Nonnegative Matrix Factorization (NMF): Models, Algorithms and Applications

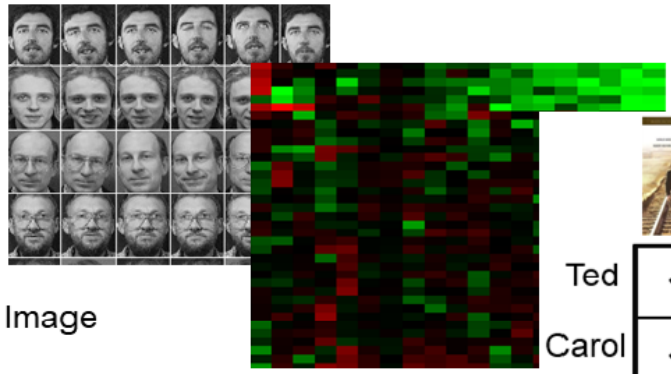
Shihua Zhang

Fall 2019

Overview

1. What is NMF?
2. Algorithms of NMF
3. Complexity of NMF
4. Variants of NMF
   - Constrained NMF
   - Structured NMF
   - Generalized NMF
5. NMF towards Big Data

Many Data Are Nonnegative!



Image

Gene expression

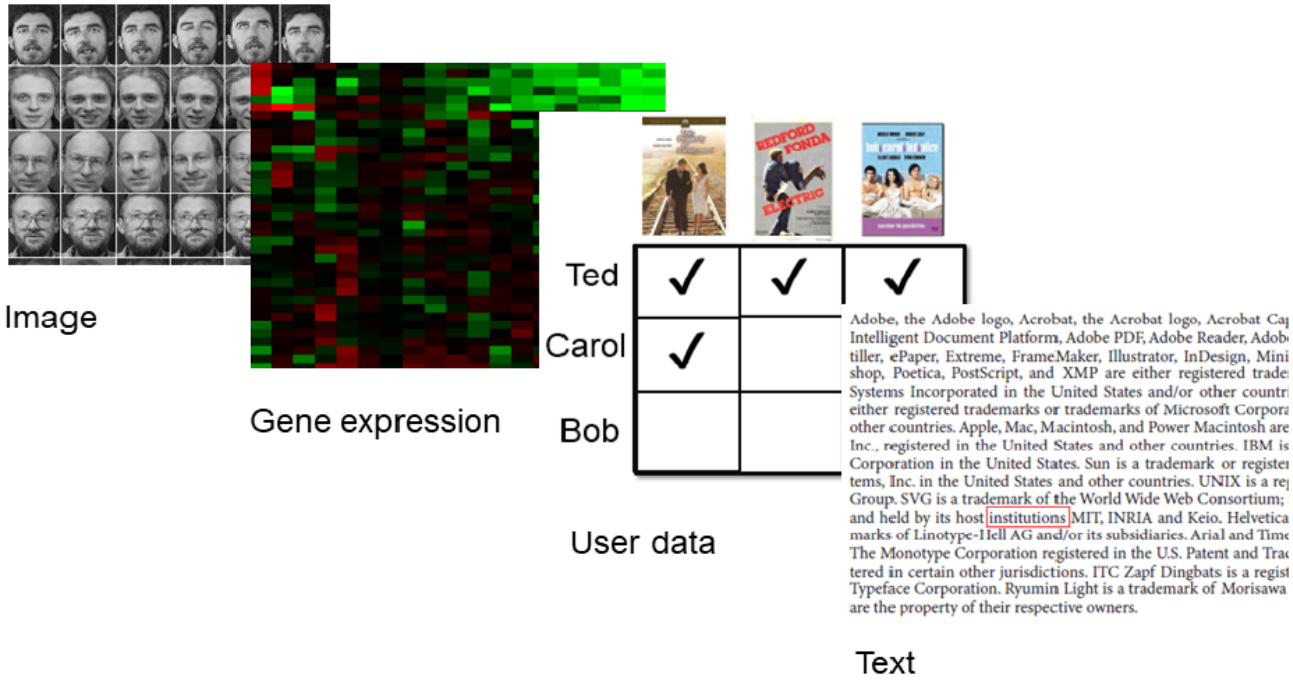
Ted	✓	✓	✓
Carol	✓		
Bob			

User data

Adobe, the Adobe logo, Acrobat, the Acrobat logo, Acrobat Cap
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 register
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 Trad
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

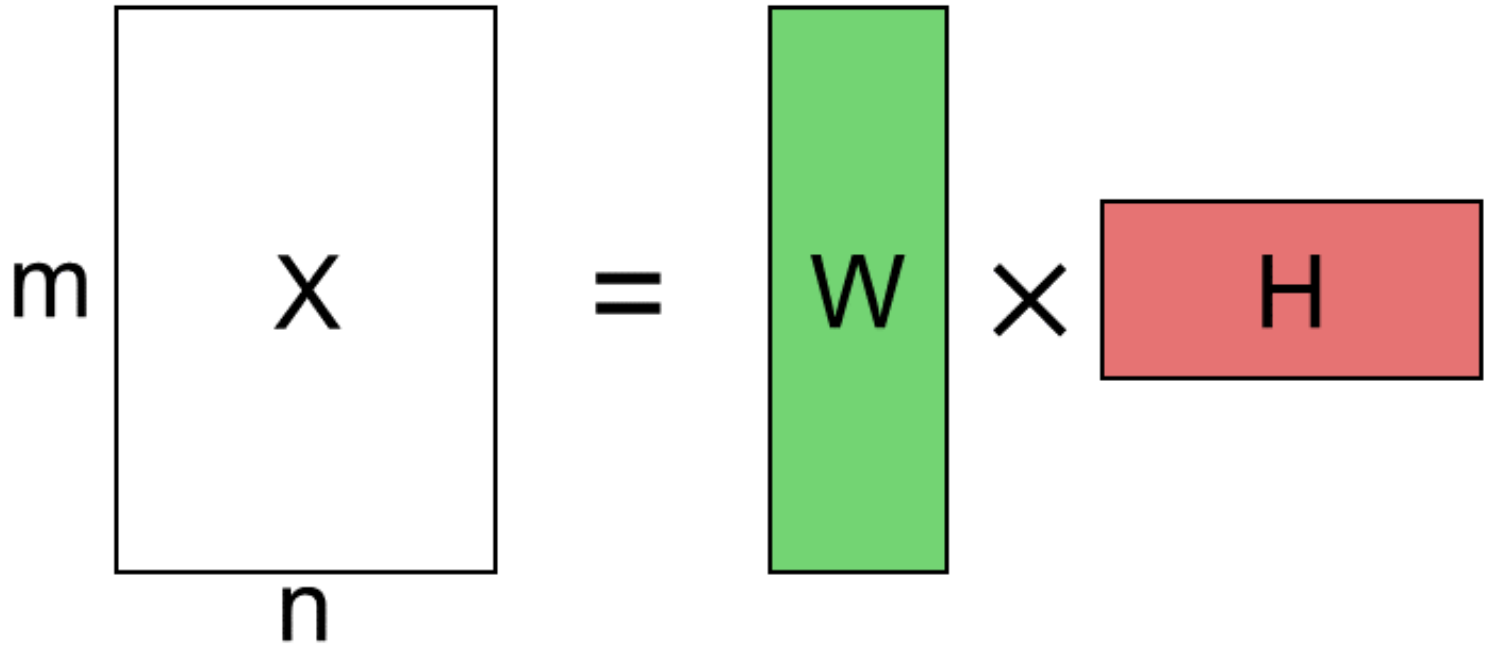
Many Data Are Nonnegative!



