic Derivation of PCs

To find $\boldsymbol{w}_1$ that maximizes $var[z_1]$ subject to $\boldsymbol{w}_1{}^{\boldsymbol{T}}\boldsymbol{w}_1$=1

Let λ be a Lagrange multiplier

$$L = \boldsymbol{w}_1{}^{\boldsymbol{T}}\boldsymbol{S}\boldsymbol{w}_1 + \lambda(\boldsymbol{w}_1{}^{\boldsymbol{T}}\boldsymbol{w}_1 - 1)$$

$$\Rightarrow \qquad \frac{\partial L}{\partial \boldsymbol{w}_1} = \boldsymbol{S}\boldsymbol{w}_1 - \lambda \boldsymbol{w}_1 = 0 \qquad \Rightarrow \qquad (\boldsymbol{S} - \lambda \boldsymbol{I}_d)\boldsymbol{w}_1 = 0$$

Therefore $\boldsymbol{w}_1$ is an eigenvector of $\boldsymbol{S}$.

The corresponding to the largest eigenvalue λ=λ_1

WHY?

Algebraic Derivation of PCs

Similarly, $\boldsymbol{w}_2$ is also an eigenvector of $\boldsymbol{S}$, whose eigenvalue $\lambda = \lambda_2$ is the second largest.

In general $\quad var[z_k] = \boldsymbol{w}_k{}^T \boldsymbol{S} \boldsymbol{w}_k = \lambda_k$

The k-th largest eigenvalue of $\boldsymbol{S}$ is the variance of the k-th PC.
The k-th PC z_k retains the k-th greatest fraction of the variation in the sample.

Algebraic Derivation of PCs

- Main Steps for computing PCs:
  - Form the covariance matrix $\boldsymbol{S}$.
  - Compute its eigenvectors: $\{\boldsymbol{w}_i\}_{i=1}^{d}$
  - The first p eigenvectors $\{\boldsymbol{w}_i\}_{i=1}^{p}$ form the p PCs
  - The transformation G consists of the p PCs

$$\boldsymbol{G} \leftarrow \left[\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_p\right] \in \mathbb{R}^{d \times p}$$

$$\boldsymbol{y} = \boldsymbol{G}^{\boldsymbol{T}} \boldsymbol{x} \in \mathbb{R}^{p}$$

Algebraic Definition of PCs

- In practice, we compute the PCs via singular value decomposition (SVD) on the centered data matrix.

- Form the centered data matrix:

$$\boldsymbol{X} = [\,(\boldsymbol{x}_1 - \overline{\boldsymbol{x}}); \ldots; (\boldsymbol{x}_m - \overline{\boldsymbol{x}})] \in \mathbb{R}^{d\times m}$$

- Compute its SVD:

$$\boldsymbol{X} = \boldsymbol{U}_{d\times d}\boldsymbol{D}_{d\times m}(\boldsymbol{V}_{m\times m})^{\boldsymbol{T}}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices, $\boldsymbol{D}$ is a diagonal matrix.

Algebraic Definition of PCs

- Note that the scatter/covariance matrix can be written as

$$\boldsymbol{S} = \boldsymbol{X}\boldsymbol{X}^T = \boldsymbol{U}\boldsymbol{D}^2\boldsymbol{U}^T \qquad \boldsymbol{X} = \boldsymbol{U}_{d\times d}\boldsymbol{D}_{d\times m}(\boldsymbol{V}_{m\times m})^T$$

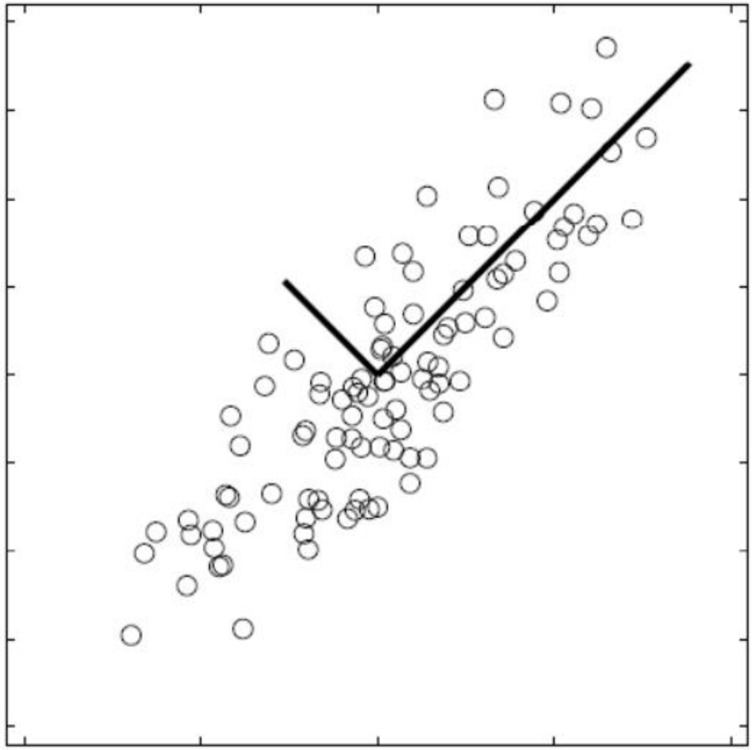
- The eigenvectors of $\boldsymbol{S}$ are the columns of $\boldsymbol{U}$ and the eigenvalues are the diagonal elements of $\boldsymbol{D}^2$.

- Take only a few significant eigenvalue-eigenvector pairs $p \ll d$. The new reconstructed sample from low-dim space is:

$$\hat{\boldsymbol{x}}_i = \overline{\boldsymbol{x}} + \boldsymbol{U}_{d\times p}\left(\boldsymbol{U}_{d\times p}\right)^T(\boldsymbol{x}_i - \overline{\boldsymbol{x}})$$

