EPAT 2014
Ecole de Printemps sur l'Apprentissage arTificiel
La Calanque, Carry-le-Rouet
du 7 au 12 Juin 2014

Apprentissage par factorisation matricielle

Younès BENNANI et Ievgen REDKO
LIPN, Université Paris 13 – Sorbonne Paris Cité

EPAT'14 – Carry-Le-Rouet 7-12 juin 2014

Matrix factorizations

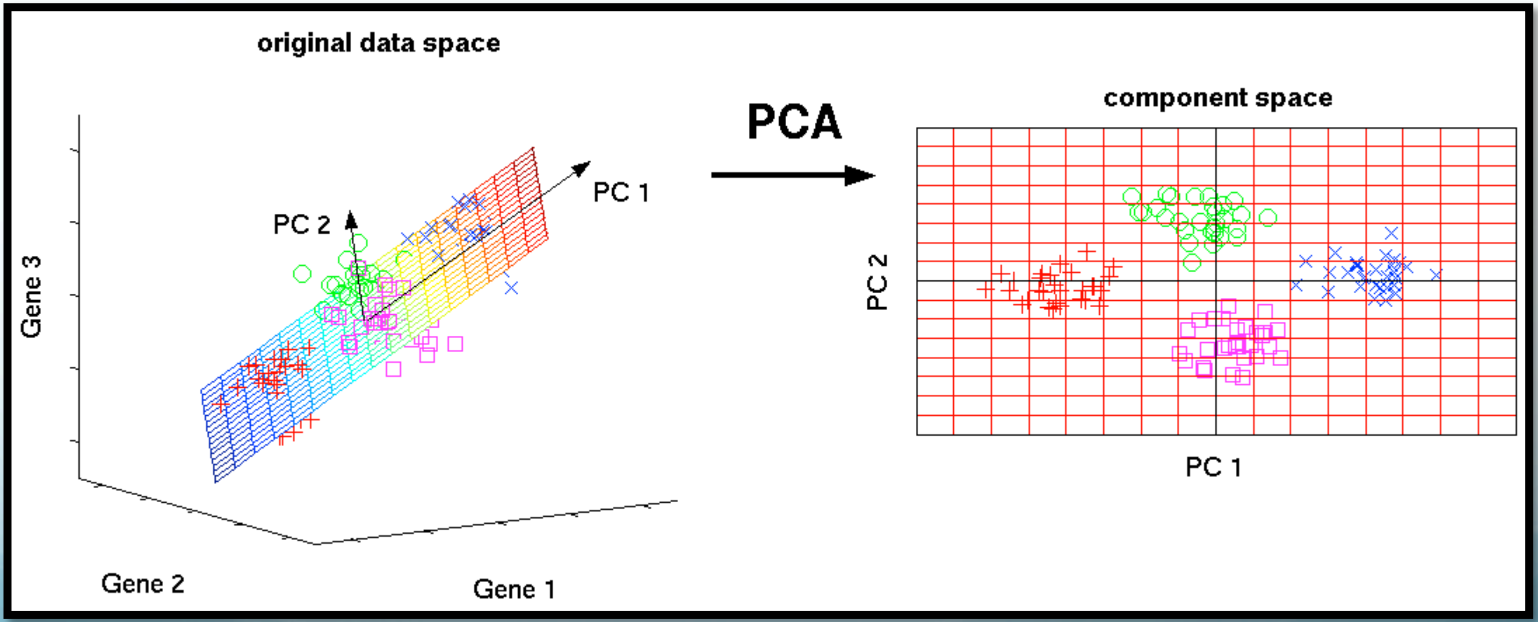
- What can a matrix represent?
  - System of equations
  - User rating matrix
  - Image
  - Matrix structure in graph theory
    - Adjacent matrix
    - Distance matrix

Some common matrix factorizations…

Principal Components Analysis

- PCA (Principal Components Analysis)
  - PCA computes the most meaningful basis a noisy, garbled data set. The hope is that this new basis will filter out the noise and reveal the hidden dynamics.

Example:



Singular Value Decomposition

- SVD (Singular Value Decomposition)
  - SVD is based on a theorem which says that a rectangular matrix A can be broken down into the product of three matrices $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ where $\mathbf{U}^T\mathbf{U} = \mathbf{I}_m$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}_n$; the columns of **U** are orthonormal eigenvectors of $\mathbf{A}\mathbf{A}^T$, the columns of **V** are orthonormal eigenvectors of $\mathbf{A}^T\mathbf{A}$, and **S** is a diagonal matrix containing the square roots of eigenvalues from **U** or **V** in descending order.

Example:

$$A = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 2/\sqrt{6} & 1/\sqrt{3} \\ 1/\sqrt{2} & -1/\sqrt{6} & 1/\sqrt{3} \\ 1/\sqrt{2} & 1/\sqrt{6} & -1/\sqrt{3} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

Why Nonnegative?

- Some datasets are intrinsically non-negative:
  - Counters (e.g., no. occurrences of each word in a text document)
  - Intensities (e.g., intensity of each color in an image)
  - Similarity matrices

- Data matrix X has only non-negative values:
  - Decompositions such as SVD may give a result with negative values
  - Negative values describe the absence of something
  - They have no natural interpretation

EPAT 2014
Ecole de Printemps sur l'Apprentissage arTificiel
La Calanque, Carry-le-Rouet
du 7 au 12 Juin 2014

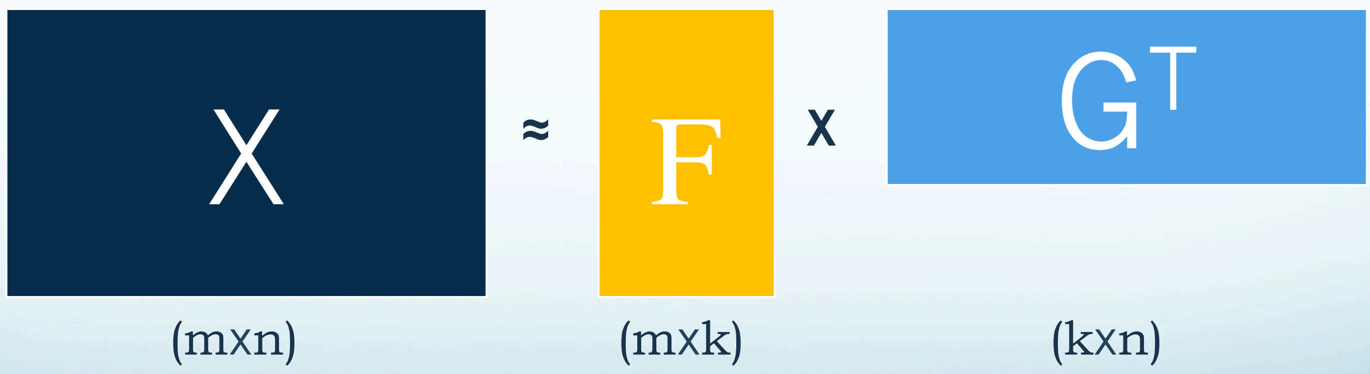
Standard NMF

[Lee and Seung, 1999]

- Standard NMF seeks the following decomposition:

$$X_+ \cong F_+ G_+^T,\ X \in R^{m\times n}, F \in R^{m\times k}, G \in R^{k\times n}$$

Example:



EPAT 2014
Ecole de Printemps sur l'Apprentissage arTificiel
La Calanque, Carry-le-Rouet
du 7 au 12 Juin 2014

Standard NMF

[Lee and Seung, 1999]

- Standard NMF seeks the following decomposition:

$$X_+ \cong F_+ G_+^T,\ X \in R^{m\times n}, F \in R^{m\times k}, G \in R^{k\times n}$$

