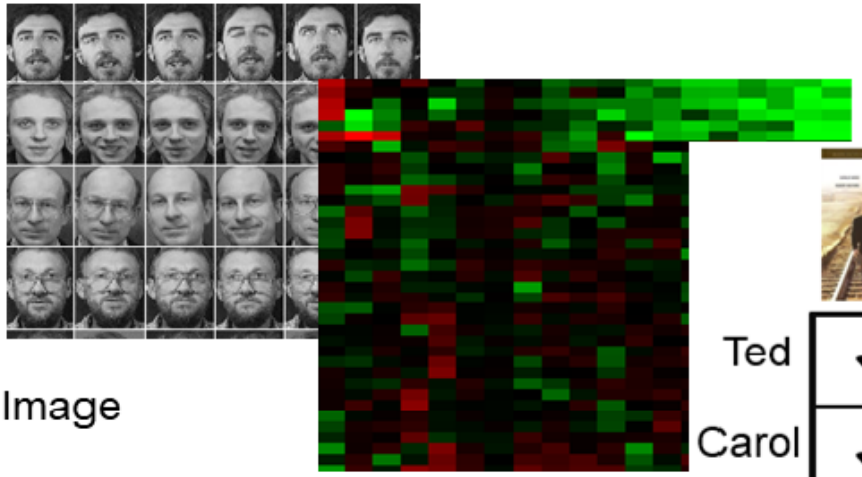
Nonnegative Matrix Factorization: Algorithms, Theory and Applications

Chihao Zhang

August 10, 2019

Data in Matrix



Image

Gene expression

Ted	✓	✓	✓
Carol	✓		
Bob			

User data

Adobe, the Adobe logo, Acrobat, the Acrobat logo, Acrobat Ca
Intelligent Document Platform, Adobe PDF, Adobe Reader, Adob
tiller, ePaper, Extreme, FrameMaker, Illustrator, InDesign, Mini
shop, Poetica, PostScript, and XMP are either registered trade
Systems Incorporated in the United States and/or other countri
either registered trademarks or trademarks of Microsoft Corpora
other countries. Apple, Mac, Macintosh, and Power Macintosh are
Inc., registered in the United States and other countries. IBM is
Corporation in the United States. Sun is a trademark or registe
tems, Inc. in the United States and other countries. UNIX is a re
Group. SVG is a trademark of the World Wide Web Consortium;
and held by its host institutions MIT, INRIA and Keio. Helvetica
marks of Linotype-Hell AG and/or its subsidiaries. Arial and Time
The Monotype Corporation registered in the U.S. Patent and Trac
tered in certain other jurisdictions. ITC Zapf Dingbats is a regist
Typeface Corporation. Ryumin Light is a trademark of Morisawa
are the property of their respective owners.

Text

Many data are organized in non-negative matrices.

Given a non-negative data matrix $X \in R_+^{m \times n}$ of m features and n samples.

Problem

How do we retrieval information from the high-dimensional and redundant primitive data ?

Dimension Reduction via Matrix Factorization

Principal Component Analysis

Assume X is centralized, using the minimizing reconstruction error

$$\min ||X - WH||_F^2$$
$$\text{s.t. } W^T W = I$$

where $W \in R^{m \times r}$ is the projection matrix

Singular Value Decomposition

$$X = U\Sigma V = \sum_{i=1}^{\min(m,n)} \sigma_i u_i v_i^T$$

where U and V are orthogonal matrix, $\Sigma = \text{diag}(\sigma_i)$. Use first r eigenvector as the representation of X

Many data are organized in non-negative matrices. PCA and SVD are popular data exploring tools for real-valued matrix.

Problem

How do we make use of the non-negativity of data?

Non-negative Matrix Factorization (NMF)

Approximate the primitive data matrix X by the factorization of two low-rank matrices.

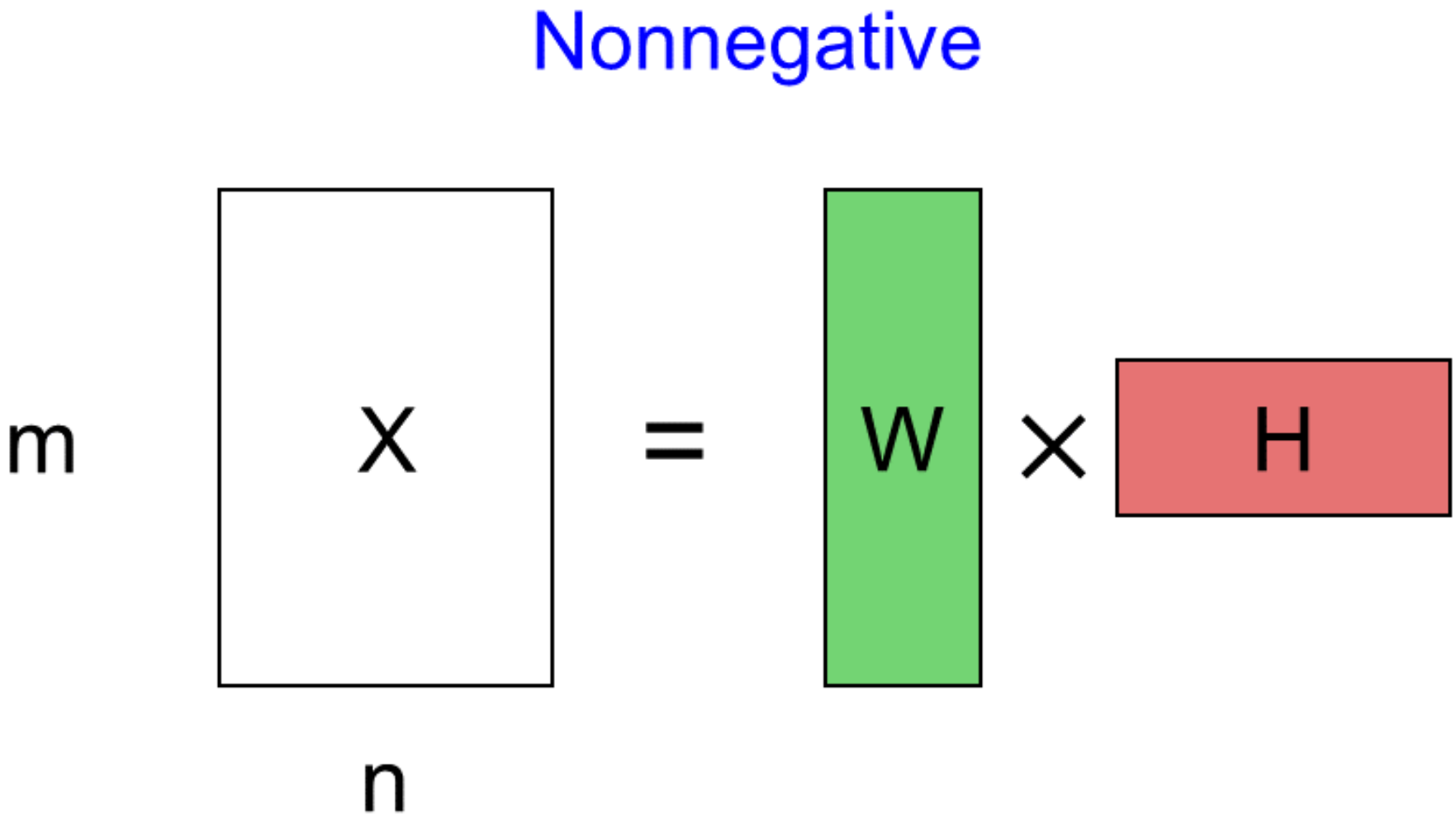
$$X \approx WH$$

where $X \in R_+^{m \times n}$, $W \in R_+^{m \times r}$, $H \in R_+^{r \times n}$, where $r < \min(m, n)$. Formally, we want to solve the following optimization problem:

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

where D is the divergence function that measures the distance between X and WH.

Illustration of NMF



Understanding NMF

Each column of data is approximated by

$$x_{\cdot i} \approx \sum_{i=1}^{r} w_{\cdot i} h_{ij}$$

where $w_{\cdot i}$ is the column vectors of W. Therefore, $x_{\cdot i}$ is the non-negative linear combination of $w_{\cdot i}$. W is referred as the basis matrix and H is the coefficient matrix.

Why Non-negative?

- Part-based representation
- Addictive components
- Natural sparsity

Intuitions of NMF

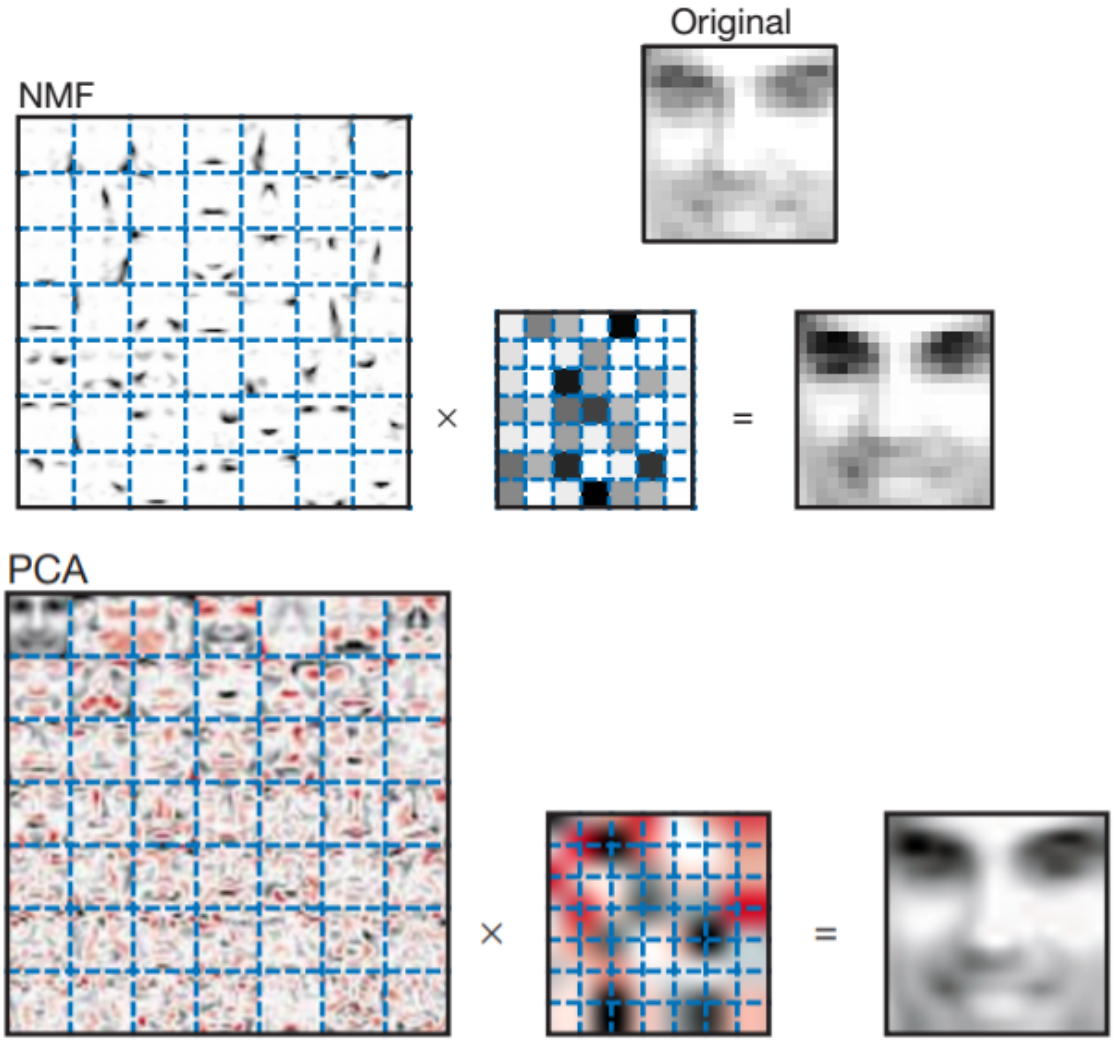


Figure: Representation of human face. [Lee and Seung, 1999]

History of NMF

NMF is more than 30-year old

- previous variants referred as
  - non-negative rank factorization
    [Jeter and Pye, 1981, Chen, 1984]
  - positive matrix factorization [Paatero and Tapper, 1994]
- poparized by [Lee and Seung, 1999] for ”learning the parts of objects”

Since then, widely used in various research areas for diverse applications

The Opt. Problem of NMF

Recall the general objective function of NMF:

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

where $W \geqslant 0$ and $H \geqslant 0$

How do We Measure the Similarity?

The divergence function is the way we measure the similarity of the primitive data matrix X and the approximated matrix WH. The most commonly used are

Frobenius Norm

$$D(X||WH) = ||X - WH||_F^2$$

Kullback-Leibler Divergence

$$D(X||WH) = \sum_{ij} \left(X_{ij} \ln \frac{X_{ij}}{(WH)_{ij}} - X_{ij} + (WH)_{ij} \right)$$

More Divergence Functions

The Kullback-Leibler divergence is the special form of α-Divergence [Cichocki et al., 2008]

α-Divergence

$$D_\alpha(p||q) = \frac{1}{\alpha(1-\alpha)} \int \alpha p + (1-\alpha)q - p^\alpha q^{1-\alpha} d\mu$$

where $\alpha \in (-\infty, +\infty)$

- As α approaches 0, KL-divergence

$$\lim_{\alpha \to 0} D_\alpha(p||q) = KL(p||q)$$

More Divergence Functions, Cont'd

- For $\alpha = 0.5$, Hellinger divergence

$$D_{0.5}(p||q) = 2\int \frac{(\sqrt{p} - \sqrt{q})^2}{q} d\mu$$

- For $\alpha = 2$, χ^2-divergence

$$D_2(p||q) = \frac{1}{2}\int \frac{(p-q)^2}{q} d\mu$$

Problem

How to choose the appropriate divergence function for a given application ?