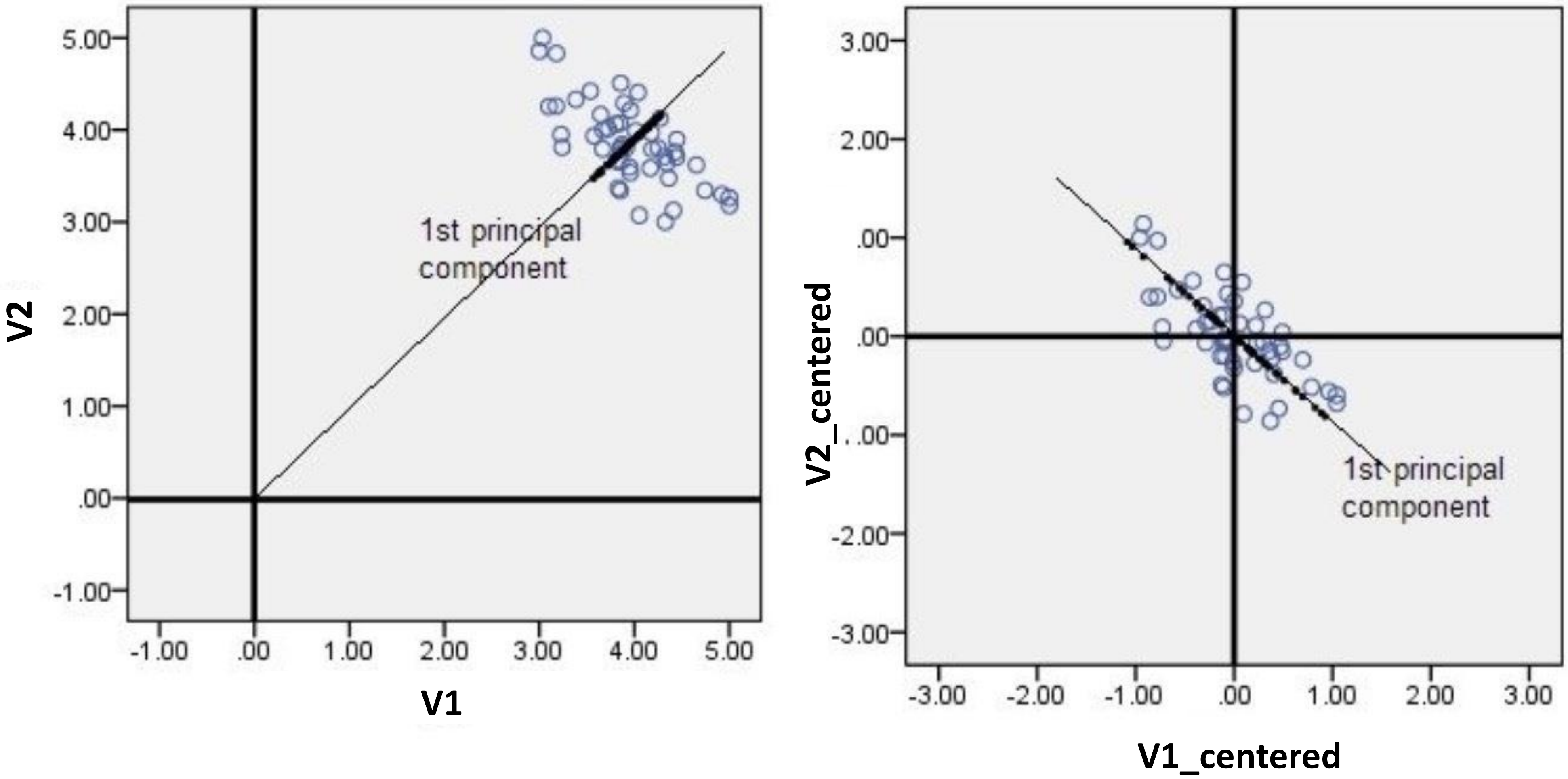
WHY?

Visualize PCs



Data points are represented in a rotated orthogonal coordinate system: the origin is the mean of the data points and the axes are provided by the eigenvectors.

The Necessity of Centralization



How Many PCs to Keep?

How Many PCs to Keep?

To choose p based on percentage of energy to retain, we can use the following criterion (smallest p):

$$\frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq Threshold \quad (e.g., 0.95)$$

PCA and Classification

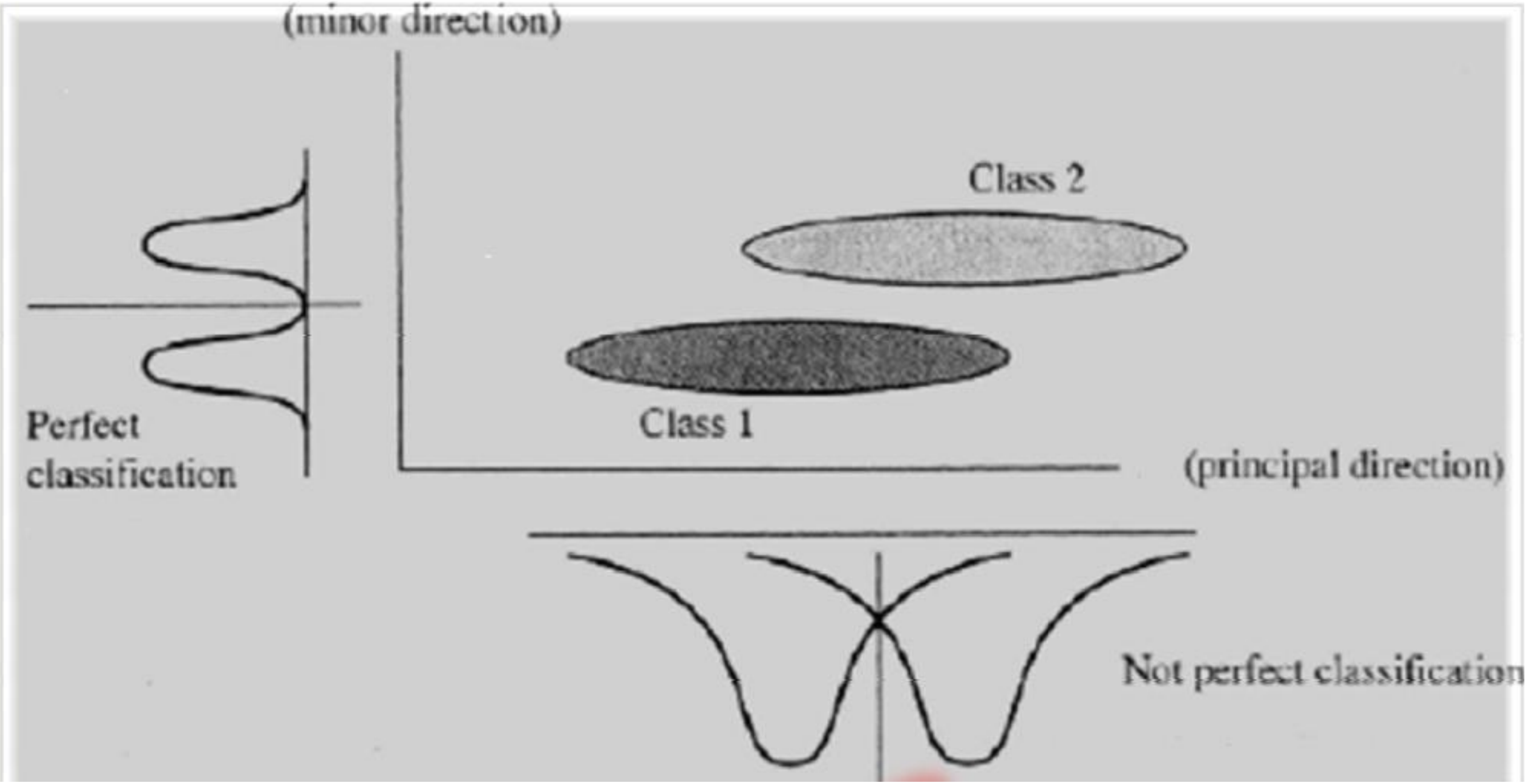
- Classification with PCA
  - Project both training and testing data into the PCs space
  - For each testing sample, use NN for classification
  - Issue: accuracy is sensitive to the number of PCs