Problem

Given a (non-negative) data matrix $X \in R^{m \times n}$ ($R_+^{m \times n}$) with m features and n samples, how to extract information from the high-dimensional and redundant primitive data?

Illustration of Matrix Factorization



Matrix Factorization is an effective way!!!

What is NMF?

Given $X \in R_+^{m \times n}$, 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 $r < \min(m, n)$.

What is NMF?

Given $X \in R_+^{m \times n}$, 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 $r < \min(m, n)$.

Optimization problem:

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

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

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, it have been widely used in various research areas for diverse applications

Why does it work? The intuition.

Each column of data is approximated by

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

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

Why does it work? The intuition.

Each column of data is approximated by

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

where $w_{\cdot i}$ is the column vectors of W. Therefore, $x_{\cdot j}$ 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

NMF vs PCA

Given $X \in R^{m \times n}$ (or $R_+^{m \times n}$) with proper preprocessing, PCA and NMF can be formulated as follows

Principal Component Analysis (PCA)

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

where $W \in R^{m \times r}$ is the basis matrix, and $H \in R^{r \times n}$ is the coefficient matrix.

NMF vs PCA

Given $X \in R^{m \times n}$ (or $R_+^{m \times n}$) with proper preprocessing, PCA and NMF can be formulated as follows

Principal Component Analysis (PCA)

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

where $W \in R^{m \times r}$ is the basis matrix, and $H \in R^{r \times n}$ is the coefficient matrix.

Non-negative Matrix Factorization (NMF)

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

where $W \in R_+^{m \times r}$ is the basis matrix, and $H \in R_+^{r \times n}$ is the coefficient matrix.

NMF vs PCA

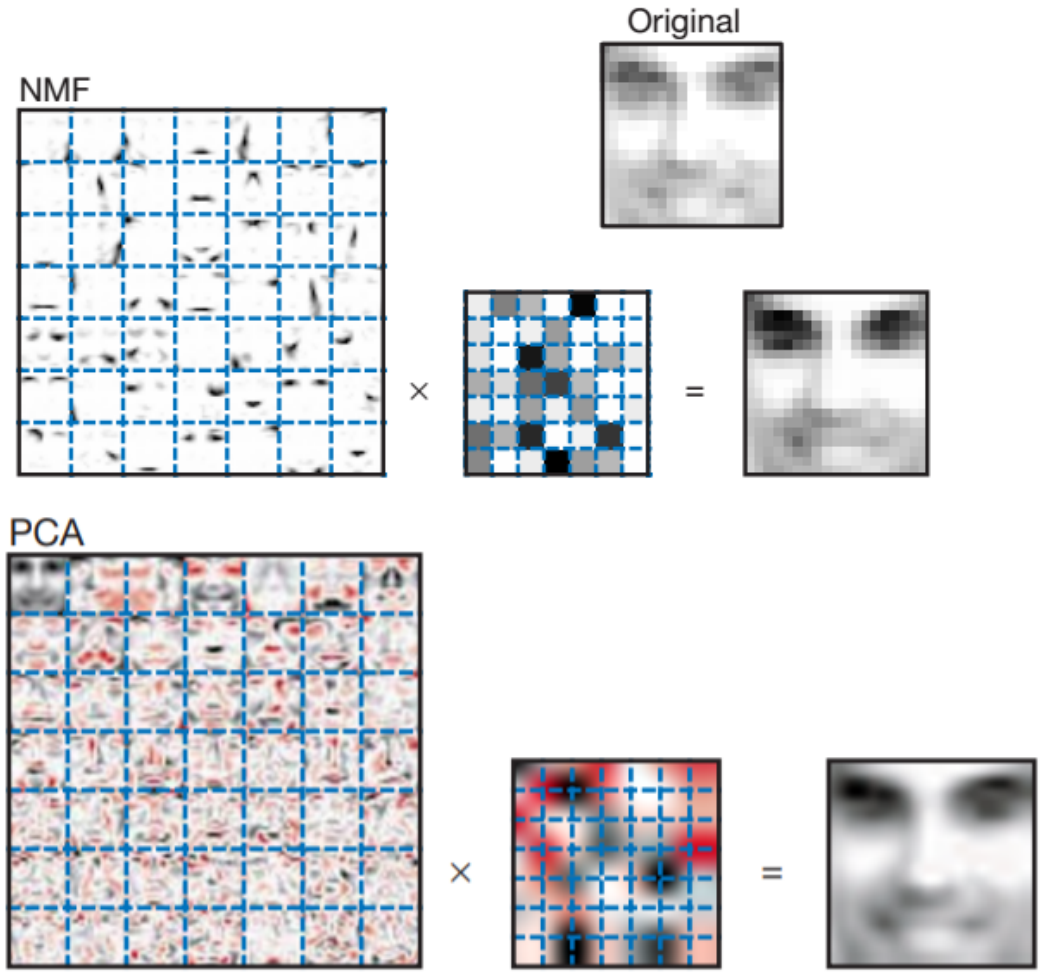


Figure: Representation of human face [Lee and Seung, 1999]

How to Measure the Divergence?