Answer

No clear answer. From a statical perspective, the divergences can be determined based on a prior knowledge about the probability of noise.

- Frobenius norm Maximum likelihood estimator due to additive Gaussian noise
- KL divergence Maximum likelihood for Poisson process [Cichocki et al., 2009]

Images with Different Noise

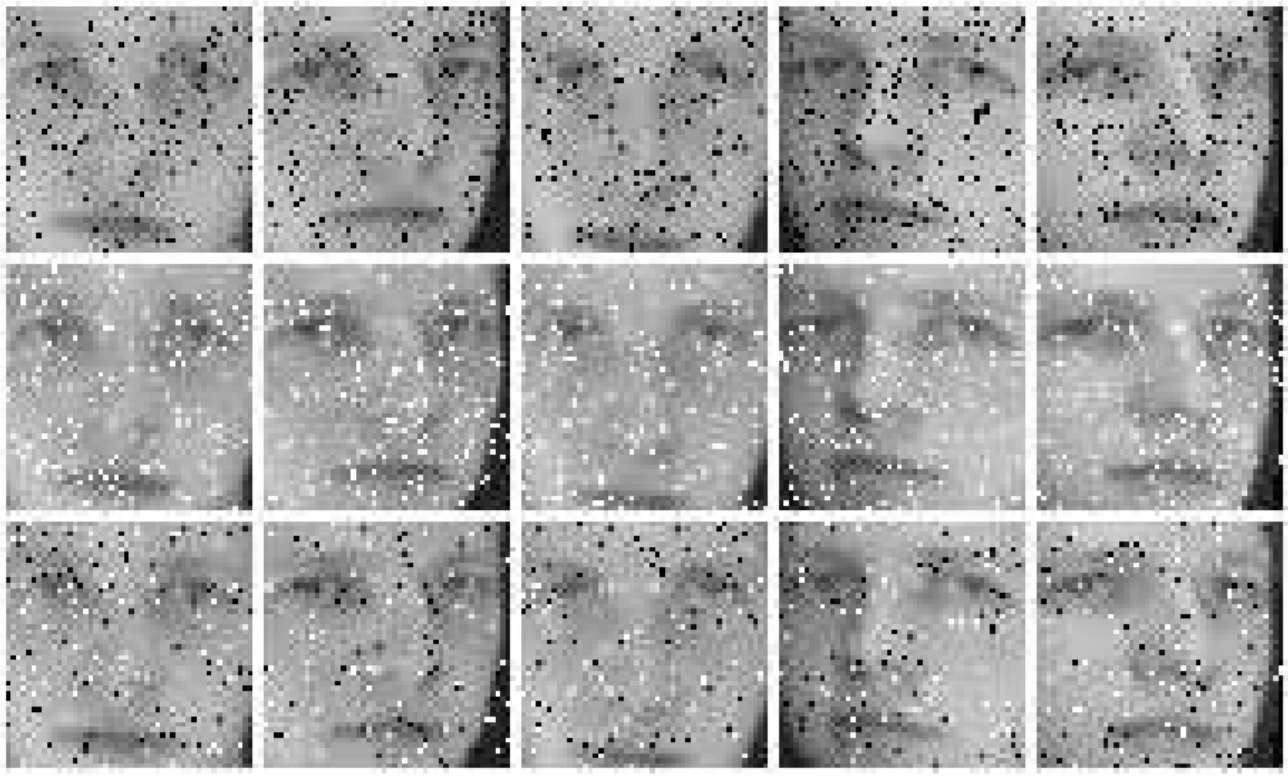


Figure: From top to bottom: images contaminated by pepper (black), salt (white), salt and pepper noise, respectively

Types of Noise and Divergences

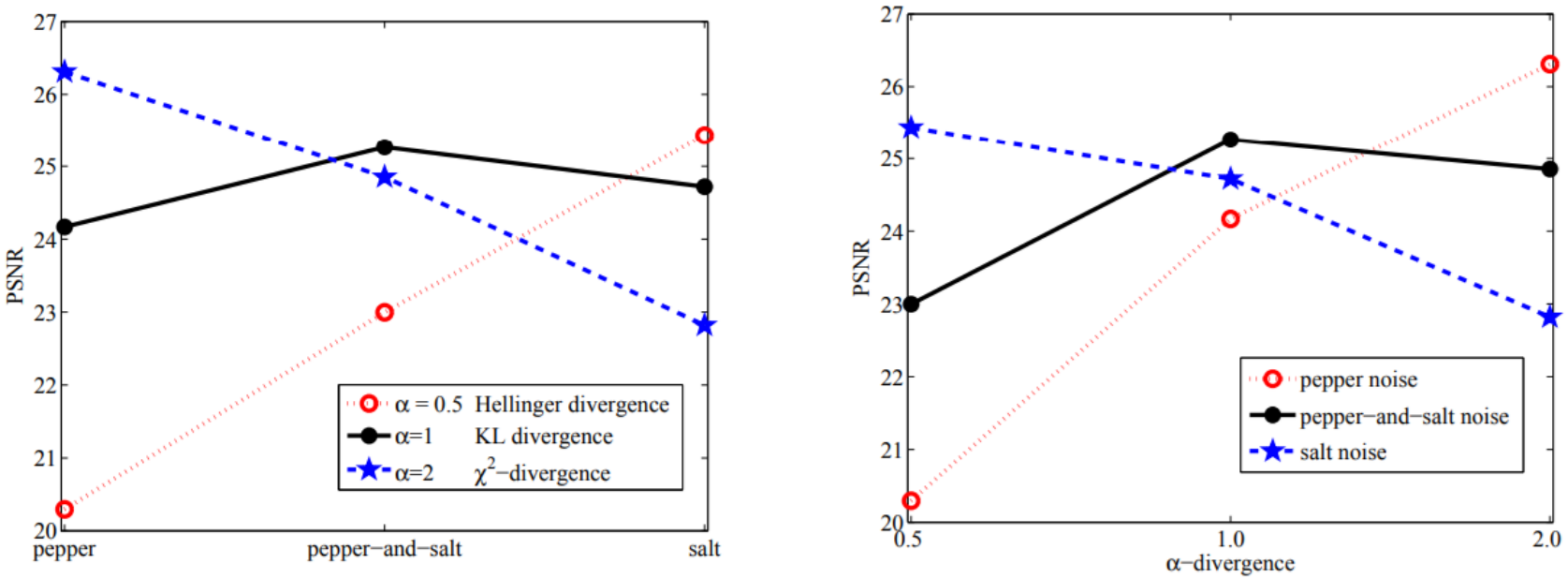


Figure: Peak noise to signal ration (PSNR) under different types of noise and divergences

Algorithms for NMF

Develop algorithms for NMF

- Non-convex optimization which is NP-hard [Vavasis, 2009]
- Heuristic local searching optimization is adopted. Alternatively update W and H

Choose the Frobenius norm as the divergence function for simplicity

$$\min_{W,H} \frac{1}{2}||X - WH||_F^2$$
$$\text{s.t. } W \geqslant 0, H \geqslant 0$$

Heuristic Local Searching

Most of the existing algorithms update W and H alternatively

- Multiply Update Rule [Lee and Seung, 2001]
- Projected Gradient [Lin, 2007]
- Active Set [Kim and Park, 2008a]
- Nesterov's optimal gradient [Guan et al., 2012a]
- Proximal Alternating Non-negative Least Square [Zhang et al., 2014]
- ...

Subproblem 1

$$\min_W \frac{1}{2}||X - WH||_F^2$$
$$\text{s.t } W \geqslant 0$$

Subproblem 2

$$\min_H \frac{1}{2}||X - WH||_F^2$$
$$\text{s.t } H \geqslant 0$$

The KKT Condition of NMF

The Lagrangian of Subproblem 1

$$\frac{1}{2}||X - WH||_F^2 + \text{tr}(G^T W)$$

where $G \in R^{m \times r}$ is the Lagrangian multiplier. The KKT condition is as follows:

$$XH^T - WHH^T = G \text{ Stationarity}$$
$$G_{ik} W_{ik} = 0 \text{ Complementary Slackness}$$
$$W \geqslant 0 \quad G \geqslant 0 \text{ Feasibility}$$

Multiply Update Rule

Combine the two equations and we have

$$(WHH^T - XH^T)_{ik} W_{ik} = 0$$

Obtain the MUR for W [Lee and Seung, 2001]:

$$W_{ik} = W_{ik} \frac{(XH^T)_{ik}}{(WHH^T)_{ik}}$$

Similarly, the MUR of H

$$H_{kj} = H_{kj} \frac{(W^T X)_{kj}}{(W^T WH)_{kj}}$$

Summary of Multiply Update Rule

Pros

- Easy to implement
- The value of the objective function decreases at the beginning

Cons

- No guarantee for local minimal
- Numerical instable when some rows or columns of X are close to zero

Projected Gradient

Gradient algorithm for the constrained problem

$$W^{k+1} = P(W^k - \alpha^k \nabla F(W^k))$$

α is the step size chosen by Armijo rule along the projection arc, such that

$$F(W^{k+1}) - F(W^k) \leqslant \sigma \nabla F(W^k)^T (W^{k+1} - W^k)$$

$\sigma \in (0, 1)$, and P is the projected operator for feasibility:

$$P(X) = \max(0, X)$$

Summary of PG

Pros

- Easy to implement
- Converge to the stationary point
- May be faster than the MUR

Cons

- Suffer from the zigzag phenomenon when approaching the local minimizer
- Line search for step-size may be time-consuming too

Active Set

- The convergence rate of gradient algorithm is only $O(\frac{1}{k})$.
- Problem: Can we obtain a much faster algorithm?
- Yes! [Kim and Park, 2008b] proposed active set algorithm for NMF

Active Set

Active set method is a general optimization method for inequality constraints

$$g_1(x) \geqslant 0, \cdots, g_k(x) \geqslant 0$$

Active Constraint

Given a point x in the feasible region, a constraint $g_i(x) \geqslant 0$ is called active at x if $g_i(x) = 0$

Reception of Active Set Algorithm

Find a feasible starting point

Algorithm 1 pseudocode for active set method

1: **repeat**
2: solve the equality problem defined by the active set (approximately)
3: compute the Lagrange multipliers of the active set
4: remove a subset of the constraints with negative Lagrange multipliers
5: search for infeasible constraints
6: **until** stopping criterion is satisfied

AS is Much Faster than MUR

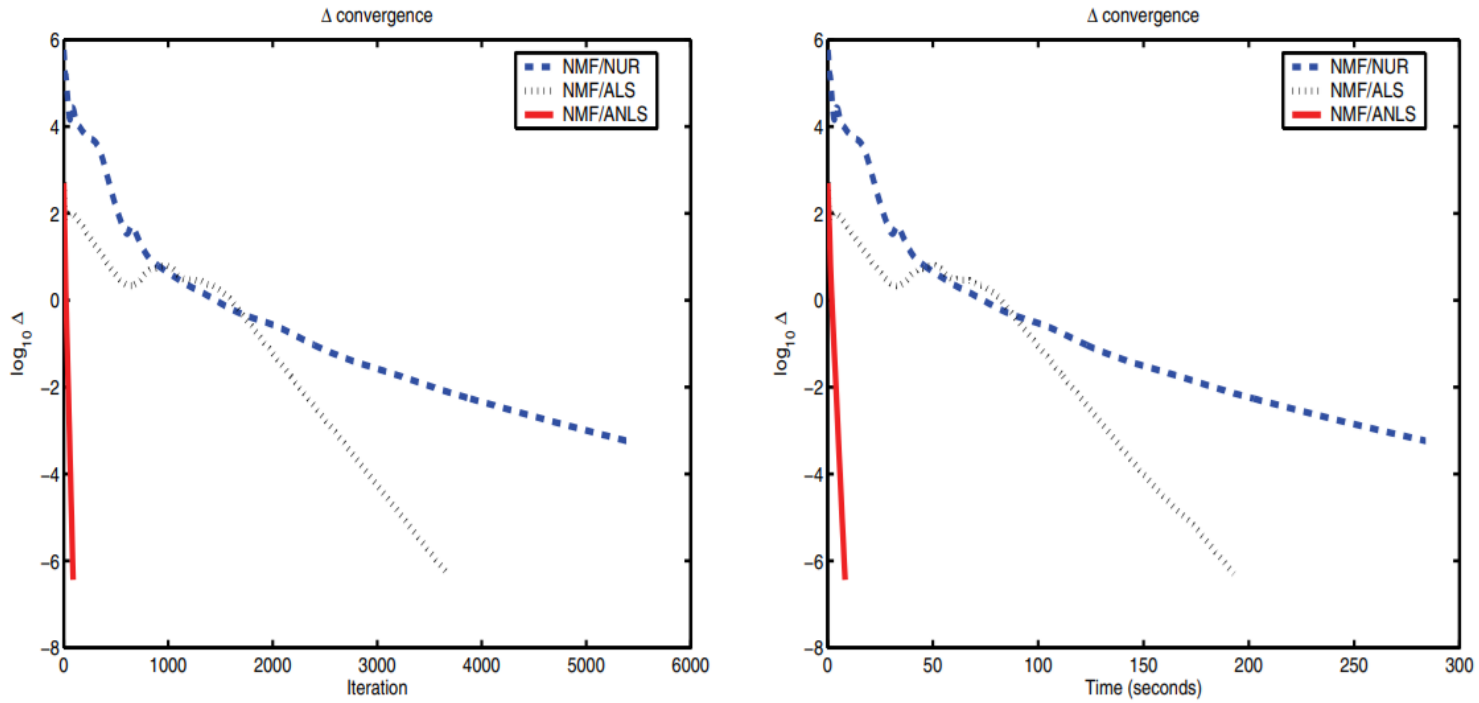


Figure: Convergence speed comparison. Blue dots line indicate MUR and red solid line indicates AS

Summary of Active Set Algorithm

Pros

- Convergence to a stationary point
- Much faster than MUR

Cons

- The algorithm is relatively complex
- Assume that each subproblem is strictly convex, which might bring about numerical instability

Nesterov's Optimal Gradient Algorithm

The convergence rate of gradient algorithm is only $O(\frac{1}{k})$.