PCA and Classification

- Classification with PCA
  - Project both training and testing data into the PCs space
  - For each testing sample, use NN for classification
  - Issue: accuracy is sensitive to the number of PCs

- PCA may not be always an optimal feature extraction technique for classification.
  - Suppose there are C classes in the training data
  - PCA is based on the sample covariance which characterizes the scatter of the entire data set, irrespective of class-membership.
  - The projection axes chosen by PCA might not provide good discrimination power.

PCA and Classification



Summary of PCA

Algorithm 1 Algorithm for PCA

Input: Samples $\{x_1, x_2, \cdots, x_N\}$.

1. Compute the covariance matrix:
$$S = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T;$$
2. Perform Eigenvalue Decomposition: $[U] = \mathrm{eig}(S)$;
3. Output PCs matrix $U(:, 1:p)$.

Unsupervised Feature Extraction

- Principal Component Analysis (PCA)
- Nonnegative Matrix Factorization (NMF)

Non-negative Matrix Factorization

- PCA review
  - Find a set of orthogonal principal components (basis)
  - The reconstructed image is a linear combination of PCs plus mean
  - PCA involves adding up some basis vectors and subtracting others.
  - Basis vectors are not physically intuitive for many applications.
    - Sometimes subtracting does not make sense.
    - How to subtract a face? Negative pixel?

- NMF: Like PCA, except that the coefficients in the linear combination cannot be negative.

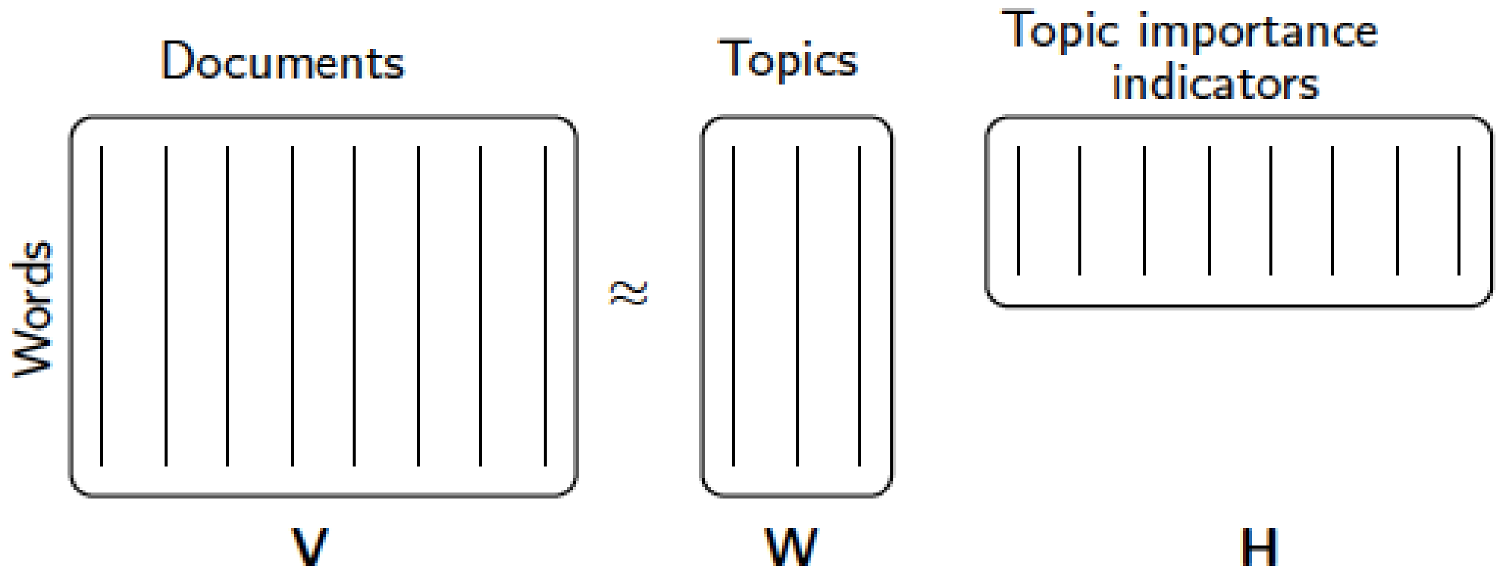
NMF Basis Vectors

- Only allowing adding of basis vectors makes intuitive sense
- Forcing the reconstruction coefficients to be nonnegative leads to nice basis vectors
  - To reconstruct vector (image), all you can do is to add in more basis vectors
  - This leads to basis vectors that represent parts

Objective Function

Assume $\boldsymbol{V} \in \mathbb{R}^{m \times n}$ is the sample matrix, the task is to approximate the original data matrix with two nonnegative data matrices $\boldsymbol{V} \approx \boldsymbol{W}\boldsymbol{H}$, $\boldsymbol{W} \in \mathbb{R}^{m \times k}$, $\boldsymbol{H} \in \mathbb{R}^{k \times n}$:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H}\|_F^2, \text{ s.t. } \boldsymbol{W} \geq \boldsymbol{0} \text{ and } \boldsymbol{H} \geq \boldsymbol{0}$$