Recall the general **optimization problem** of NMF:

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

Divergence functions measures the similarity of the primitive data matrix X and the approximated matrix WH.

How to Measure the Divergence?

Recall the general **optimization problem** of NMF:

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

Divergence functions measures the similarity of the primitive data matrix X and the approximated matrix WH.
Two most commonly used are:

Frobenius Norm

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

Kullback-Leibler (KL) Divergence

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

How to Choose the Divergence Function?

Problem

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

How to Choose the Divergence Function?

Problem

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

Answer

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

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

Algorithms for NMF

The NMF problem

- Non-convex optimization which is NP-hard [Vavasis, 2009]
- Local optimization algorithms are adopted
- Alternatively update W and H

Algorithms for NMF

The NMF problem

- Non-convex optimization which is NP-hard [Vavasis, 2009]
- Local optimization algorithms are adopted
- Alternatively update W and H

Choose the Frobenius norm as the divergence function for simplicity

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

Local Optimization Algorithms

Most of the existing algorithms update W and H alternatively

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

Local Optimization Algorithms

Most of the existing algorithms update W and H alternatively

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

Subproblem 1

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

Local Optimization Algorithms

Most of the existing algorithms update W and H alternatively

- Multiplicative Update Rule [Lee and Seung, 2001]
- Projected Gradient [Lin, 2007]
- Active Set [Kim and Park, 2008]
- 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 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:

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

Multiplicative Update Rule (MUR)

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

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

Multiplicative Update Rule (MUR)

The MUR can also be derived based on the use of gradient descent (with a specific learning rate).

Multiplicative Update Rule (MUR)

The MUR can also be derived based on the use of gradient descent (with a specific learning rate).

- This is a multiplicative update instead of an additive update.
- If the initial values of W and H are all non-negative, then the W and H can never become negative.

This lets us produce a non-negative factorization.

Multiplicative Update Rule (MUR)

The MUR can also be derived based on the use of gradient descent (with a specific learning rate).

- This is a multiplicative update instead of an additive update.
- If the initial values of W and H are all non-negative, then the W and H can never become negative.

This lets us produce a non-negative factorization.

A General trick

- KKT condition
- Update rule
- Auxiliary function
- Prove monotonicity

Summary of MUR

Pros

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

Cons

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

Projected Gradient (PG)

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

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 (AS)

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

- Problem: Can we obtain a much faster algorithm?
- Yes! an active set algorithm was proposed for NMF [Kim and Park, 2008]

Active Set (AS)

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

- Problem: Can we obtain a much faster algorithm?
- Yes! an active set algorithm was proposed for NMF [Kim and Park, 2008]

The 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

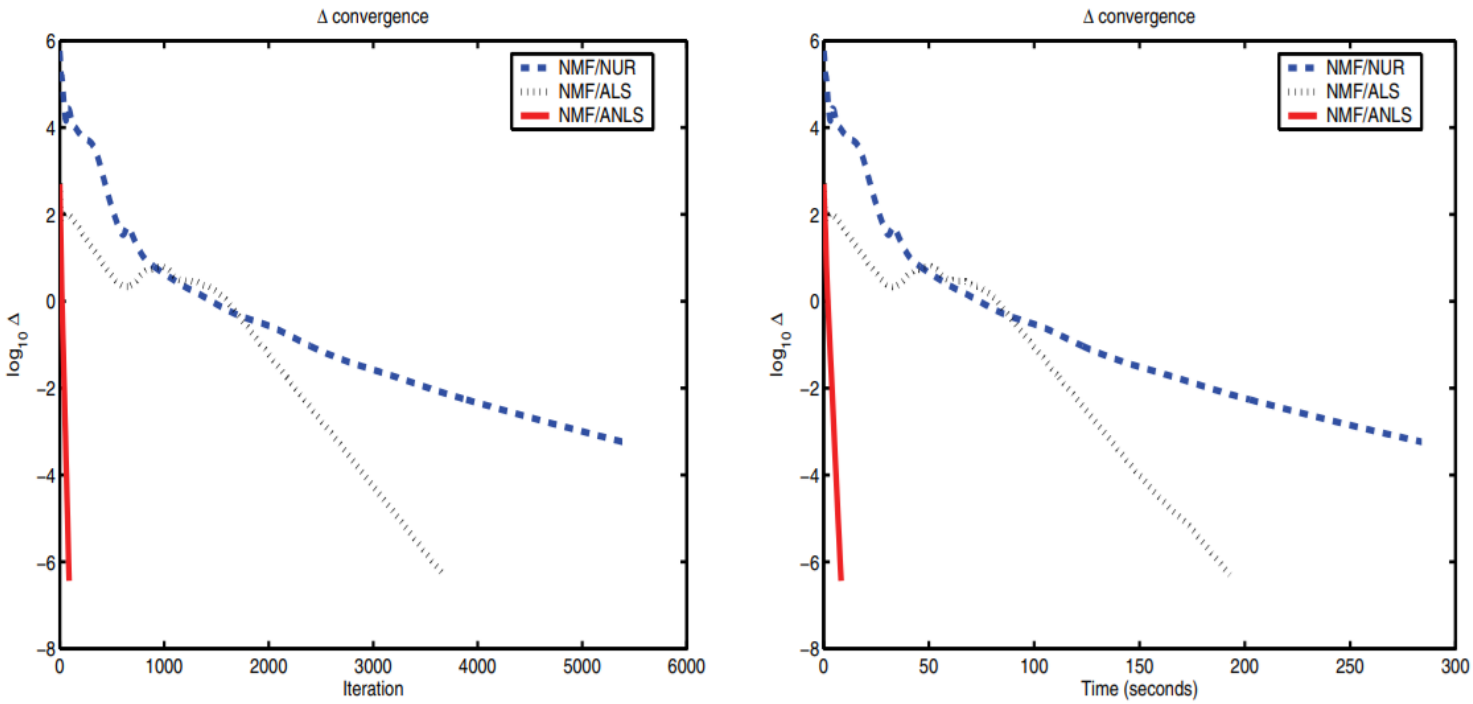


Figure: Convergence speed comparison. The blue dots line indicate MUR and the red solid line indicates AS. The grey dots line indicates an alternative least square algorithm with projection [Berry et al., 2007]

Summary of AS

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 for NMF (NeNMF)

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 minimum

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	$O(mnr + n^2r + mr^2 + nr^2) + K \times O(mr^2 + nr^2)$

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 the local search by alternatively updating W and H
- 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?