Example:

$$\begin{pmatrix} 0.185 & 0.326 & 0.761 & 2.799 & 2.375 & 2.970 & 2.585 \\ 0.508 & 0.380 & 0.884 & 2.134 & 2.374 & 2.342 & 2.524 \\ 0.452 & 0.887 & 0.457 & 2.065 & 2.484 & 2.253 & 2.163 \\ 1.486 & 1.843 & 1.858 & 0.566 & 0.103 & 0.417 & 0.269 \\ 1.496 & 1.806 & 1.610 & 0.612 & 0.158 & 0.560 & 0.784 \end{pmatrix} \approx \begin{pmatrix} 0.0403 & 0.3695 \\ 0.0889 & 0.3149 \\ 0.1033 & 0.2945 \\ 0.3882 & 0.0002 \\ 0.3794 & 0.0210 \\ \hline 16.83 & 30.64 \end{pmatrix} \times \begin{pmatrix} 0.234 & 0.287 & 0.259 & 0.080 & 0.012 & 0.063 & 0.065 \\ 0.006 & 0.014 & 0.040 & 0.223 & 0.238 & 0.244 & 0.236 \end{pmatrix}$$

$$X_+ \approx F_+ G_+^T$$

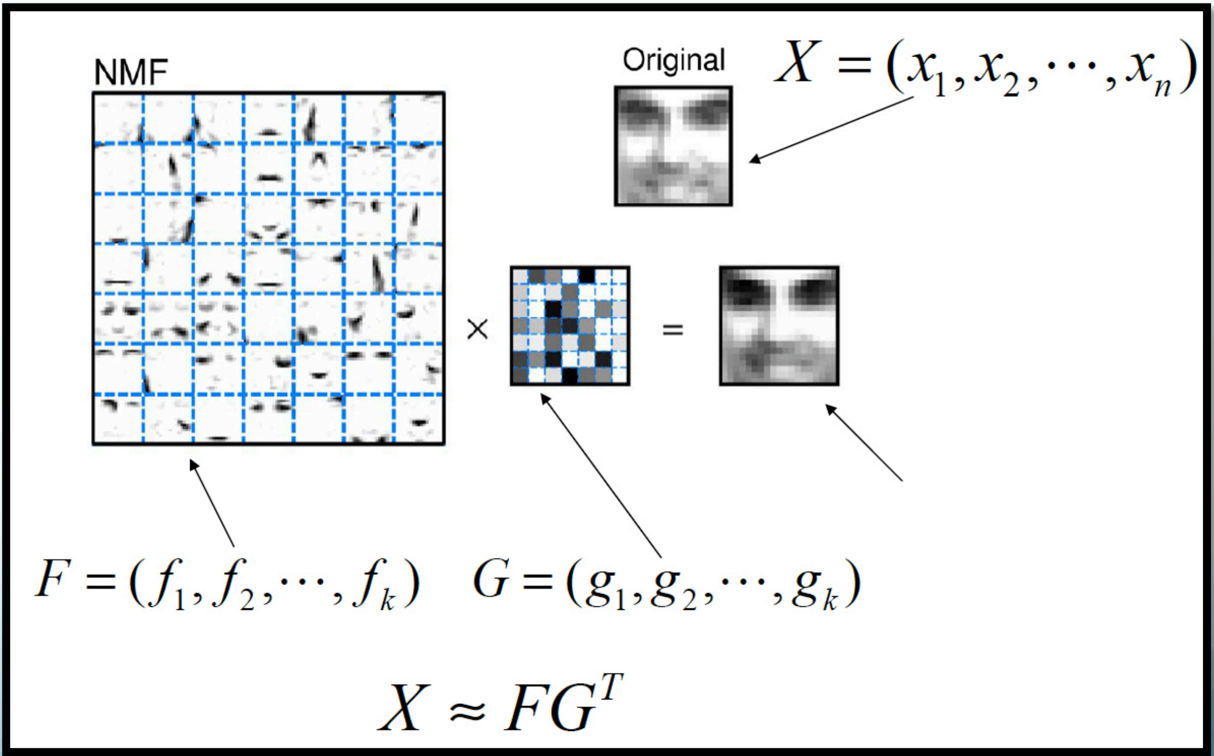
Standard NMF

[Lee and Seung, 1999]

- Standard NMF seeks the following decomposition:

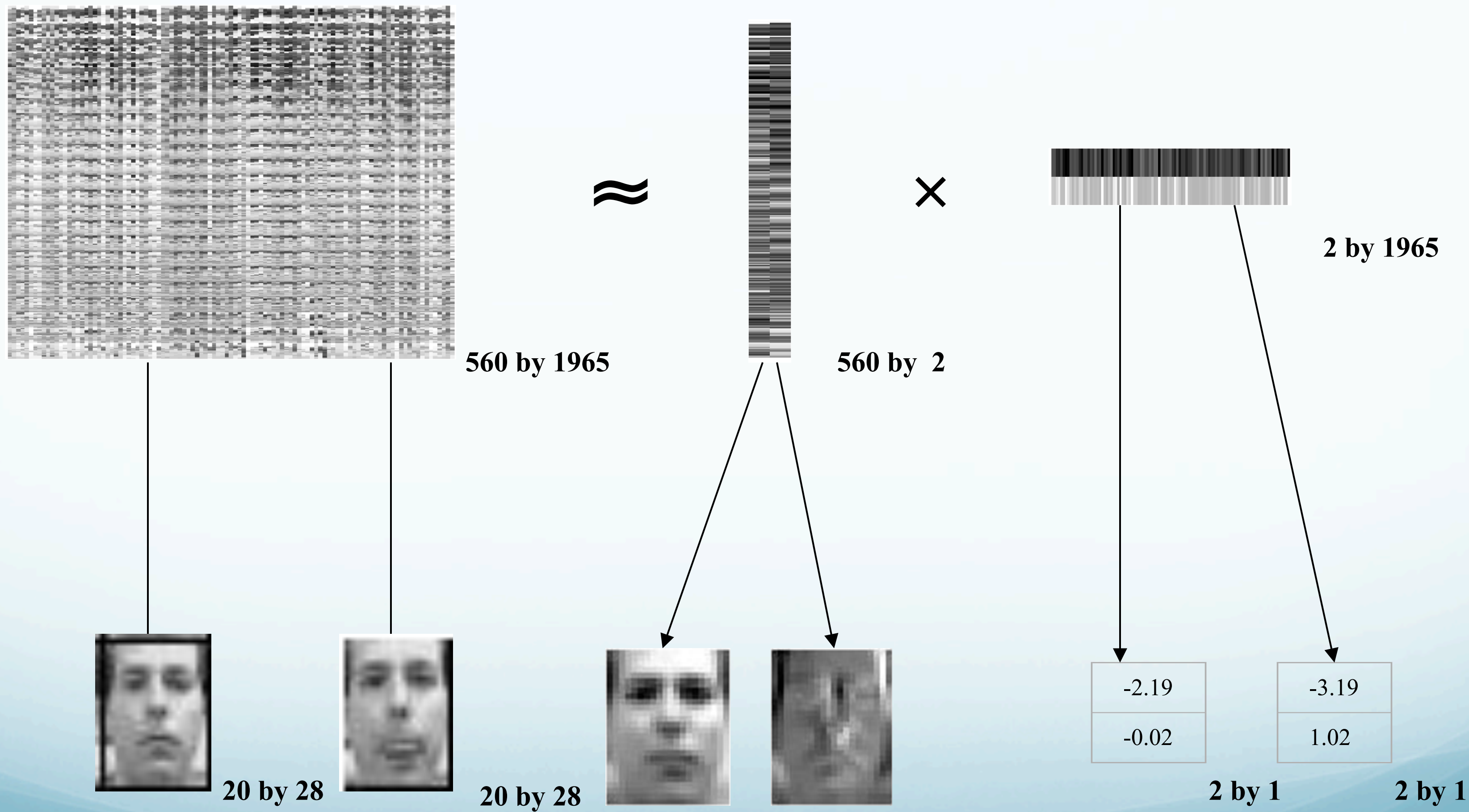
$$X_+ \cong F_+ G_+^T,\ X \in R^{m\times n},\ F \in R^{m\times k},\ G \in R^{k\times n}$$

Example:



Standard NMF

[Lee and Seung, 1999]



How to solve NMF?

- Problem is not convex:
  - Local optimum may not correspond to the global optimum
  - Little hope to find the global optimum
- But the problem is bi-convex:
  - For fixed F:

$$f(G) = \|X - FG\|_F^2$$

is convex.