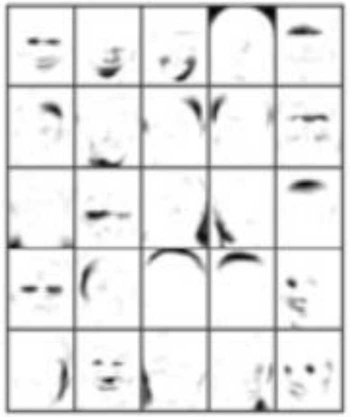


Objective Function

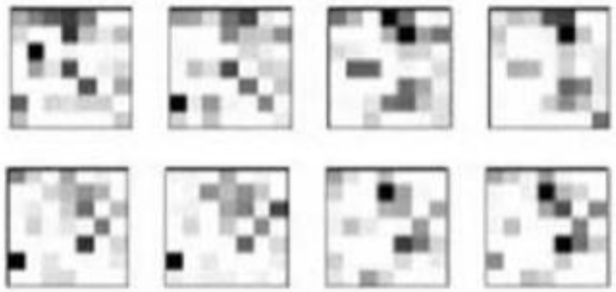
Assume $\boldsymbol{V} \in \mathbb{R}^{m \times n}$ is the sample matrix, the task is to approximate the original data matrix with two nonnegative data matrices $\boldsymbol{V} \approx \boldsymbol{W}\boldsymbol{H}$, $\boldsymbol{W} \in \mathbb{R}^{m \times k}$, $\boldsymbol{H} \in \mathbb{R}^{k \times n}$:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H}\|_F^2, \text{ s.t. } \boldsymbol{W} \geq \boldsymbol{0} \text{ and } \boldsymbol{H} \geq \boldsymbol{0}$$

$\boldsymbol{V}$: an image is a column vector. Vector shown as a matrix here.



$\boldsymbol{W}$: a basis vector is a column vector



$\boldsymbol{H}$: a coefficient vector is a column vector

Optimization

- Use gradient descent to find a local minimum

$$
\begin{aligned}
J = \|\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H}\|_F^2 &= tr((\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H})(\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H})^{\boldsymbol{T}}) \\
&= tr(\boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}} - \boldsymbol{W}\boldsymbol{H}\boldsymbol{V}^{\boldsymbol{T}} - \boldsymbol{V}\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}} + \boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}}) \\
&= tr\left(\boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}}\right) - 2tr\left(\boldsymbol{W}\boldsymbol{H}\boldsymbol{V}^{\boldsymbol{T}}\right) + tr(\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}})
\end{aligned}
$$

- The gradient descent update rule is ($\boldsymbol{H}$ only, $\boldsymbol{W}$ similar):

$$
\frac{\partial J}{\partial \boldsymbol{H}} = ?
$$

Matrix calculus

- Numerator layout: lay out according to $\boldsymbol{y}$ and $\boldsymbol{x}^{\boldsymbol{T}}$. (Jacobian formulation)
- Denominator layout: lay out according to $\boldsymbol{y}^{\boldsymbol{T}}$ and $\boldsymbol{x}$. (Hessian formulation)

Numerator layout:

分子布局

$$\frac{\partial y}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} & \cdots & \frac{\partial y}{\partial x_n} \end{bmatrix}$$

$$\frac{\partial \boldsymbol{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} \\ \frac{\partial y_2}{\partial x} \\ \vdots \\ \frac{\partial y_n}{\partial x} \end{bmatrix}$$

Denominator layout:

分母布局

$$\frac{\partial y}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix}$$

$$\frac{\partial \boldsymbol{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} & \frac{\partial y_2}{\partial x} & \cdots & \frac{\partial y_n}{\partial x} \end{bmatrix}$$

Optimization

- Use gradient descent to find a local minimum

$$
\begin{aligned}
J = \|\boldsymbol{V} - \boldsymbol{WH}\|_F^2 &= tr((\boldsymbol{V} - \boldsymbol{WH})(\boldsymbol{V} - \boldsymbol{WH})^{\boldsymbol{T}}) \\
&= tr(\boldsymbol{VV}^{\boldsymbol{T}} - \boldsymbol{WHV}^{\boldsymbol{T}} - \boldsymbol{VH}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}} + \boldsymbol{WHH}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}}) \\
&= tr\left(\boldsymbol{VV}^{\boldsymbol{T}}\right) - 2tr\left(\boldsymbol{WHV}^{\boldsymbol{T}}\right) + tr(\boldsymbol{WHH}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}})
\end{aligned}
$$

Condition	Expression	Numerator layout	Denominator layout
A is not a function of **X**	[5] $\frac{\partial \operatorname{tr}(\mathbf{X}^\top \mathbf{A}\mathbf{X})}{\partial \mathbf{X}} =$	$\mathbf{X}^\top(\mathbf{A} + \mathbf{A}^\top)$	$(\mathbf{A} + \mathbf{A}^\top)\mathbf{X}$
A is not a function of **X**	[5] $\frac{\partial \operatorname{tr}(\mathbf{X}^{-1} \mathbf{A})}{\partial \mathbf{X}} =$	$-\mathbf{X}^{-1}\mathbf{A}\mathbf{X}^{-1}$	$-(\mathbf{X}^{-1})^\top\mathbf{A}^\top(\mathbf{X}^{-1})^\top$
A, **B** are not functions of **X**	$\frac{\partial \operatorname{tr}(\mathbf{A}\mathbf{X}\mathbf{B})}{\partial \mathbf{X}} = \frac{\partial \operatorname{tr}(\mathbf{B}\mathbf{A}\mathbf{X})}{\partial \mathbf{X}} =$	$\mathbf{B}\mathbf{A}$	$\mathbf{A}^\top\mathbf{B}^\top$
A, **B**, **C** are not functions of **X**	$\frac{\partial \operatorname{tr}(\mathbf{A}\mathbf{X}\mathbf{B}\mathbf{X}^\top\mathbf{C})}{\partial \mathbf{X}} =$	$\mathbf{B}\mathbf{X}^\top\mathbf{C}\mathbf{A} + \mathbf{B}^\top\mathbf{X}^\top\mathbf{A}^\top\mathbf{C}^\top$	$\mathbf{A}^\top\mathbf{C}^\top\mathbf{X}\mathbf{B}^\top + \mathbf{C}\mathbf{A}\mathbf{X}\mathbf{B}$

https://en.wikipedia.org/wiki/Matrix_calculus

Optimization

- Use gradient descent to find a local minimum

$$
\begin{aligned}
J = \|\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H}\|_F^2 &= tr((\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H})(\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H})^{\boldsymbol{T}}) \\
&= tr(\boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}} - \boldsymbol{W}\boldsymbol{H}\boldsymbol{V}^{\boldsymbol{T}} - \boldsymbol{V}\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}} + \boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}}) \\
&= tr\left(\boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}}\right) - 2tr\left(\boldsymbol{W}\boldsymbol{H}\boldsymbol{V}^{\boldsymbol{T}}\right) + tr(\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}})
\end{aligned}
$$

- The gradient descent update rule is ($\boldsymbol{H}$ only, $\boldsymbol{W}$ similar):

$$H_{a\mu} \leftarrow H_{a\mu} + \eta_{a\mu}[(W^T V)_{a\mu} - (W^T W H)_{a\mu}]$$

WHY?