NMF is NP-hard.

EXACT NMF

Given a nonnegative matrix X, the output is *yes*, if there exists a pair of matrices (W, H) such that $X = WH$, else it is *no*.

NMF is NP-hard.

EXACT NMF

Given a nonnegative matrix X, the output is *yes*, if there exists a pair of matrices (W, H) such that $X = WH$, else it is *no*.

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

- Show the equivalence between the EXACT NMF and the INTERMEDIATE SIMPLEX problem in polyhedral combinatorics.
- Show the NP-hardness of this problem.

How to Find the Global Solution?

- Due to its NP-hardness, 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?

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.

1. What is NMF?
2. Algorithms of NMF
3. Complexity of NMF
4. Variants of NMF
   - Constrained NMF
   - Structured NMF
   - Generalized NMF
5. NMF towards Big Data

Variants of NMF

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

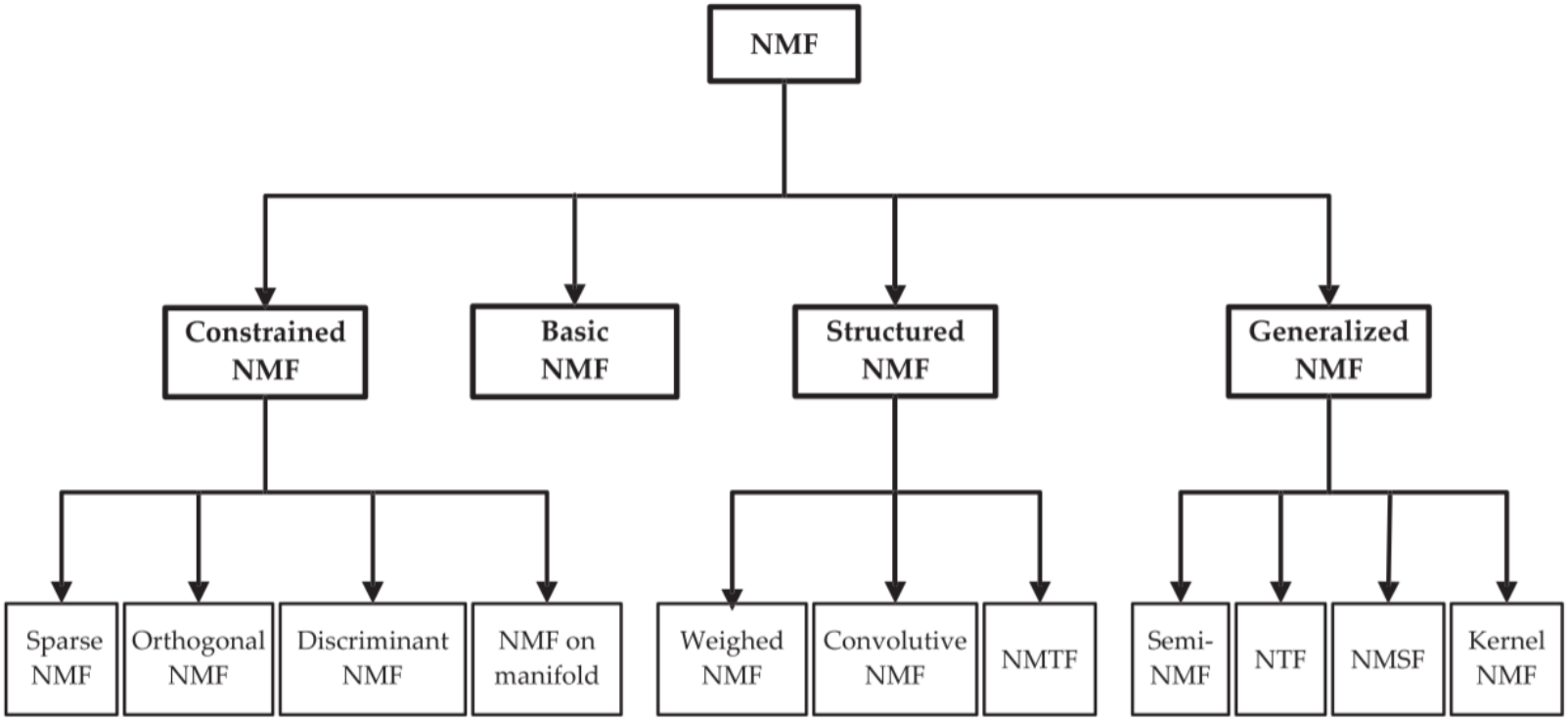


Figure: The taxonomy of NMF models [Wang and Zhang, 2012]

Constrained NMF

Here is the general form of constrained NMF

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

Constrained NMF

Here is the general form of 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 subclasses:

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

Part-based representation of NMF

NMF doesn't always result in part-based representations!!!