- Problem: Can we improve the convergence rate of gradient algorithm?
- Yes! Due to [Nesterov, 2004], the optimal convergence rate of gradient algorithm is $O(\frac{1}{k^2})$.

NeNMF

- Update two sequences recursively
- H^k is the approximated sequence obtained
- Y^k is the combination of the last two steps which mimics the information of Hessian
- The step size of PG is determined by the Lipschitz constant

Algorithm 1: Optimal gradient method (OGM)

Input: W^t, H^t

Output: H^{t+1}

1: Initialize $Y_0 = H^t$, $\alpha_0 = 1$, $L = \|{W^t}^T W^t\|_2$, $k = 0$

repeat

2: Update H_k, α_{k+1} and Y_{k+1} with

$$2.1: H_k = P\left(Y_k - \frac{1}{L}\nabla_H F\left(W^t, Y_k\right)\right),$$

$$2.2: \alpha_{k+1} = \frac{1+\sqrt{4\alpha_k^2+1}}{2},$$

$$2.3: Y_{k+1} = H_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(H_k - H_{k-1}).$$

3: $k \leftarrow k+1$

until Stopping criterion (14) is satisfied

4: $H^{t+1} = H_K$

Summary of NeNMF

Pros

- Line search for the PG step size is no longer needed
- Achieves the optimal convergence rate $O(\frac{1}{k^2})$ empirically
- More numerically robust than the other methods

Cons

- No guarantee for converging to the local minimal

Comparison of Time Complexity

Algorithm	Time complexity
MUR	$O(mnr + n^2r + mr^2 + nr^2)$
PG	$O(mnr + n^2r + mr^2 + nr^2) + K \times O(tmr^2 + tnr^2)$
NeNMF	$O(mnr + n^2r + mr^2 + nr^2) + K \times (mr^2 + nr^2)$

Table: Time complexity for one iteration round of each algorithm. r is the low rank, K is the inner iteration and t is the iteration of the line search.

What is the Complexity of NMF?

- Most of the algorithms are based on local search: Given W , compute H, compute W, . . .
- NMF is convex in each factor but non-convex overall
- Few algorithms proposed in the literature come with the guarantee of optimality

Problem

What is the complexity to find the global solution of NMF?

The Hardness of NMF

NMF is NP-hard.

EXACT NMF

Given nonnegative matrix A, the output is a pair of matrices (W, H).
The output is *yes* if such a W and H exists, else it is *no*.

Sketch of Proof

The proof of NP-hardness of EXACT NMF consists of two parts [Vavasis, 2009]:

- We first show the equivalence between EXACT NMF and a problem in polyhedral combinatorics that we call INTERMEDIATE SIMPLEX.
- Then we show the NP-hardness of this problem.

How to Find the Global Solution?

- Due to the NP-hardness of NMF, it is difficult to find the global solution
- The solution is sensitive to the initialization
- Heuristic methods are adopted for searching better solution
  - Genetic Algorithm
  - Simulated Annealing
- Local minimum is good enough for most applications

Find the Global Solution of NMF Exactly

Theorem [Arora et al., 2012]

There is an $(nm)^{O(2^r r^2)}$ time exact algorithm for NMF.

Can we improve the exponential dependence on r?

Theorem [Arora et al., 2012]

An exact algorithm for NMF that runs in time $(mn)^{o(r)}$ would yield a sub-exponential time algorithm for 3-SAT.

Variants of NMF

Due to the effectiveness and the simplicity of the NMF, many variants of NMF has been proposed.

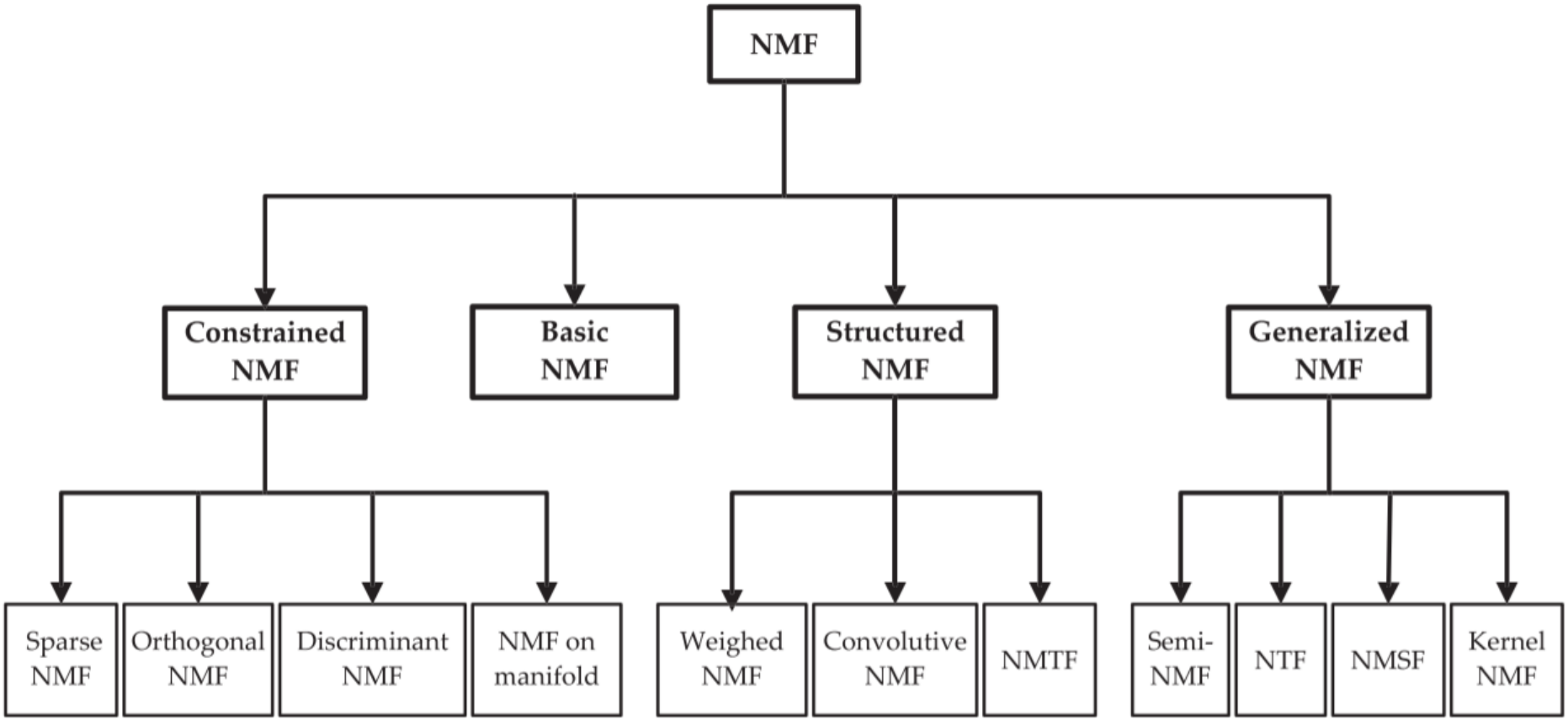


Figure: The taxonomy of NMF models [Wang and Zhang, 2012]

Constrained NMF

$$\min D(X||WH) + \lambda_1 r(W) + \lambda_2 r(H)$$

According to different formula of $r(W)$ and $r(H)$, constrained NMF algorithms are categorized into four subclass

- Sparse NMF
- Orthogonal NMF
- Discriminant NMF
- NMF on Manifold

Sparse NMF

NMF doesn't always result in part-based representations. Explicitly incorporating the sparseness to improve the found decompositions.

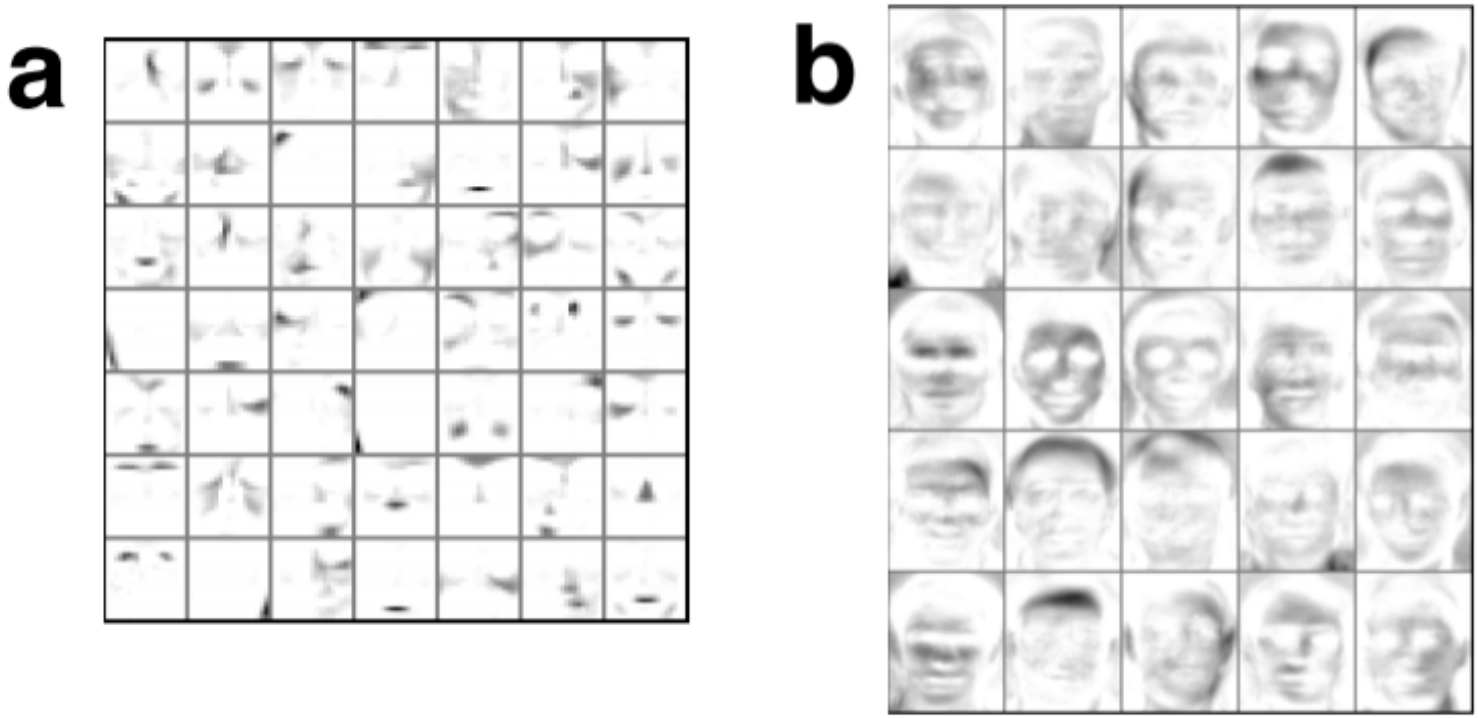


Figure: Basis faces of CBCL database (left) and ORL database (right)

Sparse NMF

Introduce l_1 penalty to regularize the sparsity of the basis matrix and coefficient matrix, respectively

$$\min_{W,H} \frac{1}{2}||X - WH||_F^2 + \lambda_1||W||_1 + \lambda_2||H||_1$$
$$\text{s.t. } W \geqslant 0, H \geqslant 0$$

where $\lambda_1 > 0, \lambda_2 > 0$ This optimization problem can be solved by the aforementioned algorithms too

Enforced Sparsity Leads Better Representation

Enforcing the sparsity of the basis matrix leads to a part-based representation.