Deriving Update Rules

- Gradient Descent Rule:

$$H_{a\mu} \leftarrow H_{a\mu} + \eta_{a\mu}[(W^T V)_{a\mu} - (W^T W H)_{a\mu}]$$

Justify later!

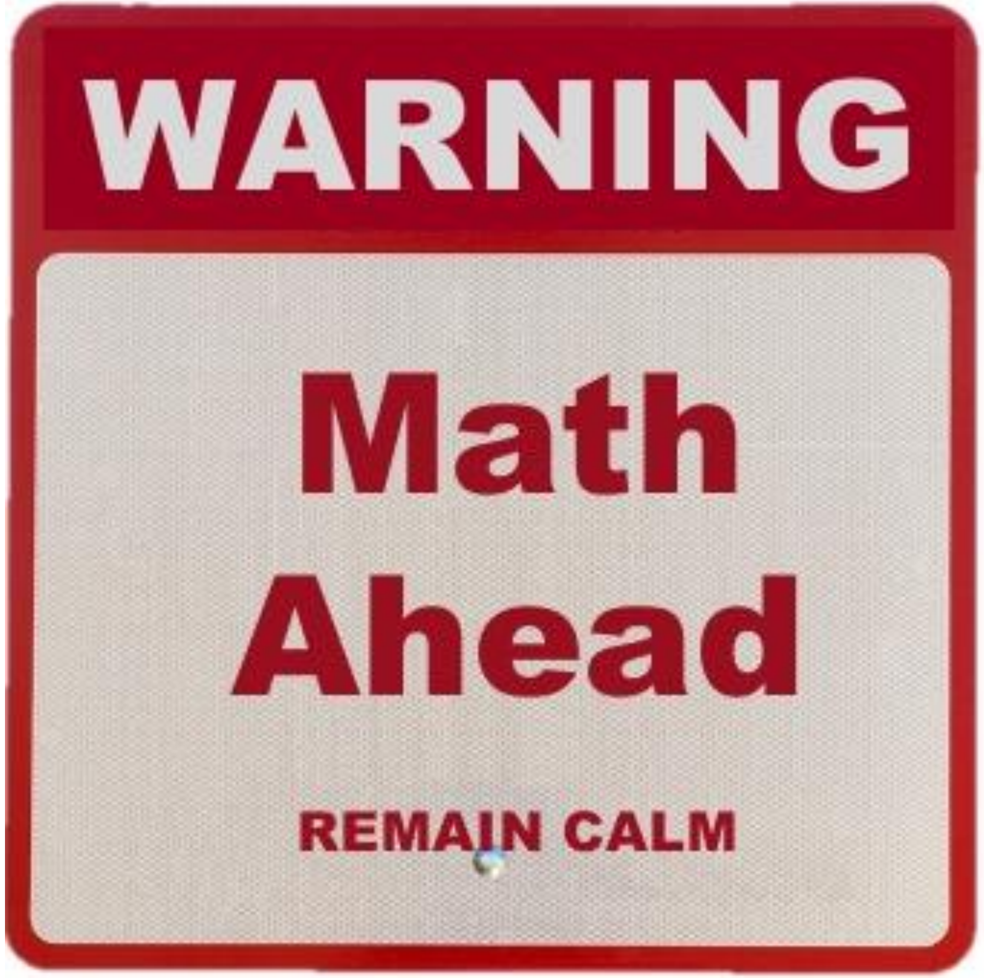
- Set $\eta_{a\mu} = \dfrac{H_{a\mu}}{(W^T W H)_{a\mu}}$, the update rule becomes

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}}$$

What is Significant about This?

- This is a multiplicative update instead of an additive update.
  - If the initial values of $\boldsymbol{W}$ and $\boldsymbol{H}$ are all non-negative, then the W and H can never become negative.
- This lets us produce a non-negative factorization

How do we know that this will converge?


WARNING
Math
Ahead
REMAIN CALM

How do we know that this will converge?

Definition 1: $G(h, h')$ is an auxiliary function for $F(h)$ if the conditions

$$G(h, h') \geq F(h) \quad G(h, h) = F(h)$$

are satisfied.

How do we know that this will converge?

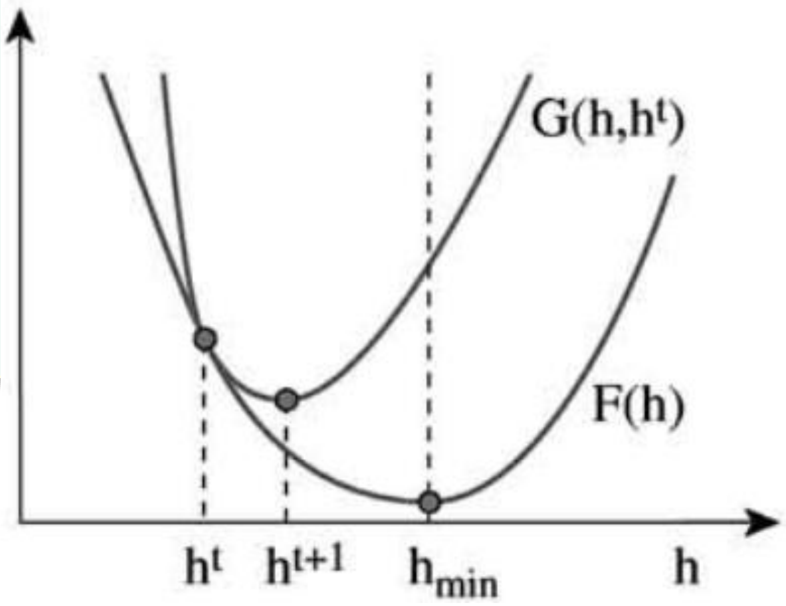
Definition 1: $G(h, h')$ is an auxiliary function for $F(h)$ if the conditions

$$G(h, h') \geq F(h) \quad G(h, h) = F(h)$$

are satisfied.