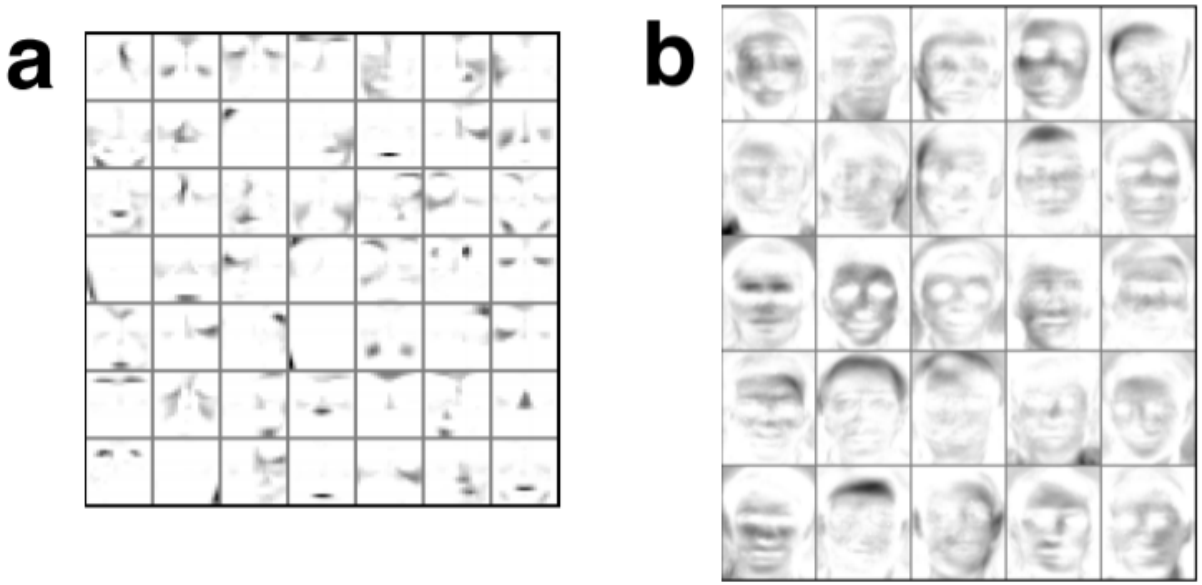


Figure: Basis faces of CBCL database (left) and ORL database (right)

Part-based representation of NMF

NMF doesn't always result in part-based representations!!!

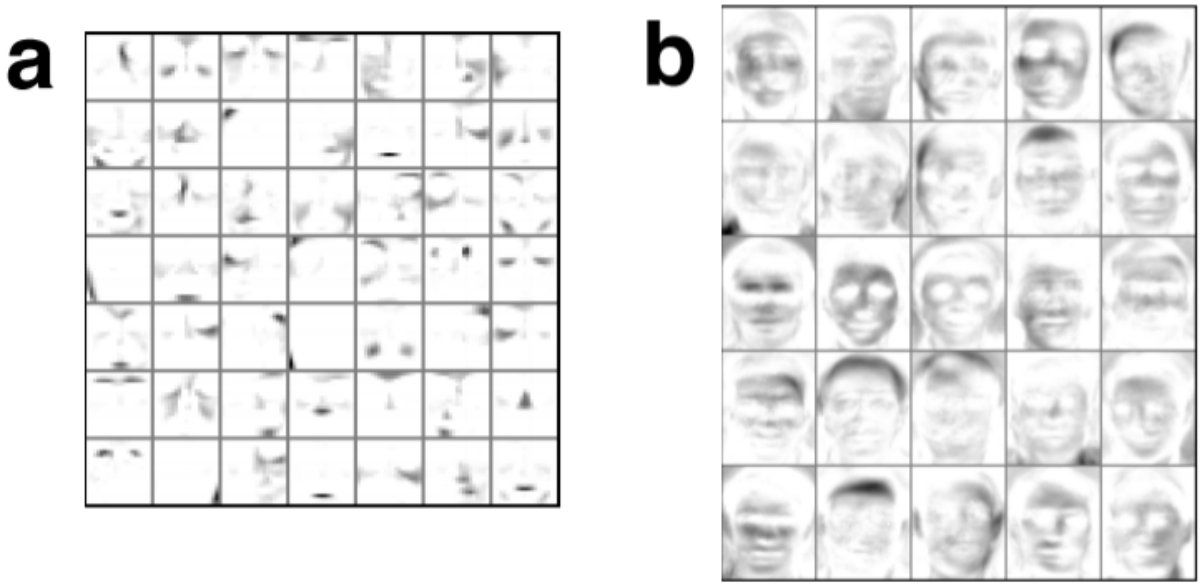


Figure: Basis faces of CBCL database (left) and ORL database (right)

Problem: How to obtain the part-based representation?

- Add sparse constraints: sparse NMF
- Add orthogonal constraints: orthogonal NMF