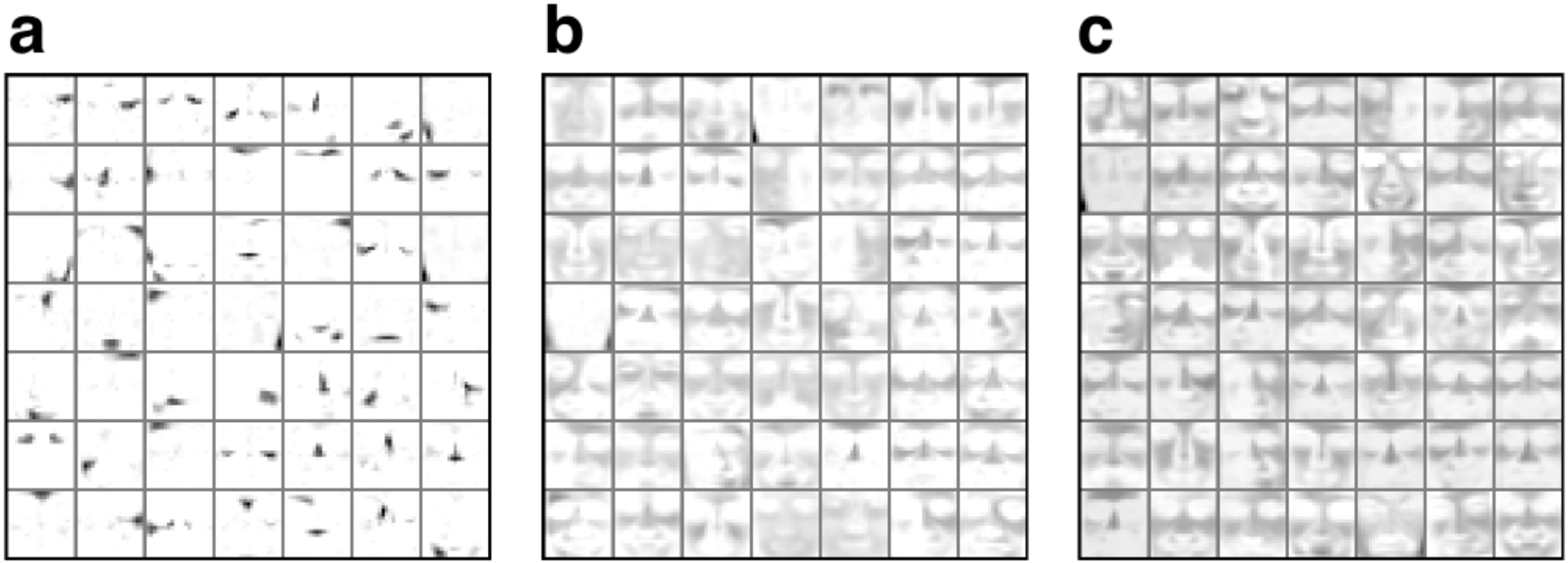


Figure: (a) Fix the sparseness of the basis matrix to 0.8 spa$(W) = 0.8$ (b) spa$(H) = 0.8$ (c) spa$(W) = 0.2$

Orthogonal NMF

- Problem: How to obtain the part-based representation?
- Add sparse constraints: sparse NMF
- Or add orthogonal constraints: orthogonal NMF

$$W^T W = I \quad HH^T = I$$

Local NMF: Li *et. al* tried to add orthogonal constraints to the objective function implicitly [Li et al., 2001]

Local NMF

Let $U = W^T W$ and $V = HH^T$

Motivations of LNMF

- Different bases should be as orthogonal as possible. $\sum_{i \neq j} u_{ij}$ should be small
- Only components giving most important information should be retained. $\sum_i v_{ii}$ should be large

$$\min_{W,H} \sum_{i,j} \left(x_{i,j} \log \frac{x_{ij}}{(WH)_{ij}} - x_{ij} + (WH)_{ij} \right) + \lambda_1 \sum_{i,j} u_{ij} - \lambda_2 \sum_i v_{ii}$$

$$\text{s.t } W \geqslant 0, H \geqslant 0$$

More about Orthogonal NMF

Benefits of Orthogonality

- Reduce the redundancy between different bases
- In the condition of non-negativity, orthogonality will result in sparseness
- Orthogonal NMF is preferable in clustering tasks

It is also possible to add the orthogonal constraints explicitly [Ding et al., 2006, Choi, 2008]

Incorporate Discriminant Information to NMF

NMF are typically considered as unsupervised learning.

Problem

How do we incorporate discriminant information to get supervised alternatives?

Discriminant variants of NMF has been proposed
[Jia and Turk, 2004, Zafeiriou et al., 2006]

Between/Within Scatter Matrices

To use the information of the classes, the within scatter of the encoding matrix

$$S_w = \frac{1}{C}\sum_{i=1}^{C}\frac{1}{n_i}\sum_{j=1}^{n_i}(h_j - u_i)(h_j - u_i)^T$$

where C is the number of classes, n_i is the number of instances in class i and $u_i = \frac{1}{n_i}\sum_{j=1}^{n_i} h_j$
The between scatter matrix

$$S_b = \frac{1}{C(C-1)}\sum_{i=1}^{C}\sum_{j=1}^{C}(u_i - u_j)(u_i - u_j)^T$$

Fisher NMF

Incorporate the between/within scatter matrices to the NMF model [Jia and Turk, 2004]

$$\min_{W,H} \sum_{i,j} \left(x_{i,j} \log \frac{x_{ij}}{(WH)_{ij}} - x_{ij} + (WH)_{ij} \right) + \lambda_1 S_w - \lambda_2 S_b$$

$$\text{s.t } W \geqslant 0, H \geqslant 0$$

Fisher NMF achieved a better performance in human face recognition than NMF and LNMF

NMF on Manifold

- The real-world data are often sampled from a nonlinear low-dimensional manifold
- If the intrinsic geometrical structure is identified and preserved, the performance can be significantly enhanced
- Solution: Preserve the local typology

Graph Regularized NMF: model the manifold structure by constructing a nearest neighborhood graph of data points [Cai et al., 2010]

Graph Regularized NMF

Motivations of GRNMF

- NMF fails to discover the intrinsic geometrical structure
- Assumption: if two data points are close in the data distribution, then the representation of the two points are also close to each other

$$\min_{W,H} \frac{1}{2}||X - WH||_F^2 + \lambda r(H)$$
$$\text{s.t } W \geqslant 0, H \geqslant 0$$

$r(H)$ is the graph regularizer

How to Construct the Graph

Use a graph with weight matrix G to capture the information of the intrinsic geometrical structure

- G_{ij} indicates the weight of the edge between data points i and j
- If two data points are close in the data distribution, then the weight of the edge should be large

Construct a k-nearest neighbor graph to model the manifold

- 0-1 Weighting If x_j is the k-nearest neighbor, then $G_{ij} = 1$
- Heating Kernel Weighting if nodes j and i are connected, put

$$G_{ij} = \exp -\frac{||x_i - x_j||^2}{\sigma}$$

The Graph Regularizer

The graph regularizer

$$r(H) = \frac{1}{2}\sum_{i,j=1}^{n} ||h_j - h_i||^2 G_{ij} = \sum_{i=1}^{N} h_j^T h_j G_{jj} - \sum_{i,j=1}^{n} h_i^T h_j G_{ij}$$
$$= \mathrm{tr}(HDH^T) - \mathrm{tr}(HGH^T) = \mathrm{tr}(HLH^T)$$

where $L = D - G$ is the graph Laplacian

$$\min_{W,H} \frac{1}{2}||X - WH||_F^2 + \lambda \mathrm{tr}(HLH^T)$$
$$\text{s.t } W \geqslant 0, H \geqslant 0$$

GRNMF has significantly higher performance than NMF in clustering task

Structured NMF

$$X \approx F(WH)$$

Structured NMF modifies the regular factorization directly. It contains three subclasses

- Weighted NMF
- Convolutive NMF
- Non-negative Matrix Trifactorization

Non-negative Matrix Trifactorization

Potential crisis of constrained NMF

Additional constraints help obtaining

- Sparse NMF
- Orthogonal NMF

But those model could be two restrictive and may lead a poor low-rank representation!

Nonsmooth NMF (nsNMF)

Motivation of nsNMF

To obtain the interpretable representation and a good approximation simultaneously, nsNMF extended the NMF model to trifactorization [Pascual-Montano et al., 2006]

$$\min_{W,H} \frac{1}{2}||X - WSH||_F^2$$
$$\text{s.t } W \geqslant 0, H \geqslant 0$$

where S is a positive symmetric matrix $S \in R^{r \times r}$ referred as smoothing matrix

The Smoothing Matrix

The parameter $1 < \theta <= 0$ controls the smoothness of the model

$$S = (1 - \theta)I + \frac{\theta}{r}\mathbf{1}\mathbf{1}^{\mathsf{T}}$$

Let $Y = XS$

- If $\theta = 0$, $Y = X$ there is no smoothing
- if $\theta \to 1$, Y tends to be the average of the elements of X
- Strong smoothing in S will force strong sparseness in both the basis and the encoding vectors in order to maintain faithfulness

Generalized NMF

Generalized NMF is the extensions of NMF in a broad sense: change the data types, alter the fraction pattern, etc.

- Semi-NMF and Convex NMF
- Non-negative Tensor Factorization
- Kernel NMF

Semi-NMF

In some cases, the data matrix are real-valued. NMF is not directly applicable. Matrix factorization method suitable for real-valued data is in favor [Ding et al., 2008]

$$\min_{W,H} \frac{1}{2}||X - WH||_F^2$$
$$\text{s.t } H \geqslant 0$$

where W is real-valued, but H is still non-negative. MUR algorithm can be derived by the same technique as NMF

Why Semi-NMF?

- Applicable for real-valued data
- The non-negative coefficient matrix retains the interpretability and the nature of sparsity

Convex NMF

Motivation of Convex NMF

In NMF and Semi-NMF, there are no constraints on the basis matrix. Again for reasons of interpretability, we may restrict the basis matrix to convex combination of the columns of X.

$$W = XF$$

$$\min_{W,H} \frac{1}{2}||X - XFH||_F^2$$
$$\text{s.t } F \geqslant 0, H \geqslant 0$$

Why Convex NMF?

- Interpret the columns of the basis matrix as weighted sum of certain data points
- Capture a notion of centroids
- Applicable for real-valued data
- Both factors tend to be very sparse

Semi-NMF vs. Convex NMF

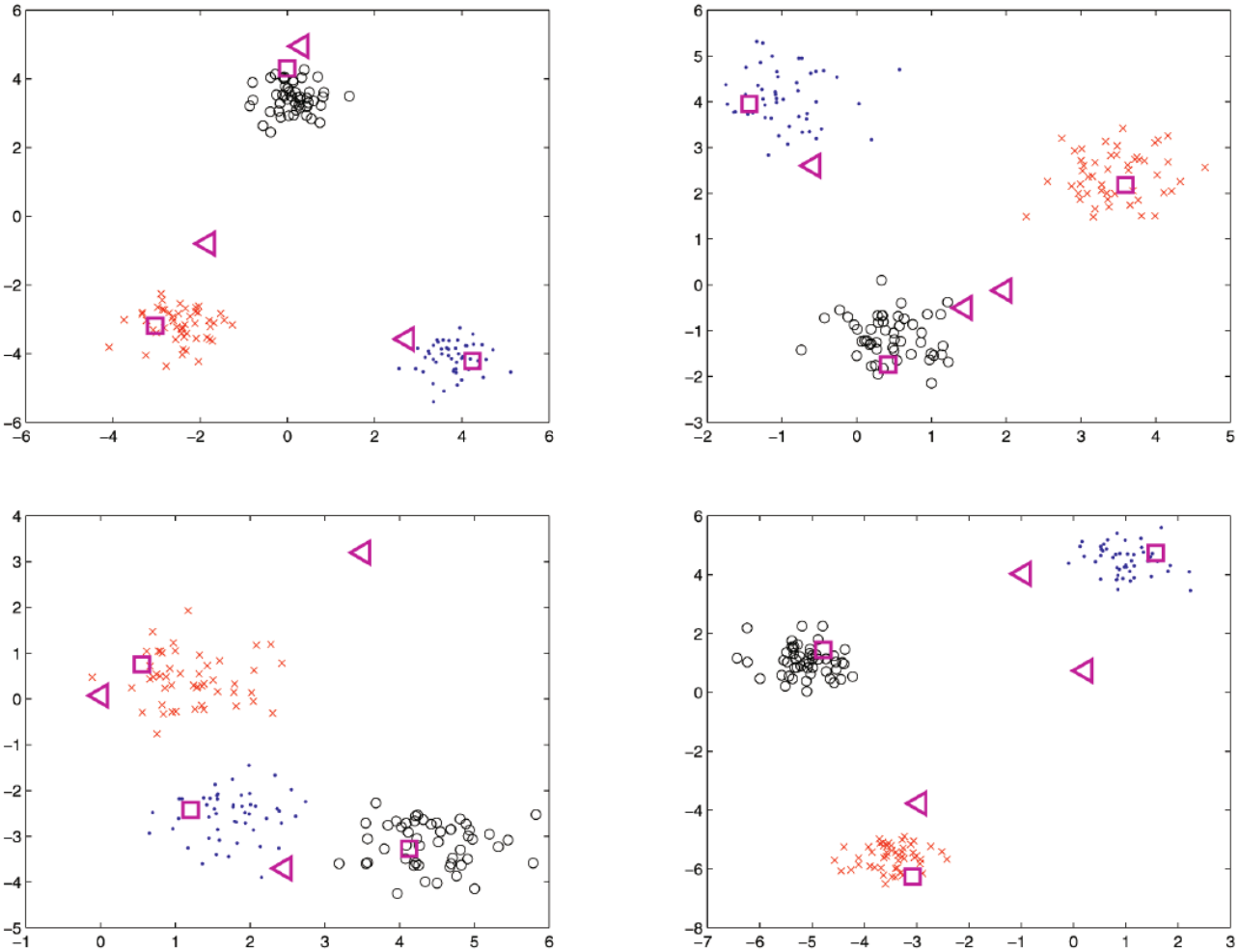


Figure: Four random datasets, each with three clusters. □ for convex NMF and ◁ for semi NMF

Kernel NMF

- Aforementioned NMF and its variants are linear, which are unable to extract non-linear structure
- Apply kernel based methods by mapping input data into implicit feature space

Formulation of Kernel NMF

Given a nonlinear mapping $\phi : R^M \to R$, the original data matrix is transformed into $X \to Y = \phi(X)$. Kernel NMF seeks to find factor matrix $Z = [\phi(w_1), \ldots, \phi(w_r)]$ and H

$$\begin{aligned} D_F(Y||ZH) &= \frac{1}{2}||Y - ZH||_F^2 \\ &= \frac{1}{2}\text{tr}(Y^T Y) - \text{tr}(Y^T ZH) - \frac{1}{2}\text{tr}(H^T Z^T ZH) \end{aligned}$$