General framework

- Gradient descent is generally slow
- Stochastic gradient descent is inappropriate
- Key approach: alternating minimization

Pick starting point F_0 and G_0

while not converged do:

1. Fix F and optimize G
2. Fix G and optimize F

end while

Convergence guaranties

Theorem: The objective function

$$f(F,G) = \|X - FG\|_F^2 \to \min$$

is non-increasing under the following update rules:

$$F = F \otimes \left[\frac{\partial f(F,G)}{\partial F}\right]_- \Big/ \left[\frac{\partial f(F,G)}{\partial F}\right]_+ = F \otimes \frac{XG^T}{FGG^T}$$

$$G = G \otimes \left[\frac{\partial f(F,G)}{\partial G}\right]_- \Big/ \left[\frac{\partial f(F,G)}{\partial G}\right]_+ = G \otimes \frac{F^T X}{F^T FG}$$

La Calanque, Carry-le-Rouet
du 7 au 12 Juin 2014

Multiplicative update rules example

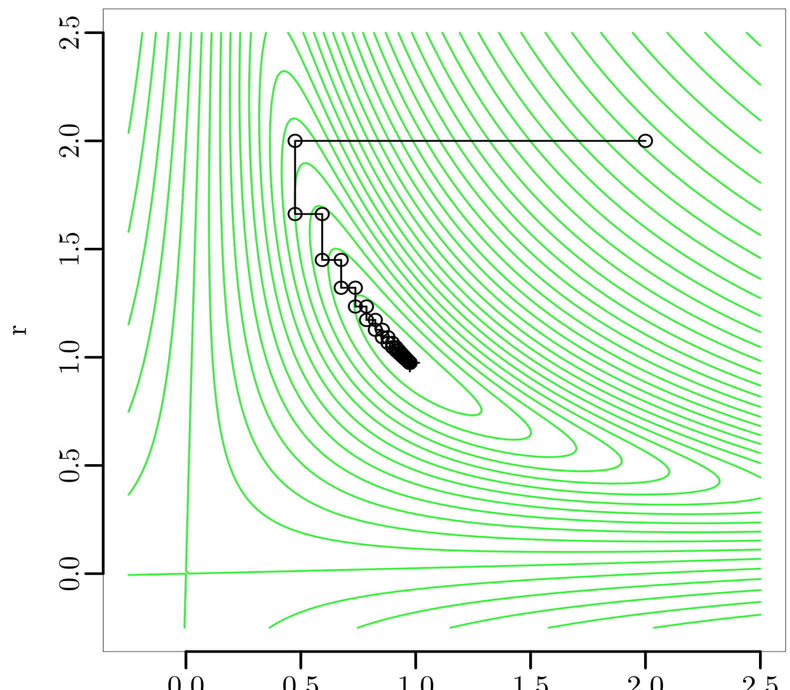
Example (multiplicative updates)

[Gemula and Miettinen, 2013]

- $f(l, r) = (1 - lr)^2 + 0.05(l^2 + r^2)$
- $l \leftarrow l \frac{1r - 0.05l}{lr^2}$
- $r \leftarrow r \frac{l1 - 0.05r}{l^2 r}$
-

Step	l	r
0	2	2
1	0.48	2
2	0.48	1.66
3	0.59	1.66
4	0.58	1.45
⋮	⋮	⋮
100	0.97	0.97

- Converges to local minimum



25 / 39

What if we don't want the initial data to be strictly non-negative?

But we still want to add non-negativity constraints on other factors

Semi-NMF
[Ding et al., 2006]

- Semi-NMF seeks the following factorization:

$$X_{\pm} = F_{\pm} G_{+}^{T}$$

- Why Semi-NMF?
  - We do not care if our data is non-negative
  - We do not know if basis vectors are non-negative
  - We **DO** want elements of **G** to be non-negative in order to interpret them as clusters assignments

EPAT 2014
Ecole de Printemps sur l'Apprentissage arTificiel
La Calanque, Carry-le-Rouet
du 7 au 12 Juin 2014

Update rules for Semi-NMF

- Step 0: Initialize **G**
- Step 1: Update **F** using the following expression:

$$F = XG(G^T G)^{-1}$$

- Step 2: Update **G** using the following equation:

$$G = G\sqrt{\frac{\left(X^T F\right)^{+} + G(F^T F)^{-}}{\left(X^T F\right)^{-} + G(F^T F)^{+}}}$$

where $A^{\pm} = \dfrac{(|A| \pm A)}{2}$.

Convergence guaranties

- **Theorem**:

  The update rules presented above decrease monotonically the objective function and converge to a fixed point that satisfies the Karush-Kuhn-Tucker(KKT) conditions.

- **Complexity**
  - Step 1: $t(mnk + nk^2)$
  - Step 2: $t(nmk + km^2 + n^2k)$

What if we want our basis vectors to be closer to the initial data?

Convex-NMF(C-NMF)

[Ding et al., 2006]

- Convex-NMF seeks the following factorization:

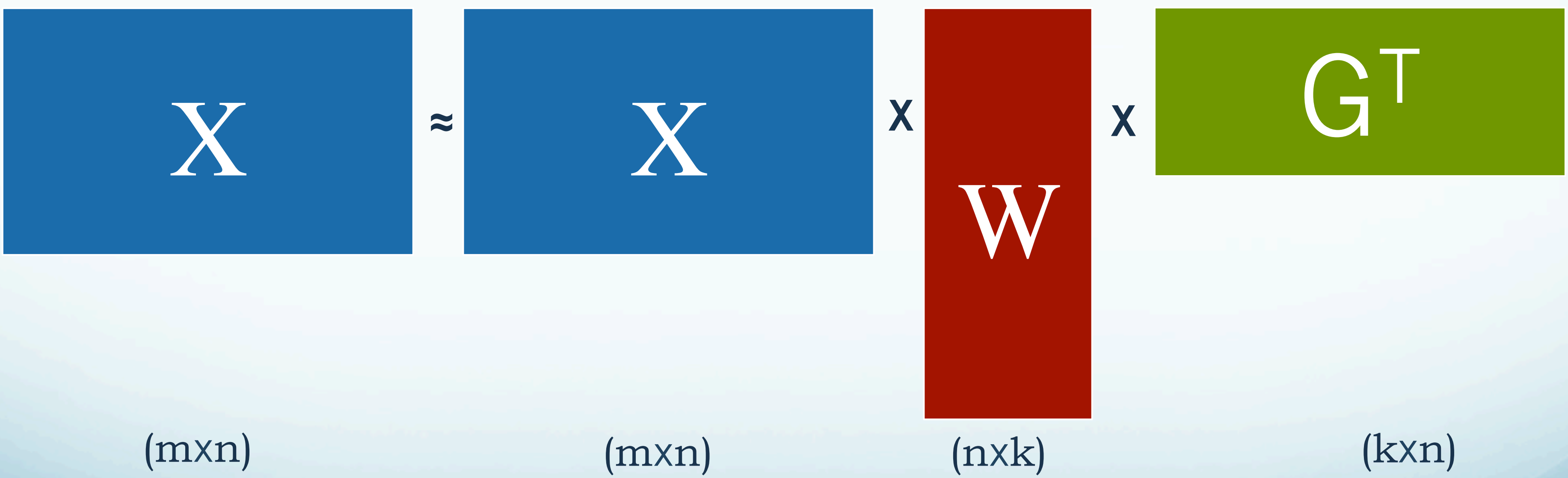
$$X_{\pm} \cong X_{\pm} W_{+} G_{+}^{T},\ X \in R^{m\times n},\ W \in R^{n\times k},\ G \in R^{k\times n}$$

- Why Convex-NMF?
  - We do not care if initial date is non-negative
  - Basis vectors lie in the vector space of the initial data so that they will capture the notion of centroids
  - Factors **W** and **G** are non-negative and tend to be very sparse

Convex-NMF(C-NMF)

[Ding et al., 2006]

$$F = XW \qquad X_{\pm} \approx X_{\pm} W_{+} G_{+}^{T}$$



Update rules for C-NMF

- Step 0: Initialize W and G.
- Step 1: Update G using the following expression:

$$G = G\sqrt{\frac{\left(X^T X\right)^+ W + GW^T (X^T X)^- W}{\left(X^T X\right)^- W + GW^T (X^T X)^+ W}}$$

- Step 2: Update W using the following equation:

$$W = W\sqrt{\frac{\left(X^T X\right)^+ G + (X^T X)^- WG^T G}{\left(X^T X\right)^- G + (X^T X)^+ WG^T G}}$$