Lemma 1: If G is an auxiliary function of F, then F is nonincreasing under the update

$$h^{t+1} = \arg\min_h G(h, h^t)$$



How do we know that this will converge?

Definition 1: $G(h, h')$ is an auxiliary function for $F(h)$ if the conditions

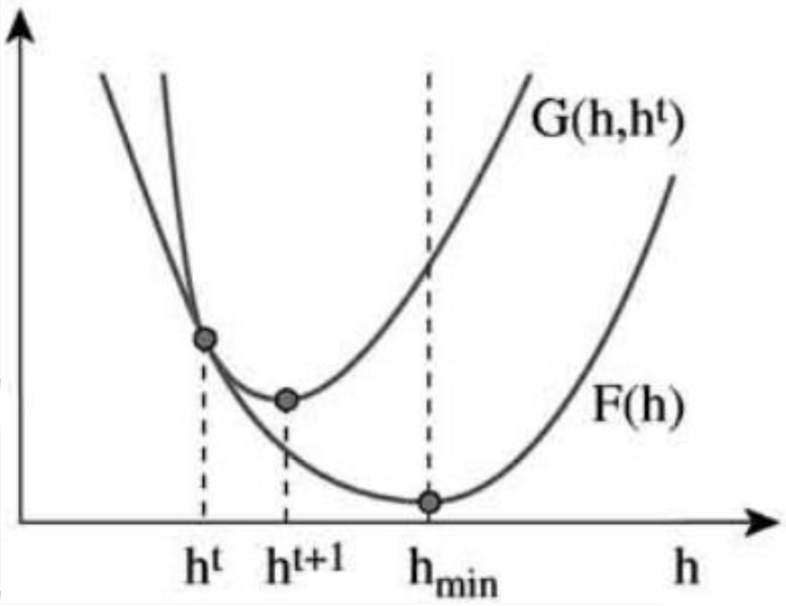
$$G(h, h') \geq F(h) \quad G(h, h) = F(h)$$

are satisfied.

Lemma 1: If G is an auxiliary function of F, then F is nonincreasing under the update

$$h^{t+1} = \arg\min_{h} G(h, h^t)$$

Proof: $F(h^{t+1}) \leq G(h^{t+1}, h^t) \leq G(h^t, h^t) = F(h^t)$



How do we know that this will converge?

Lemma 2: If $\boldsymbol{K}(\boldsymbol{h}^t)$ is the diagonal matrix

$$K_{ab}(h^t) = \delta_{ab}(W^T W h^t)_a / h_a^t$$

then $G(\boldsymbol{h}, \boldsymbol{h}^t) = F(\boldsymbol{h}^t) + (\boldsymbol{h} - \boldsymbol{h}^t)^{\boldsymbol{T}} \nabla F(\boldsymbol{h}^t) + \frac{1}{2}(\boldsymbol{h} - \boldsymbol{h}^t)^{\boldsymbol{T}} \boldsymbol{K}(\boldsymbol{h}^t)(\boldsymbol{h} - \boldsymbol{h}^t)$

is an auxiliary function for $F(\boldsymbol{h}) = \frac{1}{2}\sum_i (v_i - \sum_a W_{ia} h_a)^2$

- **Kronecker delta:** $\delta_{ab} = I_{ab}$, $\boldsymbol{I}$ is the identity matrix.

How do we know that this will converge?

Lemma 2: If $\boldsymbol{K}(\boldsymbol{h}^t)$ is the diagonal matrix

$$K_{ab}(h^t) = \delta_{ab}(W^T W h^t)_a / h_a^t$$

then $G(\boldsymbol{h}, \boldsymbol{h}^t) = F(\boldsymbol{h}^t) + (\boldsymbol{h} - \boldsymbol{h}^t)^{\boldsymbol{T}} \nabla F(\boldsymbol{h}^t) + \frac{1}{2}(\boldsymbol{h} - \boldsymbol{h}^t)^{\boldsymbol{T}} \boldsymbol{K}(\boldsymbol{h}^t)(\boldsymbol{h} - \boldsymbol{h}^t)$

is an auxiliary function for $F(\boldsymbol{h}) = \frac{1}{2}\sum_i (v_i - \sum_a W_{ia} h_a)^2$

- **Kronecker delta:** $\delta_{ab} = I_{ab}$, $\boldsymbol{I}$ is the identity matrix.

$$F(\boldsymbol{H}) = \frac{1}{2}\|\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H}\|_F^2 = \frac{1}{2}\sum_{j=1}^n \left\|\boldsymbol{V}_{\cdot j} - \boldsymbol{W}\boldsymbol{H}_{\cdot j}\right\|_2^2 = \frac{1}{2}\sum_{j=1}^n \sum_{i=1}^d (\boldsymbol{V}_{ij} - \boldsymbol{W}_{i\cdot}\boldsymbol{H}_{\cdot j})^2$$

$$= \frac{1}{2}\sum_{j=1}^n \sum_{i=1}^d (\boldsymbol{V}_{ij} - \sum_{a=1}^c \boldsymbol{W}_{ia}\boldsymbol{H}_{aj})^2$$

How do we know that this will converge?