Using kernel function $k(x, y) = \langle \phi(x), \phi(y) \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the inner product and kernel matrices

$$K_{ij}^{xx} = \langle \phi(x_i), \phi(x_j) \rangle, K_{ij}^{ww} = \langle \phi(w_i), \phi(w_j) \rangle, K_{ij}^{xw} = \langle \phi(x_i), \phi(w_j) \rangle$$

Formulation of Kernel NMF, Cont'd

The objective function can be rewritten as

$$
\begin{aligned}
D_F(Y\|ZH) &= \frac{1}{2}\|Y - ZH\|_F^2 \\
&= \frac{1}{2}\mathrm{tr}(K^{xx}) - \mathrm{tr}(K^{xw}H) - \frac{1}{2}\mathrm{tr}(H^T K^{ww} H)
\end{aligned}
$$

- [Buciu et al., 2008] proposed the above kernel NMF model in polynomial feature space
- By resorting to the PG method, it is generalized to any kernel function [Liang et al., 2010]

[0]Source: hitec-dubai

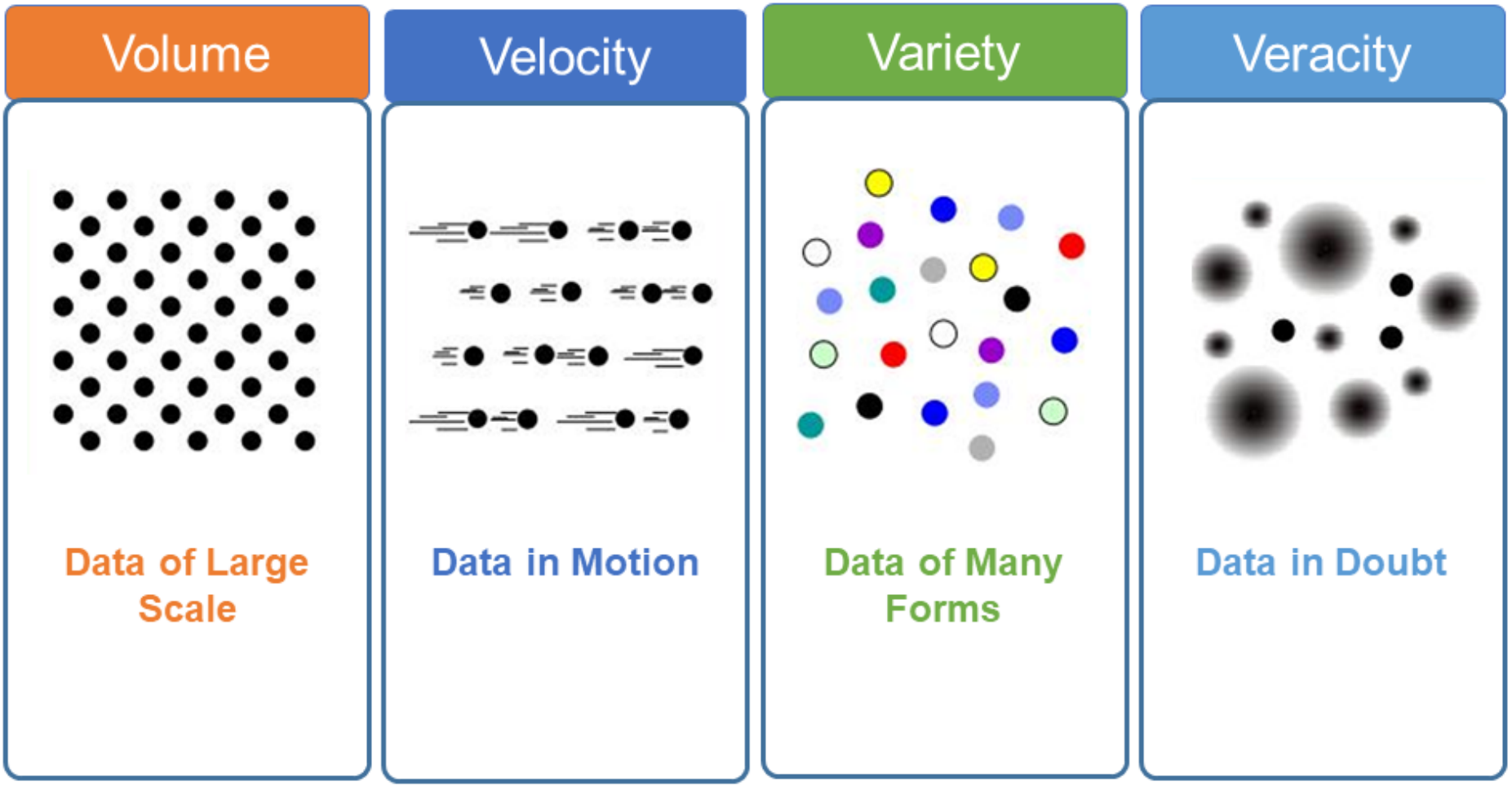


Figure: Characteristics of big data

NMF towards Big Data

The characteristics of big data bring new challenges. Many researchers devoted their efforts to address those challenges partially

- Volume: analysis large-scale data efficiently
  - Scalable NMF [Benson et al., 2014]

NMF towards Big Data

The characteristics of big data bring new challenges. Many researchers devoted their efforts to address those challenges partially

- Volume: analysis large-scale data efficiently
- Velocity: analysis data in an online fashion
  - Online Robust NMF [Guan et al., 2012b]

NMF towards Big Data

The characteristics of big data bring new challenges. Many researchers devoted their efforts to address those challenges partially

- Volume: analysis large-scale data efficiently
- Velocity: analysis data in an online fashion
- Variety: data with complex structure requires more representative model
  - Deep semi-NMF [Trigeorgis et al., 2016]

NMF towards Big Data

The characteristics of big data bring new challenges. Many researchers devoted their efforts to address those challenges partially

- Volume: analysis large-scale data efficiently
- Velocity: analysis data in an online fashion
- Variety: data with complex structure requires more representative model
- Veracity: the uncertainty of data is concerned
  - GLAD [Saddiki et al., 2014]

Scalable NMF

Tall and Skinny Matrices

In many situations, the data matrix $X \in R^{m \times n}$ has much more instances than features, i.e., $n \gg m$. Such data is referred as tall and skinny matrices. Such data matrices can be very large and thus cannot be handled by a single machine

Scalable NMF

Tall and Skinny Matrices

In many situations, the data matrix $X \in R^{m \times n}$ has much more instances than features, i.e., $n \gg m$. Such data is referred as tall and skinny matrices. Such data matrices can be very large and thus cannot be handled by a single machine

Motivation of Scalable NMF

How to apply NMF to such tall and skinny matrix efficiently?

Conic Hull

The set of conical combination for a given set S is called conical hull

$$\text{coni}(S) = \{\sum_{i=1}^{k} \alpha_i x_i | x_i \in S, 0 \leqslant \alpha_i\}$$

Extreme Ray

A ray r is an extreme ray of a cone P if there do not exist $r_1, r_2 \in P$ and a scaler μ (with $r_1 \neq \lambda r_2$ for any $\lambda > 0$ and $0 < \mu < 1$) such that $r = \mu r_1 + (1 - \mu) r_2$

Separable Assumption

- Finding W and H such that the residual is minimized is NP-hard
- To avoid alternating optimizing between W and H, scalable NMF made the separable assumption

Separable Condition

$$X = X(:, \mathcal{K})H$$

where $\mathcal{K}$ is an index set of size r

Near Separable Condition

$$X = X(:, \mathcal{K})H + N$$

where N is the small noise entries

Algorithms for Near Separable NMF

- The near-separability indicates that all columns of X lives in the conical hull of the extreme columns
- The algorithm for near separable NMF are typically described by a two step approach

Algorithm

1. Determine the extreme columns, and let $W = X(:, \mathcal{K})$
2. Solve $H = \arg\min_Y ||X - WY||_F^2$

Reduce the Problem Scale

Theorem 1

Consider a cone C and non-singular matrix M. x is an extreme ray of C if and only if Mx is an extreme ray of MC

Consider reducing the problem scale by orthogonal transformation

Orthogonal Transformation

Let $X = Q\tilde{R}$ and $X = U\tilde{\Sigma}V^T$ be full QR factorization and SVD, then

$$Q^T X = \begin{bmatrix} R \\ 0 \end{bmatrix} \quad U^T X = \begin{bmatrix} \Sigma V^T \\ 0 \end{bmatrix}$$

Orthogonal Transformation

If the separable assumption holds, we immediately have the separated representation

$$R = R(:, \mathcal{K})H \quad \Sigma V^T = \Sigma V^T(:, \mathcal{K})H$$

- The problem size has been significantly reduced
- Apply algorithm on R to find the extreme column
- Orthogonal transformation preserves the geometry
- QR and SVD can be computed by TSQR of $O(mn^2)$ flops

Computing H

Computing H involves a set of NNLS problems

$$H(:,i) = \arg\min_{y \in R_+^r} ||X(:,\mathcal{K})y - X(:,i)||_2^2$$

Reduce the Problem Scale

H can be computed efficiently. Let $X = Q\tilde{R}$

$$\begin{aligned} ||X(:,\mathcal{K})y - X(:,i)||_2^2 &= ||Q^T(X(:,\mathcal{K})y - X(:,i))||_2^2 \\ &= ||R(:,\mathcal{K})y - R(:,i)||_2^2 \end{aligned}$$

Online NMF

Streaming Data

In the big data era, data is often generated continuously. Such data should be processed incrementally. NMF and most of its variants requires full data

[Guan et al., 2012b] proposed online random stochastic NMF (online RSA-NMF) to address this challenge

Online RSA-NMF

Problem Formulation

Given n samples $\{x_1, \ldots, x_n\} \in R_+^m$ distributed in the probabilistic space $P \in R_+^m$, NMF learns a subspace $Q \subset P$ spanned by r bases $\{w_1, \ldots, w_n\} \in R_+^m$

$$\min_{W \in R_+^{m \times r}} f_n(W) = \frac{1}{n} \sum_{i=1}^{n} l(x_i, W)$$

where

$$l(x_i, W) = \min_{h_i \in R_+^r} \frac{1}{2} ||x_i - W h_i||_2^2$$

Minimizing the Expected Cost

Typically, one is usually not interested in minimizing the empirical cost $f_n(W)$, but instead in minimizing the expected cost

$$\min_{W \in R_+^{m \times r}} f(W) = E_{x \in P}(l(x, W)$$

where $E_{x \in P}$ denotes the expectation over P

Update W Online

On the arrival of sample x^t, we obtain the corresponding coefficient h^t by

$$\min_{h^t \in R_+^r} \frac{1}{2}||x^t - W^{t-1}h^t||_2^2$$

Followed by updating W^t

$$W^t = \arg \min_{W \in R_+^{m \times r}} E_{x \in P_t}(\frac{1}{2}||x - Wh||_2^2)$$

where P_t is the probabilistic space spanned by the arrived samples

Deep Semi-NMF

Limitation of NMF

NMF and most of its variants are bilinear models. Thus the model capacity is relative low

- Due to the success of deep model, [Trigeorgis et al., 2016] proposed deep semi-NMF model
- Mapping between this new representation and our original data matrix contains rather complex hierarchical information

Model Formulation

Recall semi-NMF

$$X \approx W^{\pm} H^{+}$$

Deep Semi-NMF factorizes a given data matrix into $m+1$ factors

$$X \approx W_1^{\pm} W_2^{\pm} \cdots W_m^{\pm} H^{+}$$

The representation of data can be given by the following factorization

$$\begin{aligned} H_{m-1}^{+} &\approx W_m^{\pm} H_m^{+} \\ &\vdots \\ H_1^{+} &\approx W_2^{\pm} \cdots W_m^{\pm} H_m^{+} \end{aligned}$$