C-NMF vs Semi-NMF

$$\begin{array}{cc} & \begin{array}{cc} \overbrace{\hphantom{1.3 \quad 1.8 \quad 4.8}}^{\text{Cluster 1}} & \overbrace{\hphantom{7.1 \quad 5.0 \quad 5.2 \quad 8.0}}^{\text{Cluster 2}} \end{array} \\ X = & \begin{pmatrix} 1.3 & 1.8 & 4.8 & 7.1 & 5.0 & 5.2 & 8.0 \\ 1.5 & 6.9 & 3.9 & -5.5 & -8.5 & -3.9 & -5.5 \\ 6.5 & 1.6 & 8.2 & -7.2 & -8.7 & -7.9 & -5.2 \\ 3.8 & 8.3 & 4.7 & 6.4 & 7.5 & 3.2 & 7.4 \\ -7.3 & -1.8 & -2.1 & 2.7 & 6.8 & 4.8 & 6.2 \end{pmatrix} \end{array}$$

$$F_{svd} = \left(\begin{array}{cc} -0.41 & 0.50 \\ 0.35 & 0.21 \\ 0.66 & 0.32 \\ -0.28 & 0.72 \\ -0.43 & -0.28 \\ \hline 25.5 & 15.6 \end{array}\right),$$

$$F_{semi} = \left(\begin{array}{cc} 0.05 & 0.27 \\ 0.40 & -0.40 \\ 0.70 & -0.72 \\ 0.30 & 0.08 \\ -0.51 & 0.49 \\ \hline 20.3 & 23.0 \end{array}\right), F_{conv} = \left(\begin{array}{cc} 0.31 & 0.53 \\ 0.42 & -0.30 \\ 0.56 & -0.57 \\ 0.49 & 0.41 \\ -0.41 & 0.36 \\ \hline 31.0 & 39.3 \end{array}\right),$$

$$G_{svd} = \begin{pmatrix} 0.25 & 0.05 & 0.22 & -0.45 & -0.44 & -0.46 & -0.52 \\ 0.50 & 0.60 & 0.43 & 0.30 & -0.12 & 0.01 & 0.31 \end{pmatrix}$$

$$G_{semi} = \begin{pmatrix} 0.61 & 0.89 & 0.54 & 0.77 & 0.14 & 0.36 & 0.84 \\ \boxed{0.12} & \boxed{0.53} & \boxed{0.11} & 1.03 & 0.60 & 0.77 & 1.16 \end{pmatrix}$$

$$G_{conv} = \begin{pmatrix} 0.31 & 0.31 & 0.29 & \boxed{0.02} & \boxed{0} & \boxed{0} & \boxed{0.02} \\ 0 & 0.06 & 0 & 0.31 & 0.27 & 0.30 & 0.36 \end{pmatrix}$$

$$\|X - FG^T\| = 0.27940, 0.27944, 0.30877$$

Convergence guaranties

- **Theorem**:

  The update rules presented above decrease monotonically the objective function and converge to a fixed point that satisfies the KKT conditions.

- **Complexity**
  - Step 1: $n^2m + t(2n^2k + nk^2)$
  - Step 2: $t(2n^2k + 2nk^2)$

What if we want to work with matrices based on similarities between objects but not the objects themselves?

Kernels and Gram matrices

Kernel is a function *k*:

$$k : \aleph \times \aleph \to \Re, \ (x, x') \to k(x, x')$$

satisfying

$$\forall \, (x, x') \in \aleph, \ k(x, x') = \left\langle \Phi(x), \Phi(x') \right\rangle$$

where Φ maps into some dot product space *H*, sometimes called the feature space.

Gram matrix of a kernel function *k* w.r.t a set of vectors $x_1, \ldots, x_n$ is a matrix

$$K^{n \times n} = \left(k(x_i), k(x_j) \right)_{ij}$$

Kernel functions

- Different similarity measures can be used as a kernel functions. For instance:
  - Linear kernel

$$k(x,x') = x^T x' + c$$

  - Polynomial kernel

$$k(x,x') = (ax^T x' + c)^d$$

  - Gaussian kernel

$$k(x,x') = \exp^{(-\frac{\|x-x'\|^2}{2\sigma^2})}$$

... etc

Kernel NMF(K-NMF)

[Zhang, 2006]

- Kernel NMF is a natural extension of C-NMF. It seeks the following decomposition:

$$\mathrm{K} \cong \mathrm{K}W_{+}\mathrm{G}_{+}^{T},\ \mathrm{K} \in \mathrm{R}^{n\times n},\ W \in \mathrm{R}^{n\times k},\ \mathrm{G} \in \mathrm{R}^{k\times n}$$

  where **K** is a Gram matrix of some arbitrary kernel function *k*.

- Why K-NMF?
  - Sometimes clustering based on similarities between objects gives better results
  - Some kernels preserve the non-negativity of data
  - Gram matrix can help to work with confidential data

Update rules for K-NMF

Obviously the update rules and convergence quarantines are the same as for C-NMF.

BUT!

Storing and calculating Gram matrices can lead to huge computational efforts.

AND!

We usually do not know how to choose an appropriate kernel function and its parameters beforehand

What if we want to consider data points in a graph model?

Considering a model similar to the Spectral Clustering.

Symmetric NMF(Sym-NMF)
[Kuang et al.,2012]

- Symmetric NMF seeks the following decomposition:

$$\mathbf{K} \cong \mathbf{G}_{+}\mathbf{G}_{+}^{T}, \mathbf{K} \in \mathbf{R}^{n\times n}, \mathbf{G} \in \mathbf{R}^{k\times n}$$

  where K is a Gram matrix calculated using any arbitrary kernel function with respect to initial data set.

- Why Sym-NMF?
  - It can be proved that Sym-NMF works as a spectral clustering method
  - It can be used for data which clusters lie on a nonlinear manifold

Update rules for Sym-NMF

- For non-negative **K**, **H** can be updated as follows:

$$H = H\left(0.5 + 0.5\frac{(KH)}{HH^T H}\right)$$

- Otherwise using Newton-liked method with Hessian estimations

Convergence guaranties

- **Theorem**:

  The update rules presented above decrease monotonically the objective function and converge to a fixed point that satisfies the KKT conditions.

- **Complexity**
  - $O(n^3k)$ **!!!**

What if we want to impose additional constraints on our model?

For example orthogonality.

Uni-Orthogonal NMF(UONMF)

[Ding et al., 2005]

- Uni-Orthogonal NMF takes the following form:

$$X_+ \cong F_+ G_+,\ X \in R^{m \times n},\ F \in R^{m \times k}, G \in R^{k \times n}\ s.t.\ F^T F = I\ \ or\ G^T G = I$$

- Why UONMF?
  - In case of orthogonal constraints imposed on **F** we obtain a dictionary with distinct basis vectors
  - In case of orthogonal constraints imposed on **G** we force our clusters to be as different as possible

Update rules for UONMF

- If constraints are added to the objective function [Mirzal, 2010] the update rules are:

If we impose orthogonality on **G**:

$$F = F\left(\frac{XG^T}{FGG^T}\right) \qquad G = G\left(\frac{F^T X + G}{F^T FG + GG^T G}\right)$$

- If solved as a constrained optimization problem:

$$F = F\left(\frac{XG^T}{FGG^T}\right) \qquad G = G\left(\frac{F^T X}{F^T XG^T G}\right)$$

Convergence guaranties

- The update rules presented in the original work are derived under assumption that off-diagonal elements of the Lagrangian matrix are equal to zero. Thus, the update rules have a non-increasing property of this assumption is true.
- The update rules from [Mirzal, 2010] have a robust convergence proof.

What if we impose constraints on both factors?

Bi-Orthogonal NMF

[Ding et al., 2005]

- Bi-Orthogonal NMF takes the following form:

$$X_+ \cong F_+ S_+ G_+,\ X \in R^{m\times n},\ F \in R^{m\times k}, S \in R^{k\times k}, G \in R^{k\times n}$$

$$s.t.\ F^T F = I \ \ and\ G^T G = I$$

- Why BONMF?
  - Can be seen as a co-clustering approach where **F** is a clustering of features and **G** is a clustering of data.
  - Gives unique matrix factorization!!!