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

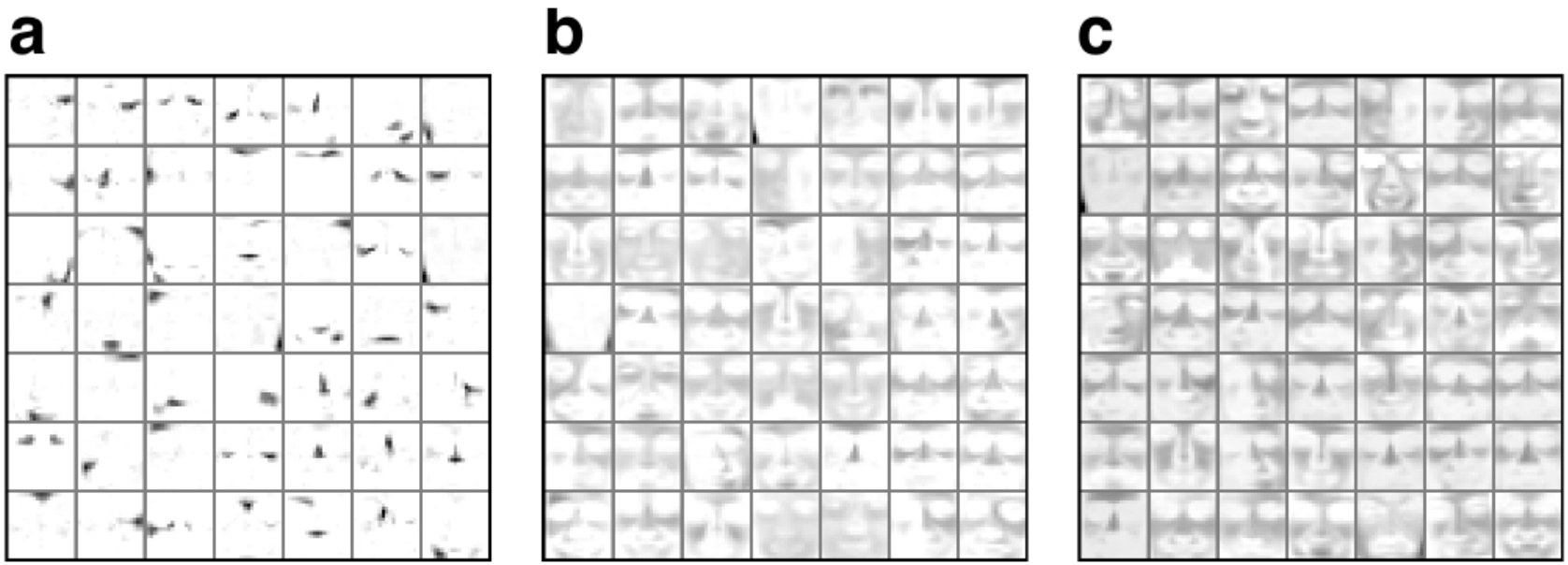


Figure: (a) Fix the sparseness of the basis matrix to 0.8 spa$(W) = 0.8$ (b) spa$(H) = 0.8$ (c) spa$(W) = 0.2$

Orthogonal NMF (Local NMF)

Idea: add orthogonal constraints

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

Local NMF tries to add orthogonal constraints implicitly [Li et al., 2001]. Let $U = W^T W$ and $V = HH^T$.

Orthogonal NMF (Local NMF)

Idea: add orthogonal constraints

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

Local NMF tries to add orthogonal constraints implicitly [Li et al., 2001]. Let $U = W^T W$ and $V = HH^T$.

Local NMF (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

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

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

Benefits of Orthogonal NMF

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

Benefits of Orthogonal NMF

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

Benefits of Orthogonality

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

Incorporate Discriminant Information to NMF

NMF are typically considered as unsupervised learning.

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

To use the information of the classes, the within scatter

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

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

Between/Within Scatter

To use the information of the classes, the within scatter

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

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

Similarly, the between scatter

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

Fisher NMF

Incorporate the between/within scatter 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 could achieve better performance in human face recognition than NMF and LNMF.

NMF on Manifold (Graph-regularized NMF)

Motivations

- The real-world data are often sampled from a nonlinear low-dimensional manifold
- NMF fails to discover the intrinsic geometrical structure
- Idea: if two data points are close in the data distribution, then the representation of the two points are also close to each other

NMF on Manifold (Graph-regularized NMF)

Motivations

- The real-world data are often sampled from a nonlinear low-dimensional manifold
- NMF fails to discover the intrinsic geometrical structure
- Idea: if two data points are close in the data distribution, then the representation of the two points are also close to each other

Graph-regularized NMF (GNMF): model the manifold by constructing a nearest neighborhood graph of data points [Cai et al., 2010]

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

where $r(H)$ is the graph regularizer

How to Construct the Graph

Use a graph with the 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

How to Construct the Graph

Use a graph with the 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.

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.

The GNMF is formulated as

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

It showed better performance than NMF in clustering tasks

Structured NMF

Structured NMF modifies the regular factorization directly.

$$X \approx F(WH)$$

Structured NMF

Structured NMF modifies the regular factorization directly.

$$X \approx F(WH)$$

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 models could be too restrictive and may lead to a poor low-rank representation!

Non-negative Matrix Trifactorization

Potential crisis of constrained NMF

Additional constraints help obtaining

- Sparse NMF
- Orthogonal NMF

But those models could be too restrictive and may lead to a poor low-rank representation!

Nonsmooth NMF (nsNMF)

nsNMF extends the NMF model to trifactorization [Pascual-Montano et al., 2006]

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

The Smoothing Matrix

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

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

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

The Smoothing Matrix

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

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

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

Let $Y = WS$

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

Generalized NMF

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

Generalized NMF

Generalized NMF is the extension 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!!!

Semi-NMF

In some cases, the data matrix are real-valued.
NMF is not directly applicable!!!

Matrix factorization methods suitable for real-valued data are 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.

Semi-NMF

In some cases, the data matrix are real-valued.
NMF is not directly applicable!!!

Matrix factorization methods suitable for real-valued data are 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.

The non-negative coefficient matrix retains the interpretability and the nature of sparsity.

Convex NMF

Motivation: In NMF and Semi-NMF, there are no constraints on the basis matrix. To improve interpretability, we may restrict the basis matrix to convex combination of the columns of X.

$$W = XF$$

Convex NMF

Motivation: In NMF and Semi-NMF, there are no constraints on the basis matrix. To improve interpretability, we may restrict the basis matrix to convex combination of the columns of X.

$$W = XF$$

The we can get

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

Convex NMF

Motivation: In NMF and Semi-NMF, there are no constraints on the basis matrix. To improve interpretability, we may restrict the basis matrix to convex combination of the columns of X.

$$W = XF$$

The we can get

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

Advantages:

- 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

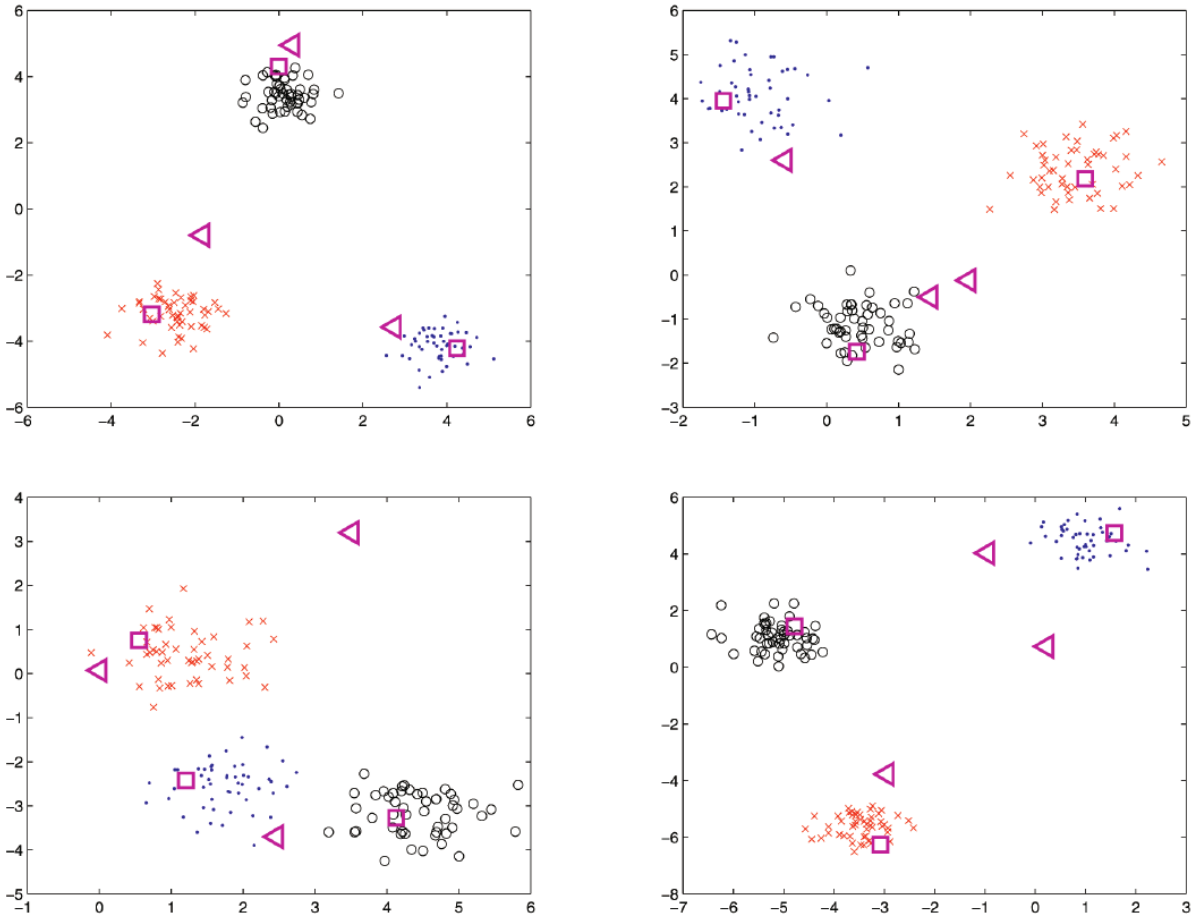


Figure: Four random datasets, each with three clusters. $\square$ for convex NMF and $\triangleleft$ for semi NMF

Kernel NMF

How to extract non-linear structure? Apply kernel ideas to map input data into implicit feature space!

Kernel NMF

How to extract non-linear structure? Apply kernel ideas to map input data into implicit feature space!

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

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

NMF towards Big Data

The characteristics of big data bring new challenges.

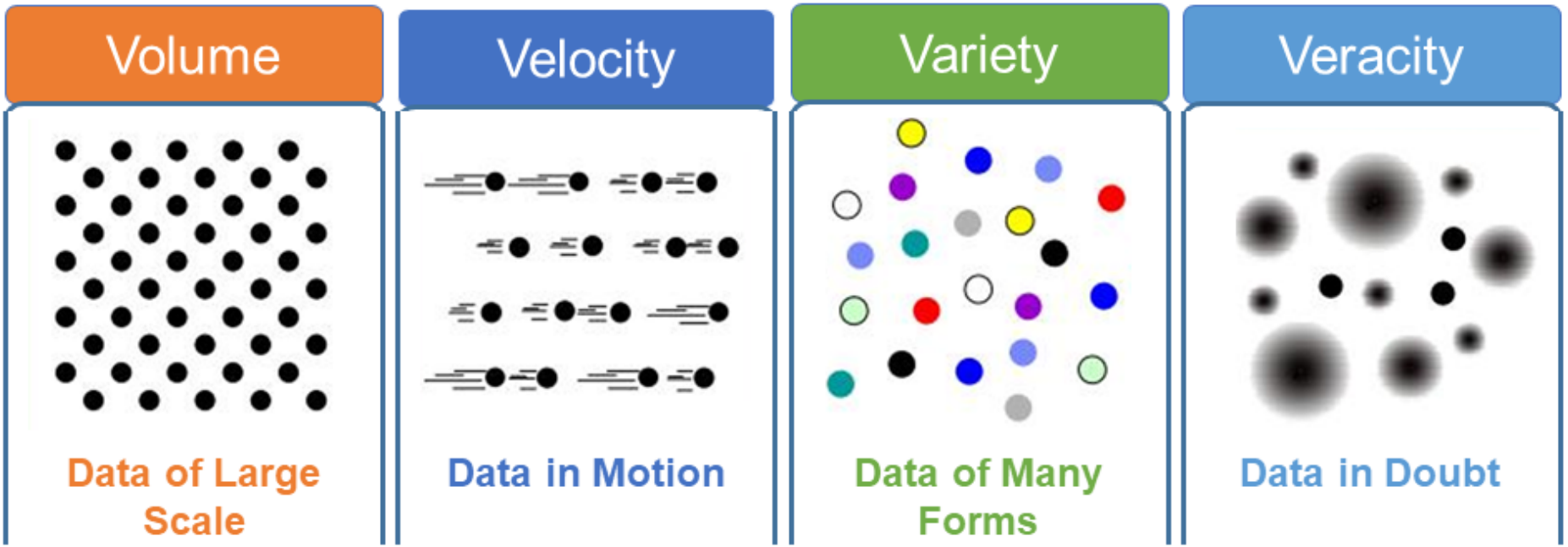


Figure: Characteristics of big data

How to address those challenges (even partially)?

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

How to address those challenges (even partially)?

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

How to address those challenges (even partially)?

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

How to address those challenges (even partially)?