Model Formulation, Cont'd

Introduce non-linearity

$$H_i \approx g(W_{i+1} H_{i+1})$$

where g is a nonlinear function such as tanh. It turns the objective function to

$$C = \frac{1}{2} \|X - W_1 g(W_2 g(\cdots g(W_m H_m)))\|_F^2$$

One can take the derivative of each layer and update with SGD

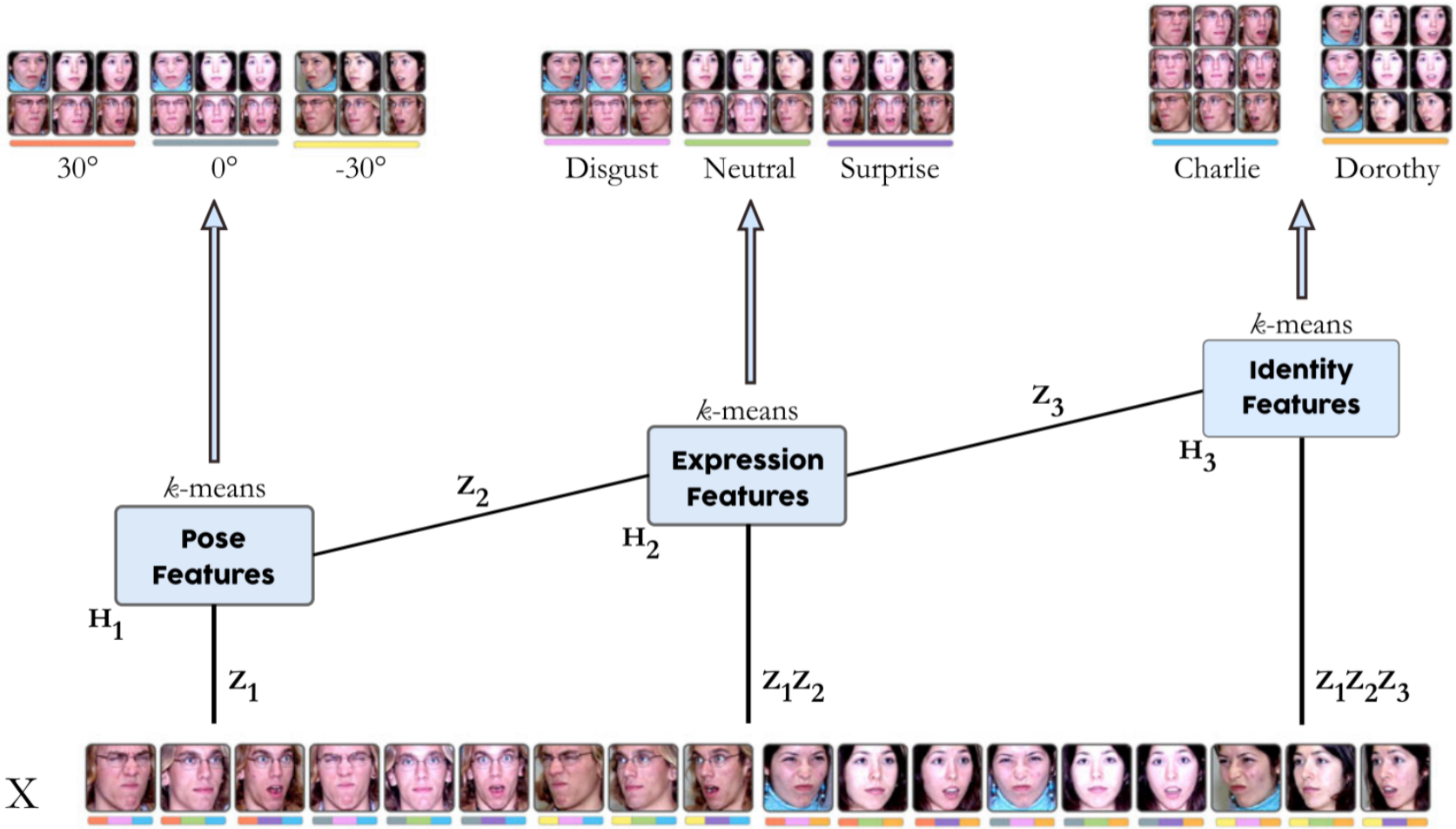


Figure: A deep semi-NMF model learns a hierarchical structure of features

Bayesian Matrix Factorization

- Veracity: the uncertainty of data is concerned
- Bayesian framework is a powerful tool to address the uncertainty of data

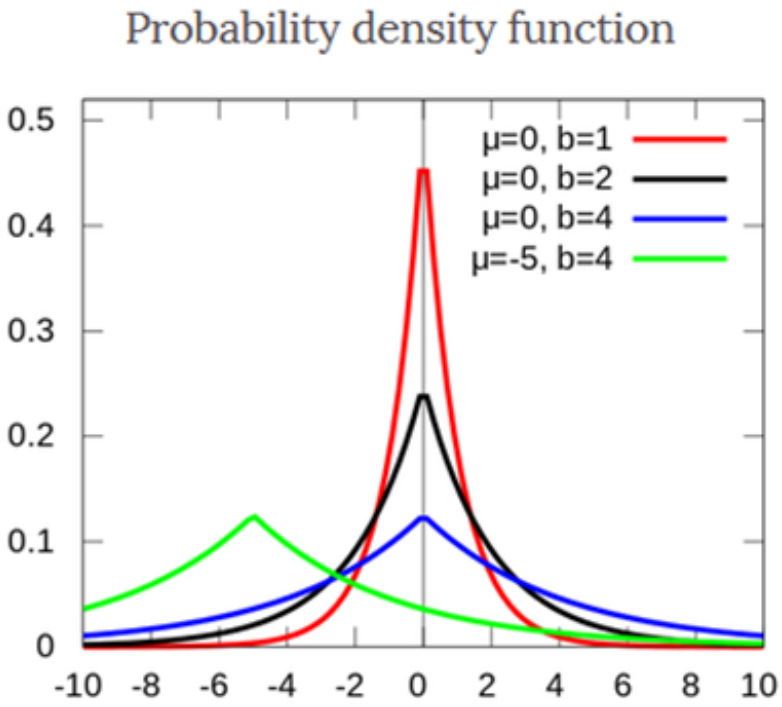
Why Bayesian Matrix Factorization

- Address the uncertainty of data naturally
- Incorporate the prior knowledge by Bayesian priors naturally

$$X = WH + \epsilon$$

Laplace Distribution

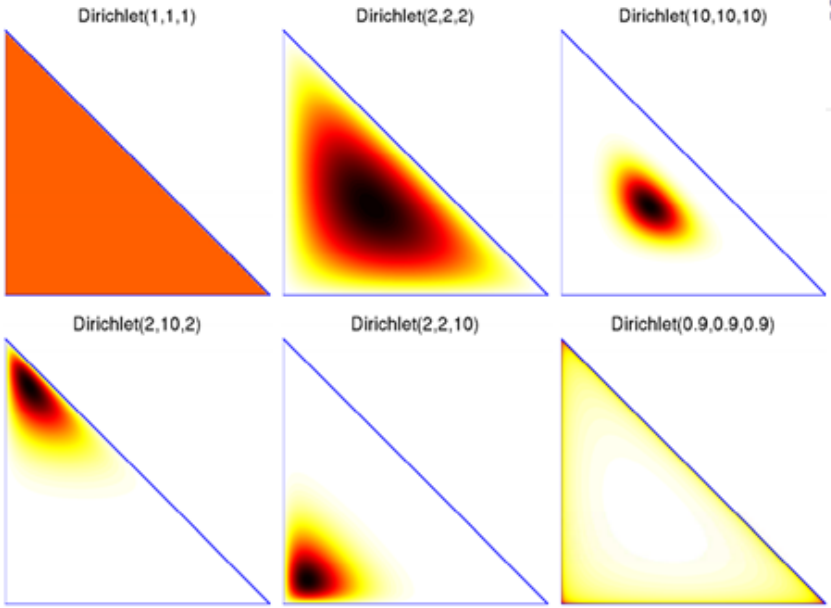
- Laplace distribution offers l_1 penalty in likelihood function



Parameters	μ location (real)
	$b > 0$ scale (real)
Support	$x \in (-\infty; +\infty)$
PDF	$\frac{1}{2b} \exp\left(-\frac{\lvert x-\mu \rvert}{b}\right)$

Dirichlet Distribution

- Dirichlet distribution is a distribution over the K-dim probability simplex.
- Examples of Dirichlet distributions over which can be plotted in 2D since : $p_3 = 1 - p_1 - p_2$



Support

$$x_1, \ldots, x_K \text{ where } x_i \in (0,1) \text{ and } \sum_{i=1}^{K} x_i = 1$$

PDF

$$\frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

$$\text{where } \mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}$$

$$\text{where } \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$$

GLAD

- GLAD is a mixed-membership model for heterogeneous tumor subtype classification [Saddiki et al., 2014]
- GLAD is a matrix factorization model involving three distribution, i.e Gaussian, Laplace and Dirichlet distribution

$$X = WH + \epsilon$$

  - W follows Laplace prior for sparsity
  - H follows Dirichlet distribution for interpretability
  - ϵ follows Gaussian distribution to model noise

Variational Inference for GLAD

- Calculating the posterior distribution directly is intractable
- Using proposal distribution q to approximate posterior distribution

$$\min KL_{q \in P}(q||p) = -\int q(W, H) \ln \frac{p(W, H|X; \alpha, \lambda, \sigma^2)}{q(W, H; \phi)} dWdH$$

$$\max \mathcal{L} = E_q[\ln p(X, W, H)] - E_q[\ln p(W, H)]$$

- If $\mathcal{L}$ can be computed analytically, VI turns the inference to an optimization problem
- GLAD further approximates $\mathcal{L}$ by Laplace approximation
  - Computational expensive, unrealistic for real-world data

Multi-view Data

Mutli-view Data

The multi-view data set collection contains sets of different features extracted from the same samples/individuals.

Such data of different views are often complementary to each other

Multi-view Data

Mutli-view Data

The multi-view data set collection contains sets of different features extracted from the same samples/individuals.

Such data of different views are often complementary to each other

Problem

How do we conduct an integrative analysis for such multi-view data?

Joint NMF

- Goal: Identify multi-dimensional modules across multiple types of genomics data

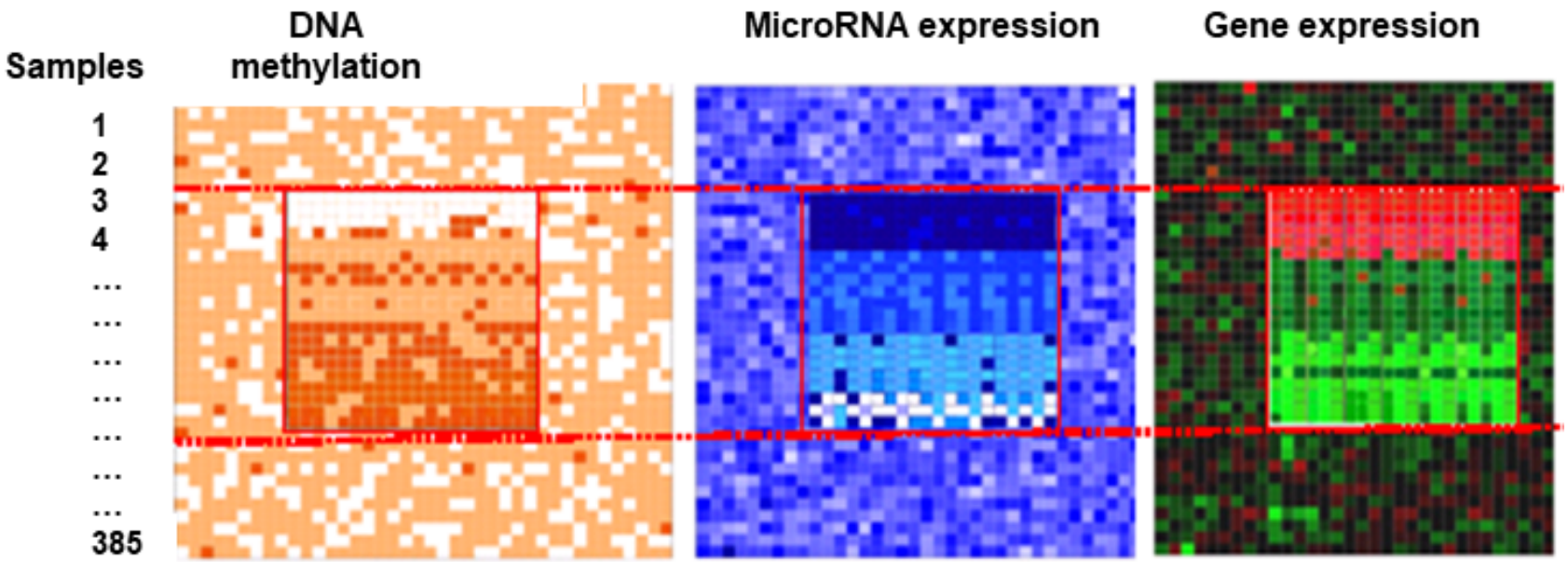
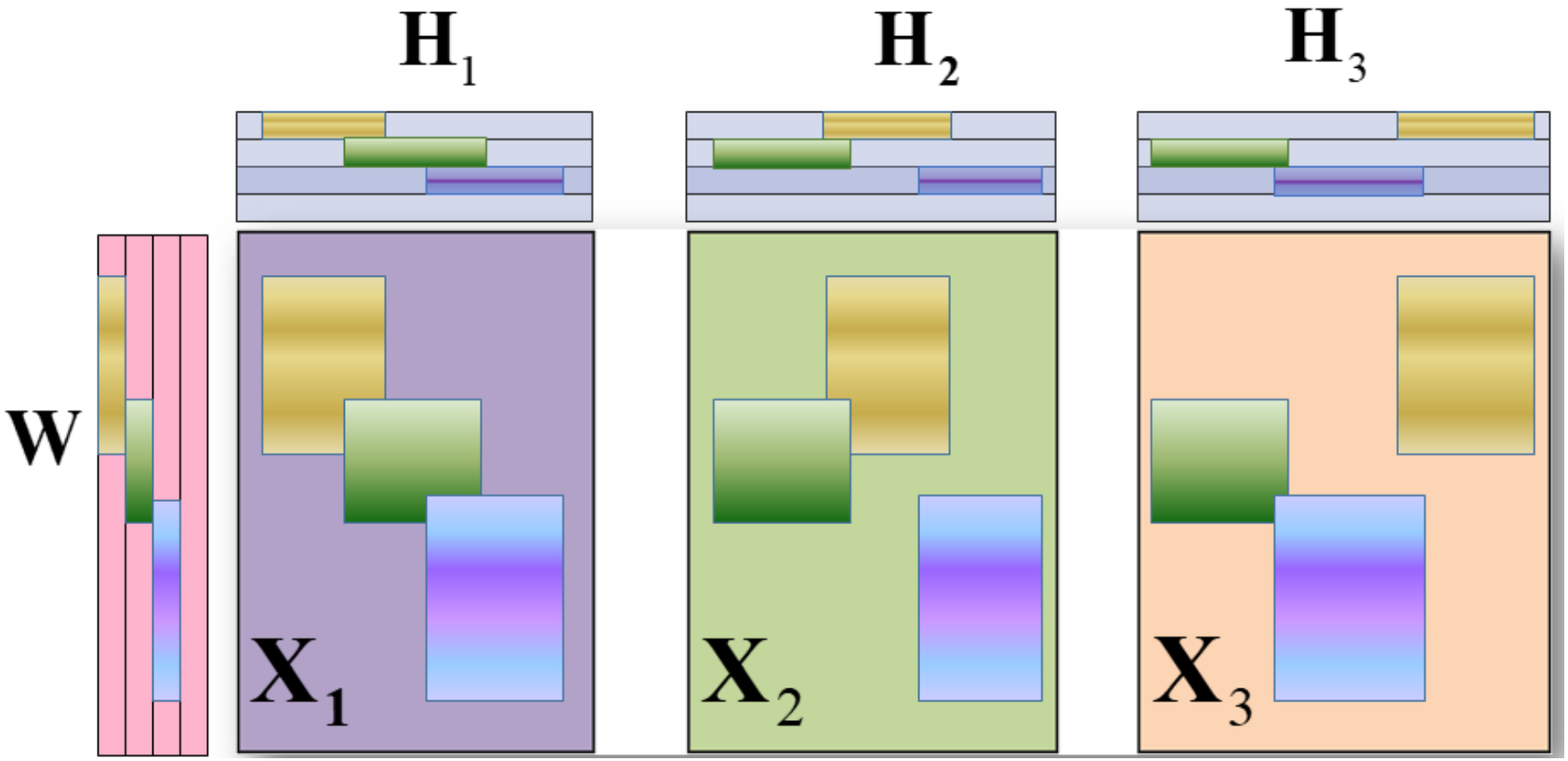


Figure: A multi-dimensional module is a set of DNA methylation markers, miRNAs and genes that show correlated profiles across a subset of samples

Joint NMF, Cont'd



$$\min_{W,H_1,H_2,H_3 \geqslant 0} \sum_{i=1}^{3} ||X_i - WH_i||_F^2$$

Integrative NMF

Joint NMF is a powerful tool for integrative analysis of multi-view data. Data from different sources share the same basis matrix.