Update rules for BONMF

- If constraints are added to the objective function [Mirzal, 2010] the update rules are:

$$F = F\left(\frac{XG^T S + F}{FSGG^T S^T + FF^T F}\right) \quad S = S\left(\frac{F^T XG^T}{F^T FSGG^T}\right) \quad G = G\left(\frac{S^T F^T X + G}{S^T F^T FSG + GG^T G}\right)$$

- If solved as a constrained optimization problem:

$$F = F\left(\frac{XG^T S^T}{FF^T XG^T S^T}\right) \quad S = S\left(\frac{F^T XG^T}{F^T FSGG^T}\right) \quad G = G\left(\frac{S^T F^T X}{F^T S^T XG^T G}\right)$$

Convergence guaranties

- The update rules presented in the original work are derived under assumption that off-diagonal elements of the Lagrangian matrix are zero. Thus, the update rules have a non-increasing property of this assumption is true.
- The update rules from [Mirzal, 2010] have a robust convergence proof.

Is there another way to impose orthogonality on the set of basis vectors?

Projective NMF(PNMF)

[Yuan et al., 2007]

- Projective NMF seeks the following decomposition:

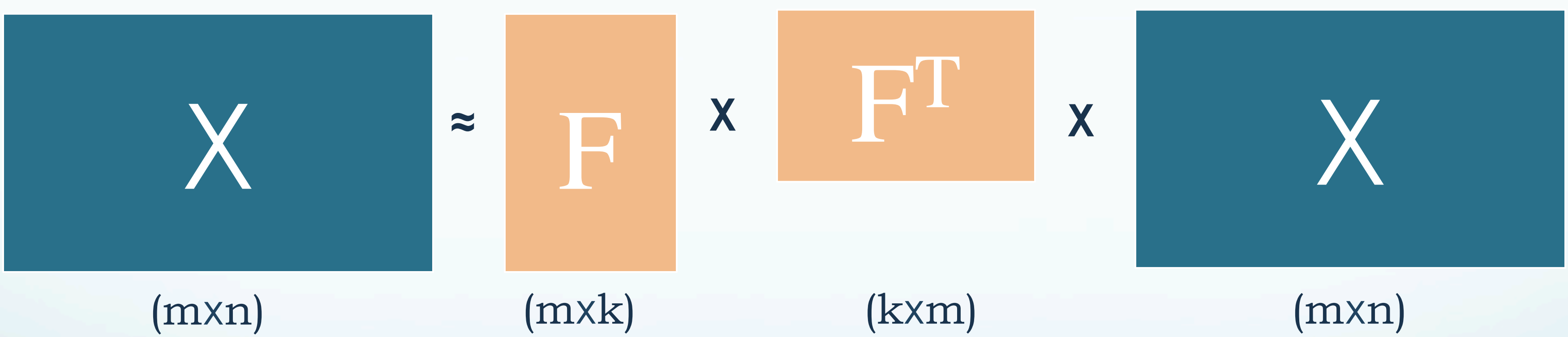
$$X_+ \cong F_+ F_+^T X_+, \; X \in R^{m \times n}, F \in R^{m \times k}$$

- Why Projective NMF?
  - Can be useful for dictionary learning
  - Gives very sparse basis vectors that can have good discriminative power.

Projective NMF(PNMF)

[Yuan et al., 2007]

$$X_+ \cong F_+ F_+^T X_+,\ X \in R^{m\times n},\ F \in R^{m\times k}$$



Update rules for PNMF

- Update **F** using the following expression:

$$F = F\left(\frac{XX^T F}{FF^T XX^T F + XX^T FF^T F}\right)$$

- Normalize columns of **F**:

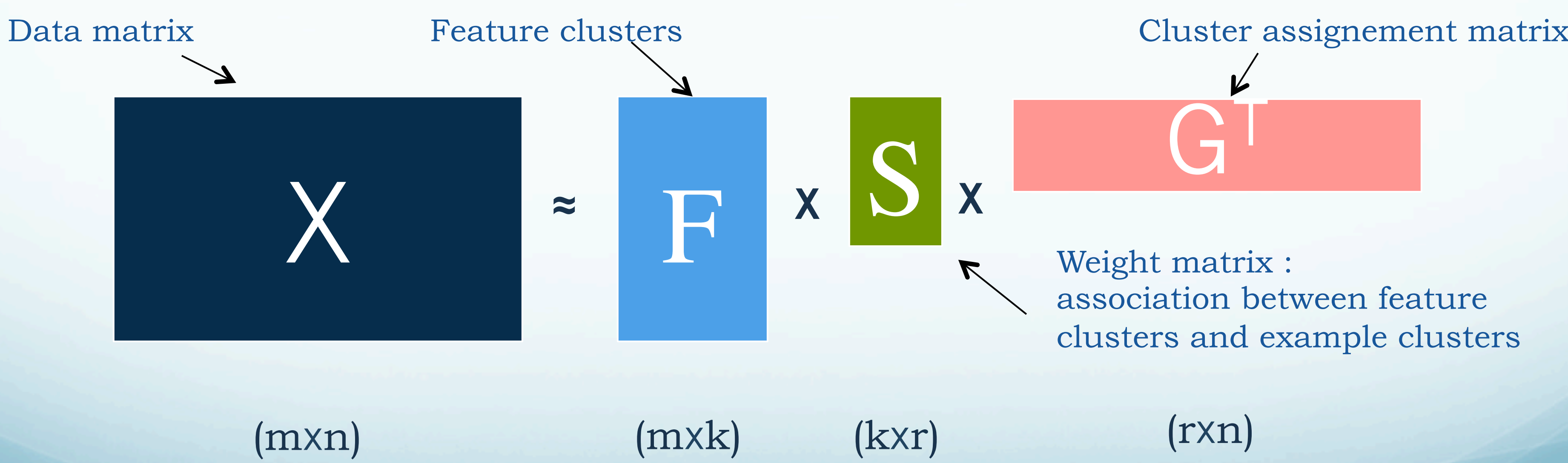
$$F = \frac{F}{\max_i(\|f_i\|)}$$

What does it mean more sparse?

- The fraction of non-zero elements in a matrix is called **sparsity.**

Tri-NMF

$$X \approx F\ S\ G^T$$



Tri-NMF

$$
\begin{cases}
\{F,S,G\} = \underset{F,S,G}{\arg\min}\, D_F\left(X \| FSG\right) \\
\qquad\qquad = \underset{F,S,G}{\arg\min} \left\| X - F\ S\ G^T \right\|_F^2 \\
s.c \quad F \geq 0, S \geq 0, G \geq 0 \quad et \quad F^T F = I, G^T G = I
\end{cases}
$$