Lemma 2: If $\boldsymbol{K}(\boldsymbol{h}^t)$ is the diagonal matrix

$$K_{ab}(h^t) = \delta_{ab}(W^T W h^t)_a / h_a^t$$

then $G(\boldsymbol{h}, \boldsymbol{h}^t) = F(\boldsymbol{h}^t) + (\boldsymbol{h} - \boldsymbol{h}^t)^{\boldsymbol{T}} \nabla F(\boldsymbol{h}^t) + \frac{1}{2}(\boldsymbol{h} - \boldsymbol{h}^t)^{\boldsymbol{T}} \boldsymbol{K}(\boldsymbol{h}^t)(\boldsymbol{h} - \boldsymbol{h}^t)$

is an auxiliary function for $F(\boldsymbol{h}) = \frac{1}{2}\sum_i (v_i - \sum_a W_{ia} h_a)^2$

❖ **Kronecker delta:** $\delta_{ab} = I_{ab}$, $\boldsymbol{I}$ is the identity matrix.

$$F(\boldsymbol{H}) = \frac{1}{2}\|\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H}\|_F^2 = \frac{1}{2}\sum_{j=1}^n \left\|\boldsymbol{V}_{\cdot j} - \boldsymbol{W}\boldsymbol{H}_{\cdot j}\right\|_2^2 = \frac{1}{2}\sum_{j=1}^n \sum_{i=1}^d (\boldsymbol{V}_{ij} - \boldsymbol{W}_{i\cdot}\boldsymbol{H}_{\cdot j})^2$$

$$= \frac{1}{2}\sum_{j=1}^n \sum_{i=1}^d (\boldsymbol{V}_{ij} - \sum_{a=1}^c \boldsymbol{W}_{ia}\boldsymbol{H}_{aj})^2$$

$$F(\boldsymbol{H}_{\cdot j}) = \frac{1}{2}\sum_{i=1}^d (\boldsymbol{V}_{ij} - \sum_{a=1}^c \boldsymbol{W}_{ia}\boldsymbol{H}_{aj})^2$$

How do we know that this will converge?

Lemma 2: If $\boldsymbol{K}(\boldsymbol{h}^t)$ is the diagonal matrix

$$K_{ab}(h^t) = \delta_{ab}(W^T W h^t)_a / h_a^t$$

then $G(\boldsymbol{h}, \boldsymbol{h}^t) = F(\boldsymbol{h}^t) + (\boldsymbol{h} - \boldsymbol{h}^t)^{\boldsymbol{T}} \nabla F(\boldsymbol{h}^t) + \frac{1}{2}(\boldsymbol{h} - \boldsymbol{h}^t)^{\boldsymbol{T}} \boldsymbol{K}(\boldsymbol{h}^t)(\boldsymbol{h} - \boldsymbol{h}^t)$

is an auxiliary function for $F(\boldsymbol{h}) = \frac{1}{2}\sum_i (v_i - \sum_a W_{ia} h_a)^2$

❖ **Kronecker delta:** $\delta_{ab} = I_{ab}$, $\boldsymbol{I}$ is the identity matrix.

$$F(\boldsymbol{H}) = \frac{1}{2}\|\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H}\|_F^2 = \frac{1}{2}\sum_{j=1}^{n}\left\|\boldsymbol{V}_{\cdot j} - \boldsymbol{W}\boldsymbol{H}_{\cdot j}\right\|_2^2 = \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{d}(\boldsymbol{V}_{ij} - \boldsymbol{W}_{i\cdot}\boldsymbol{H}_{\cdot j})^2$$

$$= \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{d}(\boldsymbol{V}_{ij} - \sum_{a=1}^{c}\boldsymbol{W}_{ia}\boldsymbol{H}_{aj})^2$$

$$F(\boldsymbol{H}_{\cdot j}) = \frac{1}{2}\sum_{i=1}^{d}(\boldsymbol{V}_{ij} - \sum_{a=1}^{c}\boldsymbol{W}_{ia}\boldsymbol{H}_{aj})^2 \xrightarrow[\boldsymbol{V}_{\cdot j} \text{ as } \boldsymbol{v}]{\boldsymbol{H}_{\cdot j} \text{ as } \boldsymbol{h}} F(\boldsymbol{h}) = \frac{1}{2}\sum_i (v_i - \sum_a W_{ia} h_a)^2$$

Proof of Lemma 2

Since $G(\boldsymbol{h},\boldsymbol{h}) = F(\boldsymbol{h})$ is obvious, we need to show that $G(\boldsymbol{h},\boldsymbol{h}^t) \geq F(\boldsymbol{h})$.
To do this, we compare the 2nd order Taylor polynomial

$$F(\boldsymbol{h}) = F(\boldsymbol{h}^t) + (\boldsymbol{h}-\boldsymbol{h}^t)^{\boldsymbol{T}}\nabla F(\boldsymbol{h}^t) + \frac{1}{2}(\boldsymbol{h}-\boldsymbol{h}^t)^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{W}(\boldsymbol{h}-\boldsymbol{h}^t)$$

$$\nabla F(\boldsymbol{h}) = \frac{\partial F(\boldsymbol{h})}{\partial \boldsymbol{h}} = -\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{v} + \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{h} \qquad \frac{\partial F(\boldsymbol{h})}{\partial \boldsymbol{h}\partial \boldsymbol{h}} = \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{W}$$

Since we know that

$$G(\boldsymbol{h},\boldsymbol{h}^t) = F(\boldsymbol{h}^t) + (\boldsymbol{h}-\boldsymbol{h}^t)^{\boldsymbol{T}}\nabla F(\boldsymbol{h}^t) + \frac{1}{2}(\boldsymbol{h}-\boldsymbol{h}^t)^{\boldsymbol{T}}\boldsymbol{K}(\boldsymbol{h}^t)(\boldsymbol{h}-\boldsymbol{h}^t)$$

We find that prove

$$G(\boldsymbol{h},\boldsymbol{h}^{\mathrm{t}}) \geq F(\boldsymbol{h}) \qquad \Leftrightarrow \qquad 0 \leq (\boldsymbol{h}-\boldsymbol{h}^t)^{\boldsymbol{T}}[\boldsymbol{K}(\boldsymbol{h}^t) - \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{W}](\boldsymbol{h}-\boldsymbol{h}^t)$$

Proof of Lemma 2

$0 \le (\boldsymbol{h}-\boldsymbol{h}^t)^T[\boldsymbol{K}(\boldsymbol{h}^t)-\boldsymbol{W}^T\boldsymbol{W}](\boldsymbol{h}-\boldsymbol{h}^t) \quad \Leftrightarrow \quad \boldsymbol{K}(\boldsymbol{h}^t)-\boldsymbol{W}^T\boldsymbol{W}$ is psd.

We consider matrix $\boldsymbol{M} = \boldsymbol{H}^t[\boldsymbol{K}(\boldsymbol{h}^t)-\boldsymbol{W}^T\boldsymbol{W}]\boldsymbol{H}^t \quad \boldsymbol{H}^t = \boldsymbol{H}^{t^T} = \begin{bmatrix} \boldsymbol{h}_1^t & & \\ & \ddots & \\ & & \boldsymbol{h}_n^t \end{bmatrix}$

$$M(\boldsymbol{h}^t)_{ab} = \boldsymbol{h}_a^t(\boldsymbol{K}(\boldsymbol{h}^t)-\boldsymbol{W}^T\boldsymbol{W})_{ab}\boldsymbol{h}_b^t$$

Then $\boldsymbol{K}(\boldsymbol{h}^t)-\boldsymbol{W}^T\boldsymbol{W}$ is psd if and only if $\boldsymbol{M}$ is. Why?