Motivation of Integrative NMF

Data from different sources share similar patterns, but may also demonstrate view/source specific effect.
How do we capture the view/source specific effects?

[Yang and Michailidis, 2015] proposed Integrative NMF (iNMF) to address this challenge

Formulation of iNMF

$$X_i \approx WH_i + V_i H_i$$

where W and V_k are the common and specific basis matrices, receptively

$$\min \sum_{i=1} \|X_i - (W + V_i)H_i\|_F^2 + \lambda \sum_{i=1} \|V_i H_i\|_F^2$$
$$\text{s.t. } W \geqslant 0, H_i \geqslant 0, V_i \geqslant 0$$

- Penalize the Frobenius norm of the heterogeneous effects $V_i H_i$
- WH_i can always be expressed in terms of $V_i H_i$, but not vice-visa

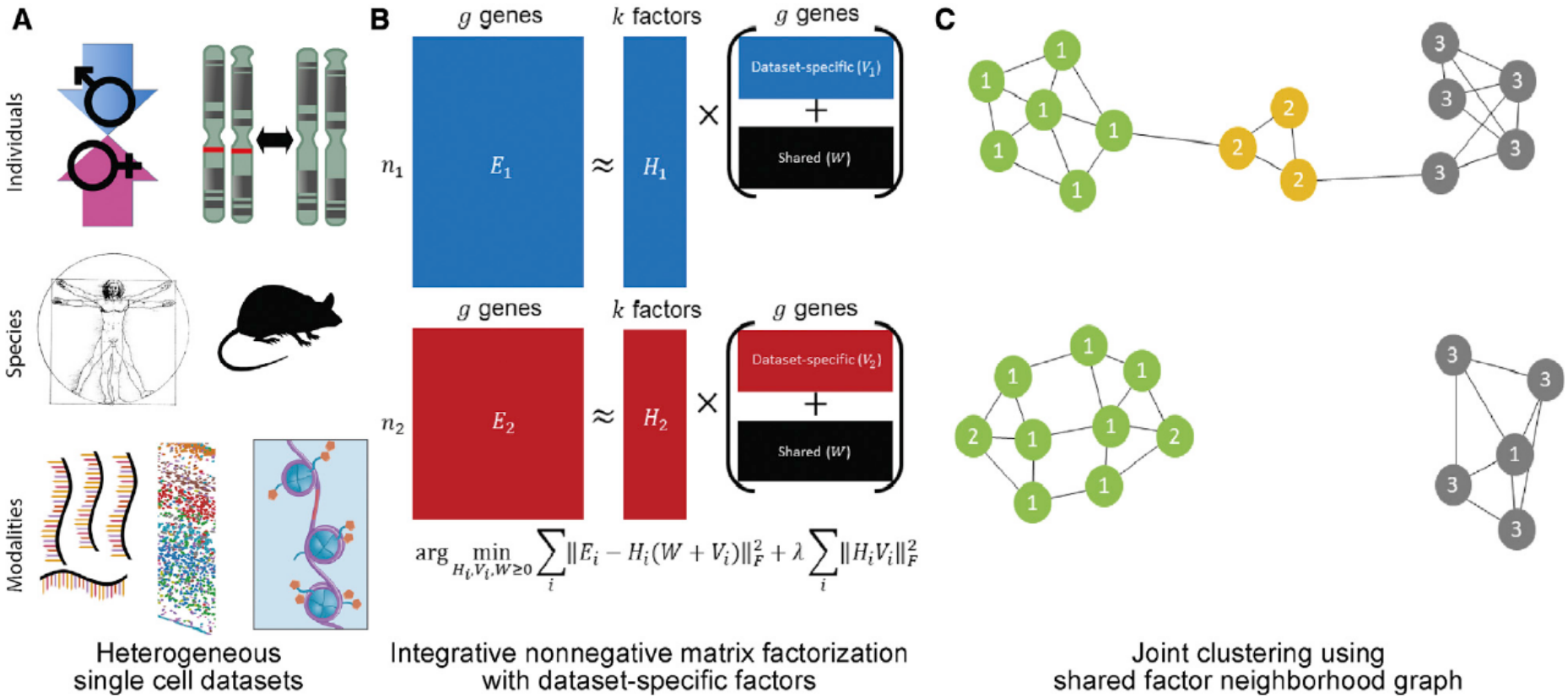


Figure: Discover the common and specific pattern in singe-cell multi-view data [Welch et al., 2019]

Motivation of CSMF

Integration and deferential analysis are two common paradigms for analyzing such data. Integration methods may ignore the differential part, and vice versa.

[Zhang and Zhang, 2019b] proposed CSMF to simultaneously reveal Common and Specific patterns via Matrix Factorization

Formulation of CSMF

Given two non-negative matrices X_1 and X_2, low ranks k_c, k_{s1}, k_{s2}

$$X_1 \approx W_c H_{c1} + W_{s1} H_{s1}$$
$$X_2 \approx W_c H_{c2} + W_{s2} H_{s2}$$

Under the Frobenius norm

$$\min_{W_c,\ldots,H_sc\geqslant 0} \|X_1 - (W_c H_{c1} + W_{s1} H_{s1})\|_F^2 + \|X_2 - (W_c H_{c2} + W_{s2} H_{s2})\|_F^2$$

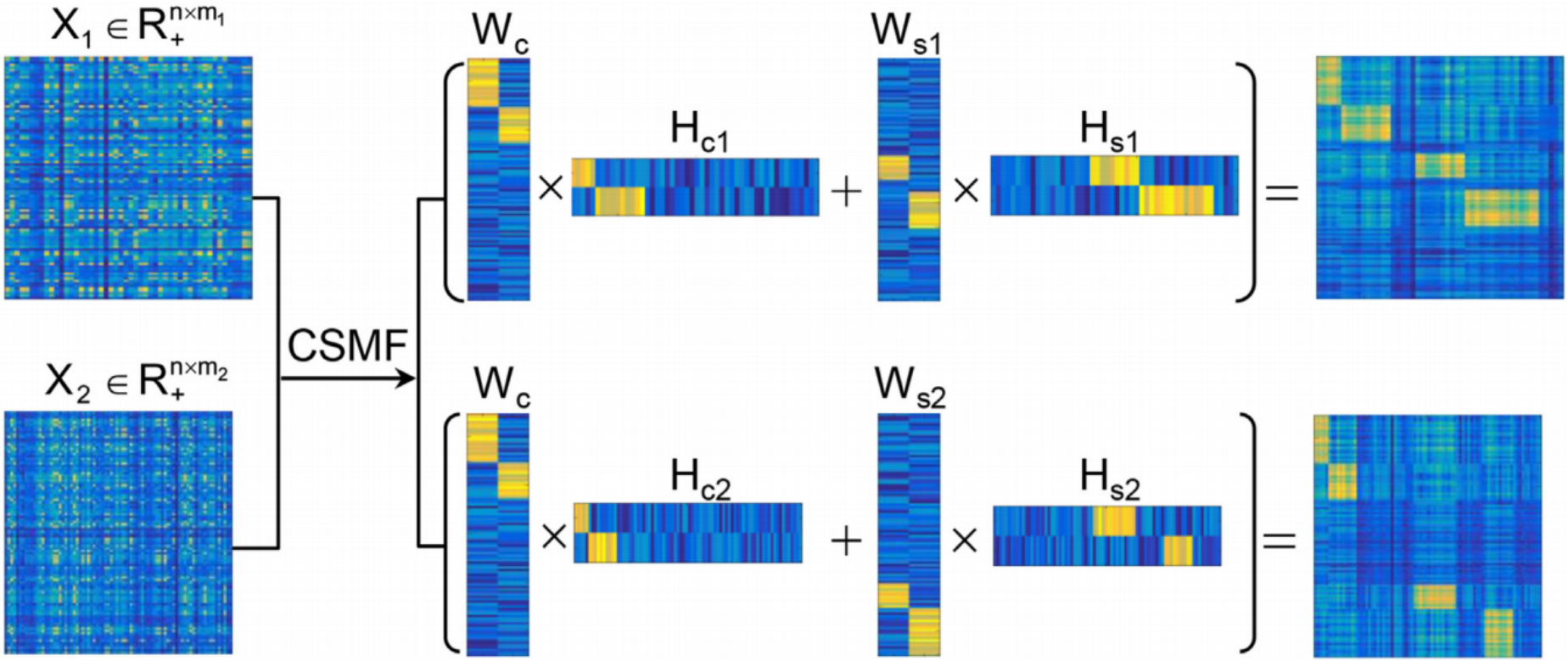


Figure: Illustration of CSMF

Unified Joint Matrix Factorization Framework

Inspired by GRNMF, [Zhang et al., 2011] introduced network regularizers for multi-view omics data

Problem

How to introduce the sparse constraints and network regularizers into one unified joint matrix factorization framework?

- [Zhang and Zhang, 2019a] proposed Joint Matrix Factorization (JMF) for data integration
- It also provided a systematic algorithmic comparison

Formulation of JMF

Introduce sparse constraints and graph regularizers

$$
\begin{aligned}
\min \quad & \sum_{l=1,2} \|X_l - WH_l\|_F^2 - \lambda_1 \sum_{l=1,2} \mathrm{tr}(H_l \Theta_l H_l^T) \\
& - \lambda_2 \mathrm{tr}(H_1 R_{12} H_2^T) + \gamma_1 \|W\|_F^2 \\
& + \gamma_2 \left(\sum_j \|h_j^1\|_1^2 + \sum_{j'} \|h_{j'}^2\|_1^2 \right) \\
\text{s.t.} \quad & W \geqslant 0, H_l \geqslant 0,
\end{aligned}
$$

- $\mathrm{tr}(H_l \Theta_l H_l^T)$ is the graph regularizer on view l
- $\mathrm{tr}(H_1 R_{12} H_2^T)$ is the graph regularizers between view 1 and 2

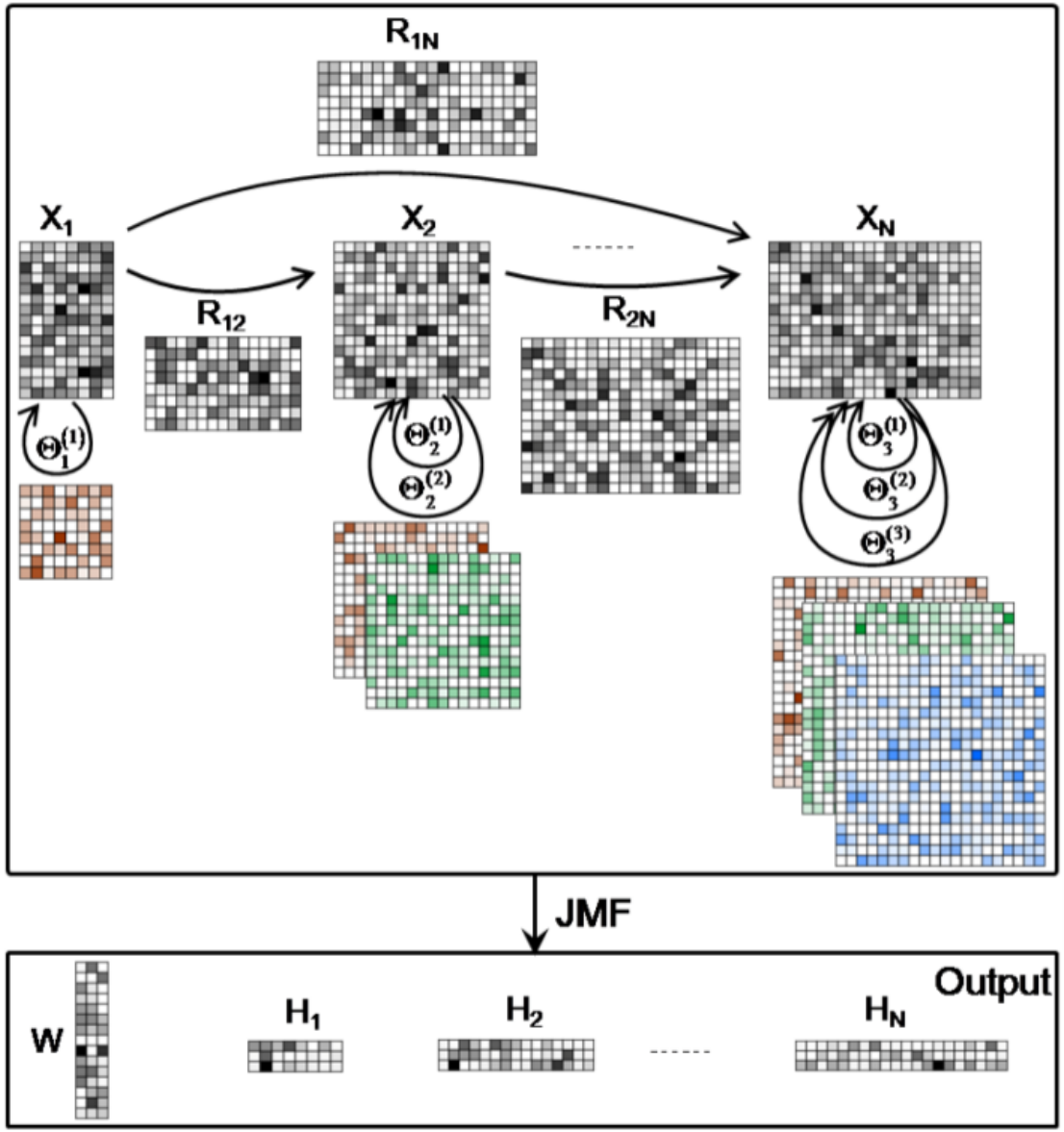


Figure: Illustration of JMF

- Veracity: data are noisy. Moreover, data of different views/sources may demonstrate different levels of noise