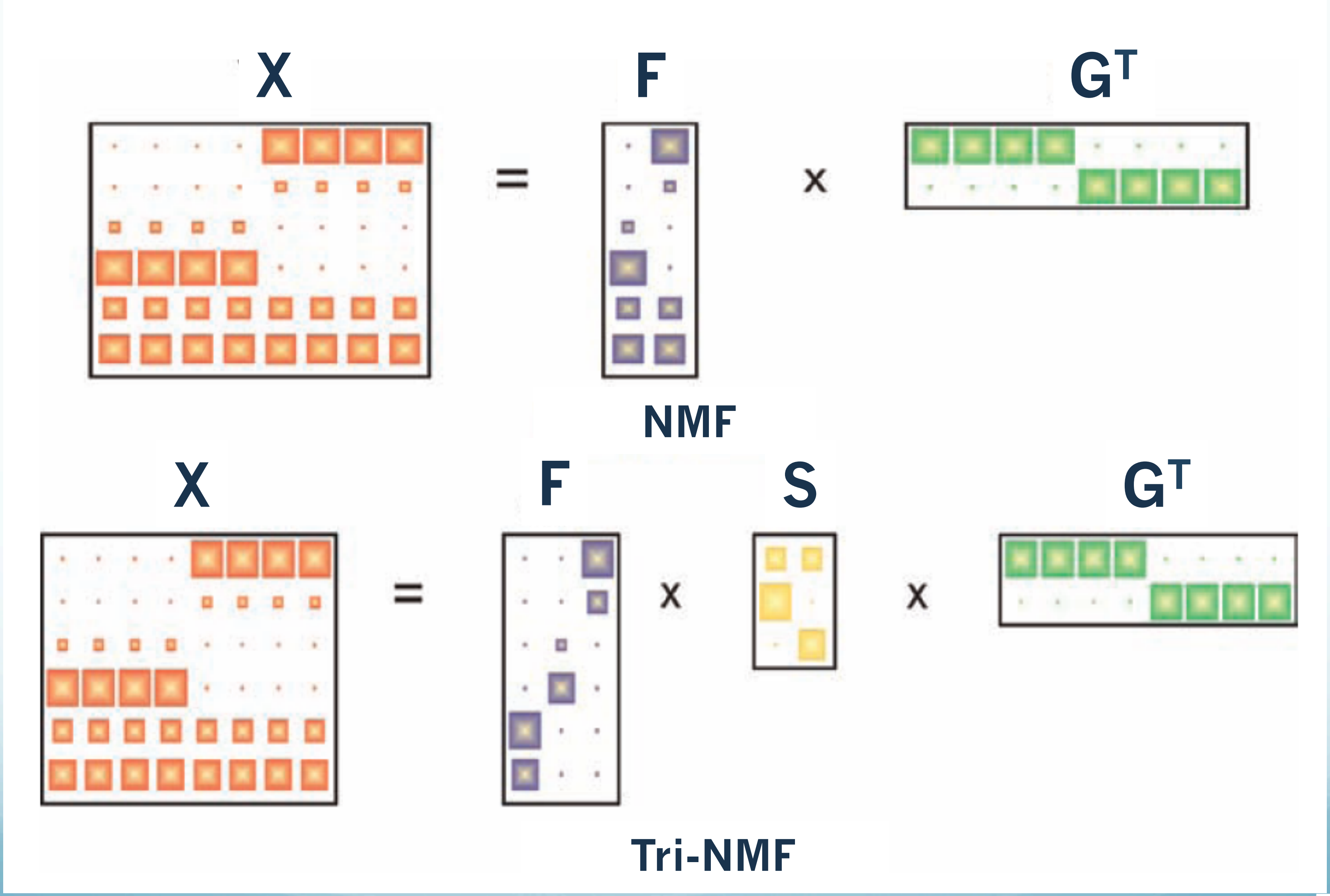
Tri-NMF

$$G_{jk} \leftarrow G_{jk} \sqrt{\frac{\left(X^T F S\right)_{jk}}{\left(G G^T X^T F S\right)_{jk}}}$$

$$F_{ik} \leftarrow F_{ik} \sqrt{\frac{\left(X G S^T\right)_{ik}}{\left(F F^T X G S^T\right)_{ik}}}$$

$$S_{ik} \leftarrow S_{ik} \sqrt{\frac{\left(F^T X G\right)_{ik}}{\left(F^T F S G^T G\right)_{ik}}}$$

Tri-NMF vs NMF



Tri-NMF vs NMF

Cluster 1 Cluster 2

$$\mathbf{X} \quad \begin{pmatrix} 0.185 & 0.326 & 0.761 & 2.799 & 2.375 & 2.970 & 2.585 \\ 0.508 & 0.380 & 0.884 & 2.134 & 2.374 & 2.342 & 2.524 \\ 0.452 & 0.887 & 0.457 & 2.065 & 2.484 & 2.253 & 2.163 \\ 1.486 & 1.843 & 1.858 & 0.566 & 0.103 & 0.417 & 0.269 \\ 1.496 & 1.806 & 1.610 & 0.612 & 0.158 & 0.560 & 0.784 \end{pmatrix}$$

$$F_{nmf} = \left(\begin{array}{cc} 0.0403 & 0.3695 \\ 0.0889 & 0.3149 \\ 0.1033 & 0.2945 \\ 0.3882 & 0.0002 \\ 0.3794 & 0.0210 \\ \hline 16.83 & 30.64 \end{array}\right), F_{Tri} = \left(\begin{array}{cc} 0.0000 & 0.3704 \\ 0.0215 & 0.3228 \\ 0.0320 & 0.3068 \\ 0.4773 & 0.0000 \\ 0.4692 & 0.0000 \\ \hline 2.6172 & 3.4036 \end{array}\right)$$

$$G_{nmf} = \begin{pmatrix} 0.234 & 0.287 & 0.259 & 0.080 & 0.012 & 0.063 & 0.065 \\ 0.006 & 0.014 & 0.040 & 0.223 & 0.238 & 0.244 & 0.236 \end{pmatrix}$$

$$G_{Tri} = \begin{pmatrix} 0.270 & 0.335 & 0.333 & 0.034 & 0.000 & 0.009 & 0.020 \\ 0.000 & 0.000 & 0.000 & 0.239 & 0.248 & 0.264 & 0.250 \end{pmatrix}$$

$$S_{Tri-factor} = \begin{pmatrix} 4.3626 & 1.0136 \\ 1.4824 & 8.4000 \end{pmatrix}$$

What if we want more than three factors?

General model of NMF for *k* different matrices.

Multilayer NMF(MultiNMF)

[Cichocki et al., 2006]

In Multilayer NMF we build up a system that has many layers or cascade connections:

- First of all we perform NMF on the initial data

$$X \cong F_1 G_1,\ X \in R^{m\times n},\ F_1 \in R^{m\times k},\ G_1 \in R^{k\times n}$$

- Then we use matrix **G** for further decompositions

$$G_{i-1} \cong F_i G_i\ \ \forall\ i = 1 \ldots L$$

- We stop when some stopping criteria is satisfied. Finally, we obtain the following factorization:

$$X \cong F_1 F_2 \ldots F_L G_L .$$

Why Multilayer NMF?

- At each level the basis vectors' sparsity is growing
- Better clutering results due to hierarchically learned representations
- Better numerical stability

How to take into account time shifts in data?

For example if we work with audio signals.

Convolutive NMF(ConvNMF)

[O'Grady et Pearlmutter, 2006]

- Convolutive NMF is of the following form:

$$X \cong \sum_{i=1}^{t} F_i \overset{i\rightarrow}{G},\ X \in R^{m\times n},\ F_i \in R^{m\times k},\ \overset{i\rightarrow}{G} \in R^{k\times n}$$

- Why Convolutive NMF?
  - Processing horizontally shifted versions of the initial matrix allows to discover more efficiently the structure of data whose frequency varies in time.
  - It can be applied to audio signals analysis

Update rules for ConvNMF

- Update **F** using the following expression

$$F = F + \eta_F \left(\frac{X}{\sum_{i=0}^{t} F_i \overset{i\rightarrow}{G}} \overset{i\rightarrow}{G^T} - 1 \overset{i\rightarrow}{G^T} \right)$$

- Rescale all the columns of **F** to the unit length
- Update G as follows:

$$H = H \left({F^T \frac{\overset{\leftarrow i}{X}}{\sum_{i=0}^{t} F_i \overset{i\rightarrow}{G}}} \Big/ {F^T 1} \right)$$

What if we want to find a consensus between different views of data?

Multiview NMF

[Liu et al., 2013]

- Multiview NMF has the following objective function:

$$\sum_{v=1}^{n_v} \left\| X^{(v)} - F^{(v)} G^{(v)} \right\|_F^2 + \sum_{v=1}^{n_v} \lambda_v \left\| G^{(v)} - G^{(*)} \right\|_F^2$$

- Why Multiview NMF?
  - Different views can provide different information about data
  - Consensus technique in the NMF framework

EPAT 2014
Ecole de Printemps sur l'Apprentissage arTificiel
La Calanque, Carry-le-Rouet
du 7 au 12 Juin 2014

Update rules for Multiview NMF

- For each view do:
  - Update **F** based on the following update rule:

$$F = F\frac{XG + \lambda_v \sum_{l=1}^{n} G_{l,\bullet} G^{*}_{l,\bullet}}{FG^T G + \sum_{p=1}^{m} G_{p,\bullet} \sum_{l=1}^{n} G^{2}_{l,\bullet}}$$