If $\boldsymbol{M}$ is positive semidefinite, then for any vector $\boldsymbol{v}$, we have

$$\boldsymbol{v}^T\boldsymbol{M}\boldsymbol{v} = \boldsymbol{v}^T\boldsymbol{H}^t[\boldsymbol{K}(\boldsymbol{h}^t)-\boldsymbol{W}^T\boldsymbol{W}]\boldsymbol{H}^t\boldsymbol{v} = (\boldsymbol{H}^t\boldsymbol{v})^T[\boldsymbol{K}(\boldsymbol{h}^t)-\boldsymbol{W}^T\boldsymbol{W}](\boldsymbol{H}^t\boldsymbol{v}) \ge 0$$

if and only if $\boldsymbol{K}(\boldsymbol{h}^t)-\boldsymbol{W}^T\boldsymbol{W}$ is also positive semidefinite.

Proof of Lemma 2

$$\boldsymbol{v}^T\boldsymbol{M}\boldsymbol{v} = \sum_{ab} v_a M_{ab} v_b = \sum_{ab} v_a h_a^t K_{ab}(h^t) h_b^t v_b - \sum_{ab} v_a h_a^t (W^T W)_{ab} h_b^t v_b$$

$$= \sum_{a} v_a h_a^t \frac{(W^T W h^t)_a}{h_a^t} h_a^t v_a - \sum_{ab} v_a h_a^t (W^T W)_{ab} h_b^t v_b$$

$$= \sum_{ab} h_a^t (W^T W)_{ab} h_b^t v_a^2 - \sum_{ab} v_a h_a^t (W^T W)_{ab} h_b^t v_b$$

symmetric

$$= \sum_{ab} (W^T W)_{ab} h_a^t h_b^t [\frac{1}{2} v_a^2 + \frac{1}{2} v_b^2 - v_a v_b] = \frac{1}{2} \sum_{ab} (W^T W)_{ab} h_a^t h_b^t (v_a - v_b)^2$$

$$\geq 0$$

$$\boldsymbol{M} = \boldsymbol{H}^t[\boldsymbol{K}(\boldsymbol{h}^t) - \boldsymbol{W}^T\boldsymbol{W}]\boldsymbol{H}^t \qquad M(\boldsymbol{h}^t)_{ab} = \boldsymbol{h}_a^t(\boldsymbol{K}(\boldsymbol{h}^t) - \boldsymbol{W}^T\boldsymbol{W})_{ab}\boldsymbol{h}_b^t$$

$$(W^T W h^t)_a = (W^T W)_a h^t = \sum_b (W^T W)_{ab} h_b^t$$

How do we know that this will converge?

- Given the auxiliary function

$$K_{ab}(h^t) = \delta_{ab}(W^T W h^t)_a / h_a^t$$

$$G(\boldsymbol{h}, \boldsymbol{h}^t) = F(\boldsymbol{h}^t) + (\boldsymbol{h} - \boldsymbol{h}^t)^{\boldsymbol{T}} \nabla F(\boldsymbol{h}^t) + \frac{1}{2}(\boldsymbol{h} - \boldsymbol{h}^t)^{\boldsymbol{T}} \boldsymbol{K}(\boldsymbol{h}^t)(\boldsymbol{h} - \boldsymbol{h}^t)$$

- According to lemma 1, F is nonincreasing under the update

$$\boldsymbol{h}^{t+1} = \arg\min_{\boldsymbol{h}} G(\boldsymbol{h}, \boldsymbol{h}^t) \quad \Rightarrow \quad \boldsymbol{h}^{t+1} = \boldsymbol{h}^t - \underbrace{\boxed{\boldsymbol{K}(\boldsymbol{h}^t)^{-1}}}_{\eta} \nabla F(\boldsymbol{h}^t)$$

- The rule can be explicitly written as

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}}$$

Deriving Update Rules

- Gradient Descent Rule:

$$H_{a\mu} \leftarrow H_{a\mu} + \eta_{a\mu}[(W^T V)_{a\mu} - (W^T W H)_{a\mu}]$$

Justify later!

- Set $\eta_{a\mu} = \frac{H_{a\mu}}{(W^T W H)_{a\mu}}$, the update rule becomes

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}}$$

Deriving Update Rules

Note: The whole procedure iterates between the optimizations of $\boldsymbol{H}$ and $\boldsymbol{W}$ until converged, given that $\boldsymbol{W}$ and $\boldsymbol{H}$ are initialized as nonnegative values.

- The update rule:

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}}$$

$$W_{ia} \leftarrow W_{ia} \frac{(V H^T)_{ia}}{(W H H^T)_{ia}}$$

Deriving Update Rules

Note: The whole procedure iterates between the optimizations of $\boldsymbol{H}$ and $\boldsymbol{W}$ until converged, given that $\boldsymbol{W}$ and $\boldsymbol{H}$ are initialized as nonnegative values.

- The update rule:

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}}$$

$$\boxed{W_{ia} \leftarrow W_{ia} \frac{(V H^T)_{ia}}{(W H H^T)_{ia}}}$$

Your homework☺

Summary of NMF

Algorithm 2 Algorithm for NMF

Input: Sample matrix $V = [v_1, v_2, \cdots, v_N]$.
Initialize W^0 and H^0 as arbitrary positive matrices.

for $t = 0 : 1 : T_{max}$ **do**

$H_{a\mu}^{t+1} = H_{a\mu}^{t} \frac{({W^t}^T V)_{a\mu}}{({W^t}^T W^t H^t)_{a\mu}}$;

$W_{a\mu}^{t+1} = W_{a\mu}^{t} \frac{(V {H^{t+1}}^T)_{a\mu}}{(W^t H^{t+1} {H^{t+1}}^T)_{a\mu}}$;

If $\|W^t - W^{t+1}\| < \epsilon$ and $\|H^t - H^{t+1}\| < \epsilon$

return;

end for

3. Output matrices W and H.

Discussion

- For new image, how to obtain the reconstruction coefficients?