Shortcoming

Ignore the heterogeneous noise among the multi-view data

[Zhang and Zhang, 2017] proposed a Bayesian Joint Matrix Decomposition model to remedy this shortcoming

Bayesian Joint Matrix Decomposition (BJMD)

Given data matrix $X_1 \in R^{m \times n_1}, \ldots, X_c \in R^{m \times n_c}$, we assume that the data matrices are generated:

$$X_i = WH_i + \epsilon_i$$

Similar to GLAD

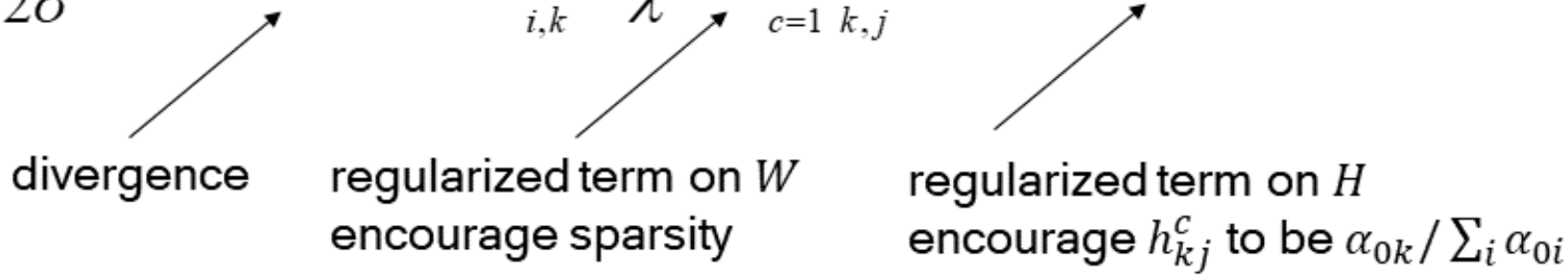
- $\epsilon_i \sim N(0, \sigma_i^2)$ models the heterogeneous noise of different source
- W has a Laplace prior for sparsity
- H_i has a Dirichlet prior for interpretability

Complete Log Likelihood of BJMD

The complete log likelihood can help us to understand it from optimization perspective

$$-LL(W, H^{(1)}, \ldots, H^{(C)} \mid X^{(1)}, \ldots, X^{(C)})$$

$$= \sum_{c=1}^{C} \sum_{i,j} \frac{1}{2\sigma^2} (x_{ij}^{(c)} - w_{i\cdot} h_{\cdot j}^{(c)})^2 + \sum_{i,k} \frac{|w_{ik}|}{\lambda} - \sum_{c=1}^{C} \sum_{k,j} (\alpha_{0k} - 1) \ln h_{kj}^{(c)}$$



- Zhang and Zhang proposed two effective algorithms to solve BJMD

Effects of Heterogeneous Noise

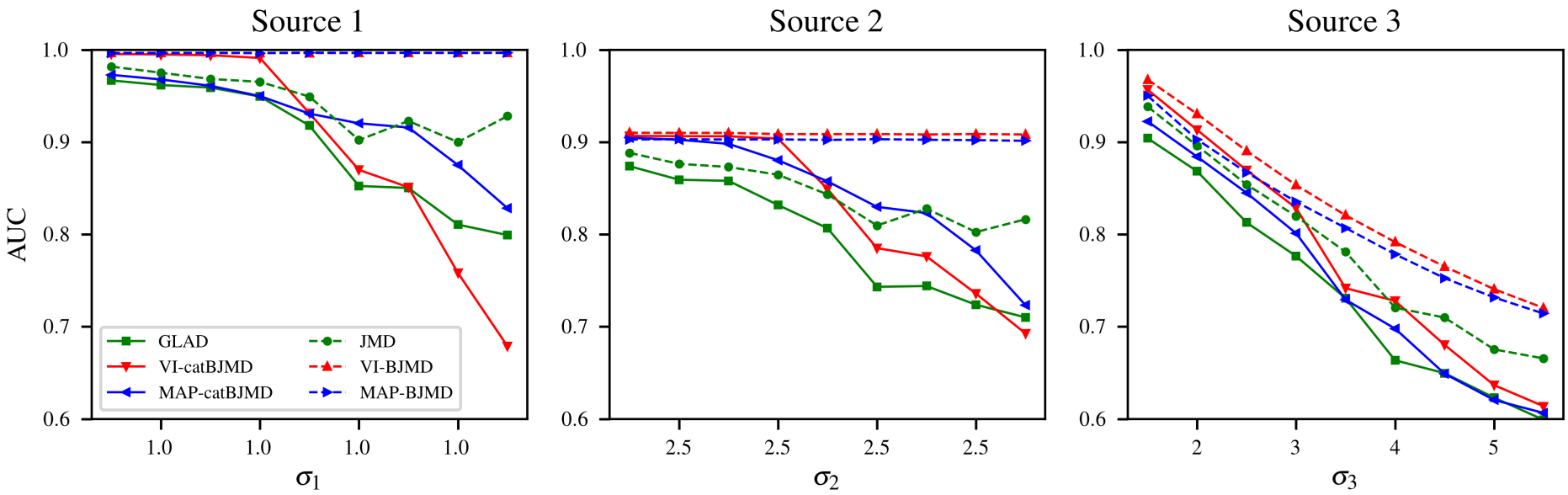


Figure: Performance comparison on simulation data.

Summary

- NMF is a powerful tool in dimension reduction and pattern recognition
- We introduce some algorithms
- Exact NMF is NP-hard, but local minimal is usually good enough for application
- Many variants of NMF have been proposed for a wide range of applications

References I

Arora, S., Ge, R., Kannan, R., and Moitra, A. (2012).

Computing a nonnegative matrix factorization–provably.

In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 145–162. ACM.

Benson, A. R., Lee, J. D., Rajwa, B., and Gleich, D. F. (2014).

Scalable methods for nonnegative matrix factorizations of near-separable tall-and-skinny matrices.

In *Advances in Neural Information Processing Systems*, pages 945–953.

Buciu, I., Nikolaidis, N., and Pitas, I. (2008).

Nonnegative matrix factorization in polynomial feature space.

IEEE Transactions on Neural Networks, 19(6):1090–1100.

Cai, D., He, X., Han, J., and Huang, T. S. (2010).

Graph regularized nonnegative matrix factorization for data representation.

IEEE transactions on pattern analysis and machine intelligence, 33(8):1548–1560.

Chen, J.-C. (1984).

The nonnegative rank factorizations of nonnegative matrices.

Linear algebra and its applications, 62:207–217.

Choi, S. (2008).

Algorithms for orthogonal nonnegative matrix factorization.

In *2008 ieee international joint conference on neural networks (ieee world congress on computational intelligence)*, pages 1828–1832. IEEE.

Cichocki, A., Lee, H., Kim, Y.-D., and Choi, S. (2008).

Non-negative matrix factorization with α-divergence.

Pattern Recognition Letters, 29(9):1433–1440.

References II

Cichocki, A., Zdunek, R., Phan, A. H., and Amari, S.-i. (2009).

Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. John Wiley & Sons.

Ding, C., Li, T., Peng, W., and Park, H. (2006).

Orthogonal nonnegative matrix t-factorizations for clustering.
In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM.

Ding, C. H., Li, T., and Jordan, M. I. (2008).

Convex and semi-nonnegative matrix factorizations.
IEEE transactions on pattern analysis and machine intelligence, 32(1):45–55.

Guan, N., Tao, D., Luo, Z., and Yuan, B. (2012a).

Nenmf: An optimal gradient method for nonnegative matrix factorization.
IEEE Transactions on Signal Processing, 60(6):2882–2898.

Guan, N., Tao, D., Luo, Z., and Yuan, B. (2012b).

Online nonnegative matrix factorization with robust stochastic approximation.
IEEE Transactions on Neural Networks and Learning Systems, 23(7):1087–1099.

Jeter, M. and Pye, W. (1981).

A note on nonnegative rank factorizations.
Linear Algebra and its Applications, 38:171–173.

Jia, Y. W. Y. and Turk, C. H. M. (2004).

Fisher non-negative matrix factorization for learning local features.
In *Proc. Asian conf. on comp. vision*, pages 27–30. Citeseer.

References III

Kim, H. and Park, H. (2008a).

Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method.
SIAM journal on matrix analysis and applications, 30(2):713–730.

Kim, H. and Park, H. (2008b).

Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method.
SIAM journal on matrix analysis and applications, 30(2):713–730.

Lee, D. D. and Seung, H. S. (1999).

Learning the parts of objects by non-negative matrix factorization.
Nature, 401(6755):788.

Lee, D. D. and Seung, H. S. (2001).

Algorithms for non-negative matrix factorization.
In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press.

Li, S. Z., Hou, X., Zhang, H., and Cheng, Q. (2001).

Learning spatially localized, parts-based representation.
CVPR (1), 207:212.

Liang, Z., Li, Y., and Zhao, T. (2010).

Projected gradient method for kernel discriminant nonnegative matrix factorization and the applications.
Signal Processing, 90(7):2150–2163.

Lin, C.-J. (2007).

Projected gradient methods for nonnegative matrix factorization.
Neural computation, 19(10):2756–2779.

References IV

Nesterov, Y. (2004).

Lectures on convex optimization, volume 137.
MA: Kluwer Academic, Boston.

Paatero, P. and Tapper, U. (1994).

Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values.
Environmetrics, 5(2):111–126.

Pascual-Montano, A., Carazo, J. M., Kochi, K., Lehmann, D., and Pascual-Marqui, R. D. (2006).

Nonsmooth nonnegative matrix factorization (nsnmf).
IEEE transactions on pattern analysis and machine intelligence, 28(3):403–415.

Saddiki, H., McAuliffe, J., and Flaherty, P. (2014).

Glad: a mixed-membership model for heterogeneous tumor subtype classification.
Bioinformatics, 31(2):225–232.

Trigeorgis, G., Bousmalis, K., Zafeiriou, S., and Schuller, B. W. (2016).

A deep matrix factorization method for learning attribute representations.
IEEE transactions on pattern analysis and machine intelligence, 39(3):417–429.

Vavasis, S. A. (2009).

On the complexity of nonnegative matrix factorization.
SIAM Journal on Optimization, 20(3):1364–1377.

Wang, Y.-X. and Zhang, Y.-J. (2012).

Nonnegative matrix factorization: A comprehensive review.
IEEE Transactions on Knowledge and Data Engineering, 25(6):1336–1353.

References V

Welch, J. D., Kozareva, V., Ferreira, A., Vanderburg, C., Martin, C., and Macosko, E. Z. (2019).

Single-cell multi-omic integration compares and contrasts features of brain cell identity.
Cell.

Yang, Z. and Michailidis, G. (2015).

A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data.
Bioinformatics, 32(1):1–8.

Zafeiriou, S., Tefas, A., Buciu, I., and Pitas, I. (2006).

Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification.
IEEE Transactions on Neural Networks, 17(3):683–695.

Zhang, C., Jing, L., and Xiu, N. (2014).

A new active set method for nonnegative matrix factorization.
SIAM Journal on Scientific Computing, 36(6):A2633–A2653.

Zhang, C. and Zhang, S. (2017).

Bayesian joint matrix decomposition for data integration with heterogeneous noise.
arXiv preprint arXiv:1712.03337.

Zhang, L. and Zhang, S. (2019a).

A general joint matrix factorization framework for data integration and its systematic algorithmic exploration.
IEEE Transactions on Fuzzy Systems.

Zhang, L. and Zhang, S. (2019b).

Learning common and specific patterns from data of multiple interrelated biological scenarios with matrix factorization.
Nucleic acids research, 47(13):6606–6617.

References VI

Zhang, S., Li, Q., Liu, J., and Zhou, X. J. (2011).

A novel computational framework for simultaneous integration of multiple types of genomic data to identify microrna-gene regulatory modules.

Bioinformatics, 27(13):i401–i409.