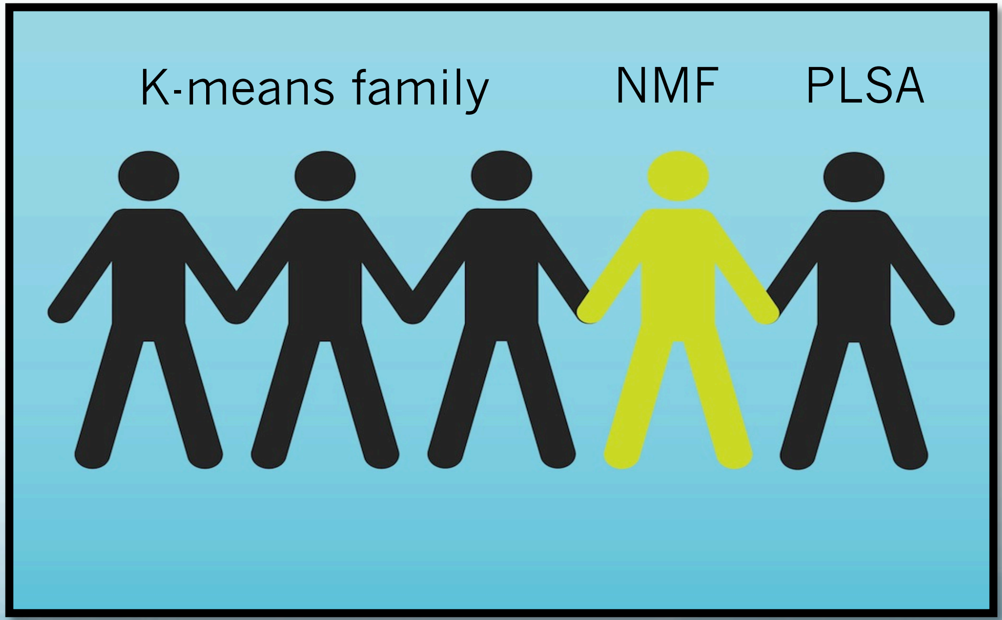
  - Update **G** using the following expression:

$$G = G\frac{X^T F + \lambda_v G^*}{GF^T F + \lambda_v G}$$

- Calculate the consensus matrix **G***:

$$G^* = \frac{\sum_{v=1}^{n_v} \lambda_v G^{(v)} Q^{(v)}}{\sum_{v=1}^{n_v} \lambda_v}, \quad Q^{(v)} = diag(\sum_{i=1}^{m} F^{(v)}_{i,1} \sum_{i=1}^{m} F^{(v)}_{i,2} \dots \sum_{i=1}^{m} F^{(v)}_{i,k})$$

What are the relationships between NMF and other Machine Learning techniques?



NMF vs K-means

NMF		K-means
• G-orthogonal NMF • Semi-NMF • Convex-NMF • Kernel NMF	[Ding et al.,2006]	Relaxed K-means clustering
• Orthogonal Symmetric NMF	[Kuang et al.,2012] →	Kernel K-means clustering

Simple example

$$X = \begin{array}{c} \\ \end{array}\overbrace{}^{\text{cluster 1}}\ \overbrace{}^{\text{cluster 2}}$$

$$X = \begin{pmatrix} 1.3 & 1.8 & 4.8 & 7.1 & 5.0 & 5.2 & 8.0 \\ 1.5 & 6.9 & 3.9 & -5.5 & -8.5 & -3.9 & -5.5 \\ 6.5 & 1.6 & 8.2 & -7.2 & -8.7 & -7.9 & -5.2 \\ 3.8 & 8.3 & 4.7 & 6.4 & 7.5 & 3.2 & 7.4 \\ -7.3 & -1.8 & -2.1 & 2.7 & 6.8 & 4.8 & 6.2 \end{pmatrix}$$

$$F_{\text{svd}} = \begin{pmatrix} -0.41 & 0.50 \\ 0.35 & 0.21 \\ 0.66 & 0.32 \\ -0.28 & 0.72 \\ -0.43 & -0.28 \end{pmatrix}, \quad F_{\text{semi}} = \begin{pmatrix} 0.05 & 0.27 \\ 0.40 & -0.40 \\ 0.70 & -0.72 \\ 0.30 & 0.08 \\ -0.51 & 0.49 \end{pmatrix}, \quad F_{\text{cnvx}} = \begin{pmatrix} 0.31 & 0.53 \\ 0.42 & -0.30 \\ 0.56 & -0.57 \\ 0.49 & 0.41 \\ -0.41 & 0.36 \end{pmatrix}, \quad C_{\text{Kmeans}} = \begin{pmatrix} 0.29 & 0.52 \\ 0.45 & -0.32 \\ 0.59 & -0.60 \\ 0.46 & 0.36 \\ -0.41 & 0.37 \end{pmatrix}$$

$$\| F_{convex} - C_{Kmeans} \| = 0.08$$

$$\| F_{semi} - C_{Kmeans} \| = 0.53$$

$$G^T_{\text{svd}} = \begin{pmatrix} 0.25 & 0.05 & 0.22 & -.45 & -.44 & -.46 & -.52 \\ 0.50 & 0.60 & 0.43 & 0.30 & -0.12 & 0.01 & 0.31 \end{pmatrix}$$

$$G^T_{\text{semi}} = \begin{pmatrix} 0.61 & 0.89 & 0.54 & 0.77 & 0.14 & 0.36 & 0.84 \\ 0.12 & 0.53 & 0.11 & 1.03 & 0.60 & 0.77 & 1.16 \end{pmatrix}$$

$$G^T_{\text{cnvx}} = \begin{pmatrix} 0.31 & 0.31 & 0.29 & 0.02 & 0 & 0 & 0.02 \\ 0 & 0.06 & 0 & 0.31 & 0.27 & 0.30 & 0.36 \end{pmatrix}$$

$$\| X - FG^T \| = 0.27940, 0.27944, 0.30877$$

SVD Semi Convex

NMF vs PLSI
[Ding et al., 2008]

Theorem:

Any (local) maximum likelihood solution of PLSI is a solution of NMF with KL divergence.

Experimental results on different data sets:

Disagreements between NMF and PLSI

	WebAce	CSTR	WebKB	Reuters	Log
A	0.083	0.072	0.239	0.070	0.010
B	0.029	0.025	0.056	0.051	0.010
C	0.022	0.013	0.052	0.040	0.012

All 3 type experiments begin with the same smoothed K-means. (A) Smoothed K-means to NMF. Smoothed K-means to PLSI. (B) Smoothed K-means to NMF to PLSI. (C) Smoothed K-means to PLSI to NMF.

NMF vs Spectral Clustering(Normalized Cut)

It can be proved that the formulation of SymNMF can be related as a generalized form of many graph clustering algorithms.

	Spectral clustering	SymNMF
Objective	$\min_{H^T H=I} \|A - HH^T\|_F^2$	$\min_{H \geq 0} \|A - HH^T\|_F^2$
Step 1	Obtain the global optimal $H_{n \times k}$ by computing k leading eigenvectors of A	Obtain a stationary point solution using some minimization algorithm
Step 2	Normalize each row of H	(no need to normalize H)
Step 3	Infer clustering assignments from the rows of H (e.g. by K-means)	The largest entry in each row of H indicates the clustering assignments

What are the applications of NMF for the real-world tasks?

Text mining

- Topic model: NMF as an alternative to PLSI ([Ding et al., 2008], [Gaussier et al.,2005])
- Document clustering([Xu et al., 2003], [Shahnaz et al.,2006])
- Topic detection and trend analysis, email analysis([Berry et al., 2005], [Keila et al.,2005], [Cao et al.,2008])

Image analysis and computer vision

- Image analysis and computer vision
  - Feature representation, sparse coding ([Lee et al., 99]; [Guillamet et al., 01]; [Hoyer et al., 02]; [Li et al. 01])
  - Video tracking ([Bucak et al., 07])

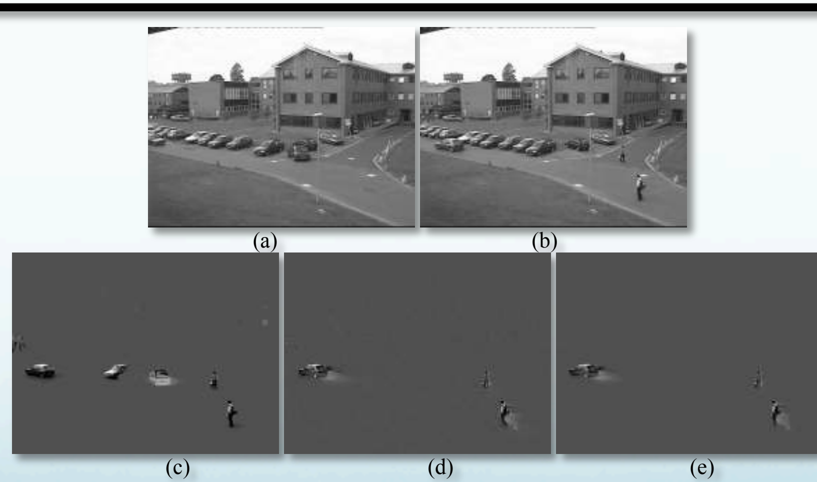
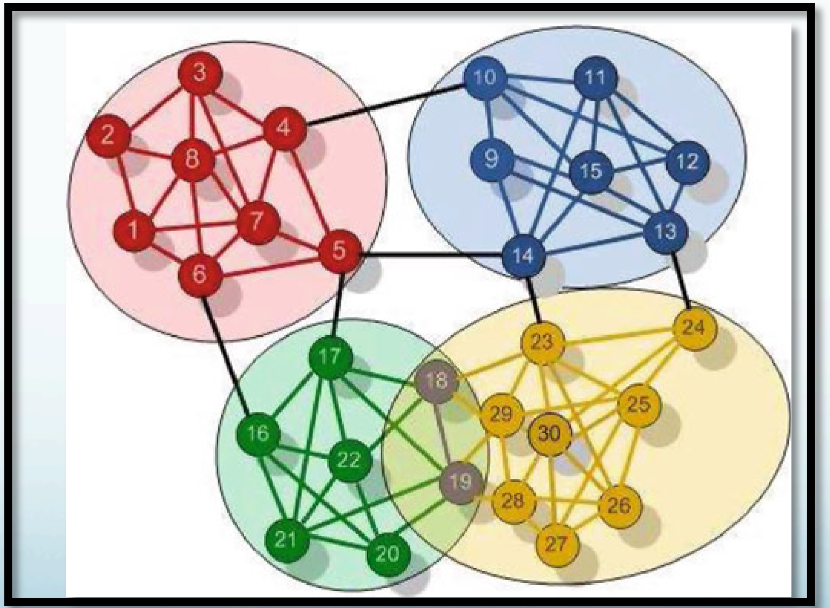


Fig. 2 (a) Original video frame 971. **(b)** Original video frame 1436. Reconstructed difference image obtained for the frame 1436 **(c)** by batch NMF with r=2, **(d)** by INMF with r=2, α =0.2 and **(e)** by IPCA with space size r=2, α =0.8.

EPAT 2014
Ecole de Printemps sur l'Apprentissage arTificiel
La Calanque, Carry-le-Rouet
du 7 au 12 Juin 2014

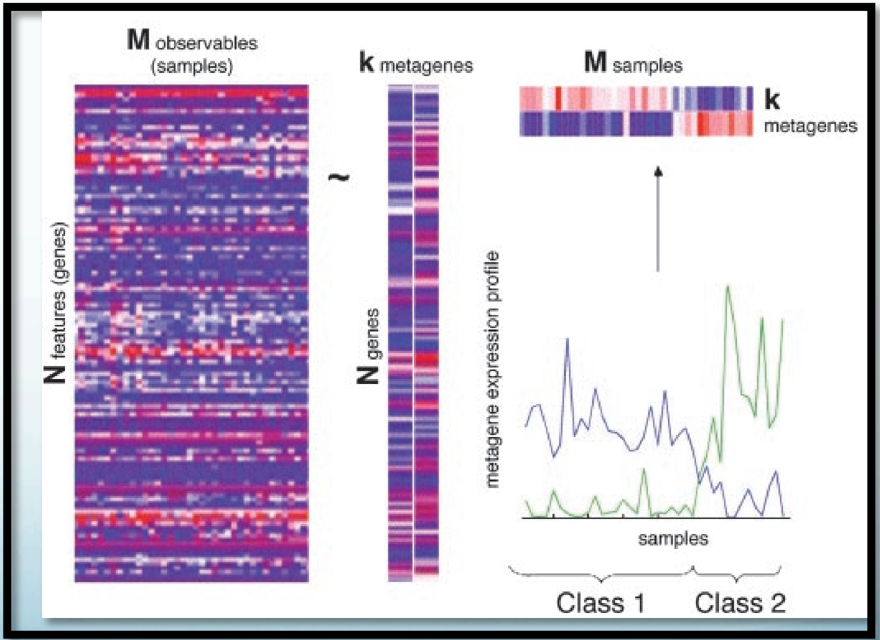
Social networks

- Social network analysis
  - Community structure and trend detection ([Chi et al., 07]; [Wang et al., 08])
  - Recommendation system ([Zhang et al., 06])



Bioinformatics

- Our goal is to discover hidden structures in biological data
  - Bioinformatics-microarray data analysis ([Brunet et al., 04], [H. Kim and Park, 07])



[Brunet, 2004]

Audio analysis

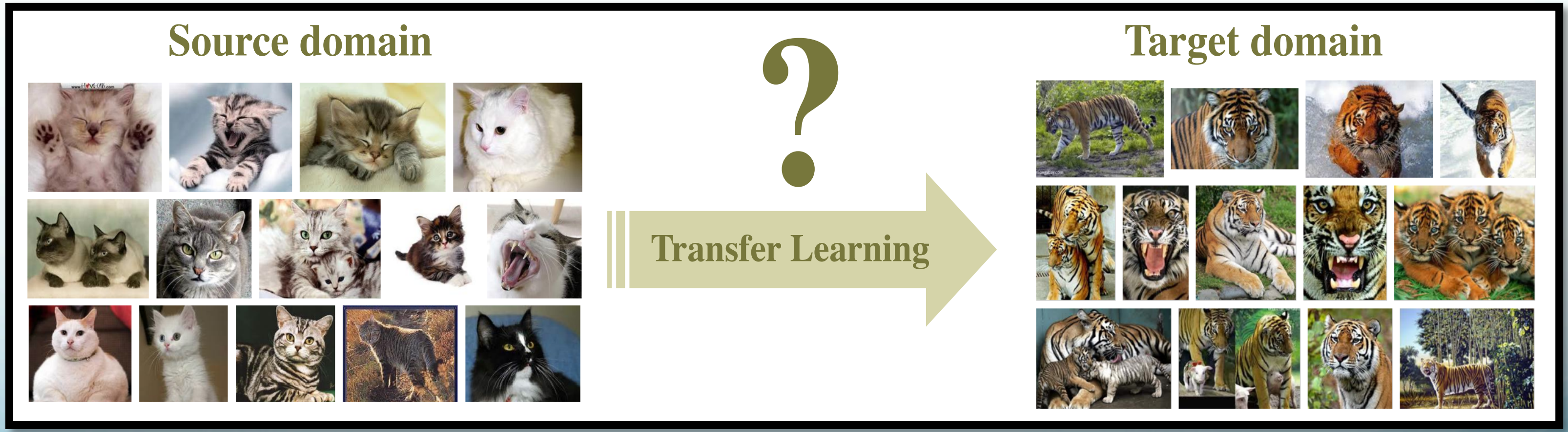
- The idea is to apply NMF to perform signal decomposition
  - Acoustic signal processing, blind source separating ([Cichocki et al., 04])

[Sapiro et al., 2012]

Transfer learning (1)

Definition

- Given a source domain $\mathbf{D_S}$ and a learning task $\mathbf{T_S}$, a target domain $\mathbf{D_T}$ and a target task $\mathbf{T_T}$, transfer learning aims to help improve the learning performance in $\mathbf{D_T}$ using knowledge gained from $\mathbf{D_S}$ and $\mathbf{T_S}$, where $\mathbf{D_S} \neq \mathbf{D_T}$ and $\mathbf{T_S} \neq \mathbf{T_T}$.



Transfer learning (2)

- Unsupervised transfer learning using kernel target alignment optimization ([Redko and Bennani, 2014])
- Unsupervised transfer learning using tri-factorization based on discovering distinct concepts ([Zhuang et al.,2013])
- Unsupervised transfer learning using Multilayer NMF ([Redko and Bennani, 2014])

Feel free to ask questions if you have any.