- Volume: analyze large-scale data efficiently
- Velocity: analyze 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

Problem

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

Preliminary

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

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 to reduce the problem scale by the 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

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

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

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

Problem Formulation of online RSA-NMF

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 $\{w_1, \ldots, w_n\} \in R_+^m$

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

Minimizing the Expected Cost and Updating W Online

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

Minimizing the Expected Cost and Updating W Online

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

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 relatively low

Deep Semi-NMF

Limitation of NMF

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

- Inspired by the success of deep models, deep semi-NMF model was proposed [Trigeorgis et al., 2016]
- 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

$$H_{m-1}^{+} \approx W_m^{\pm} H_m^{+}$$

$$\vdots$$

$$H_1^{+} \approx W_2^{\pm} \cdots W_m^{\pm} H_m^{+}$$

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

Illustration of Deep Semi-NMF

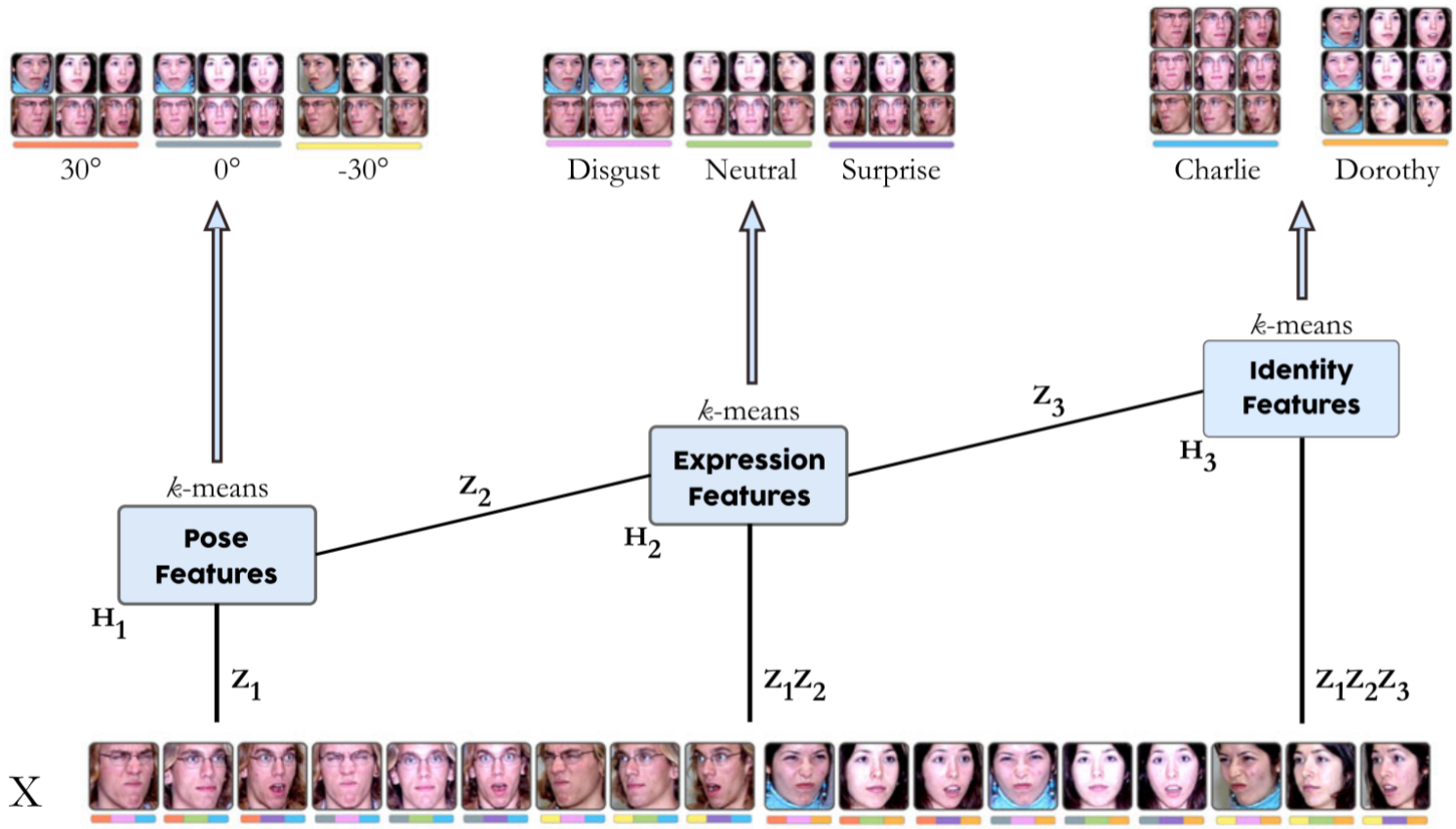


Figure: A deep semi-NMF model learns a hierarchical structure of features

Bayesian Matrix Factorization

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

Bayesian Matrix Factorization

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

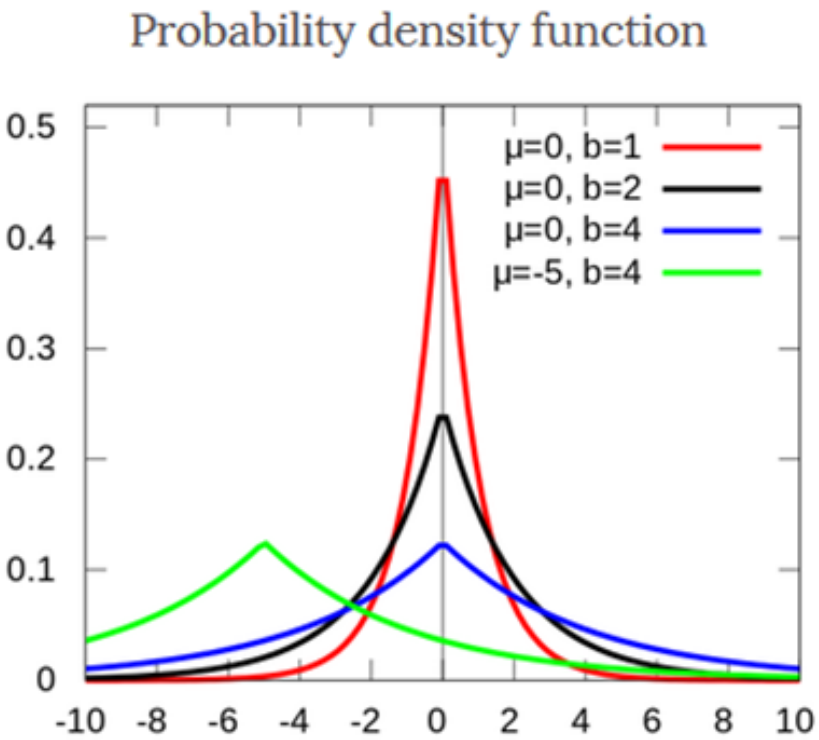
Why Bayesian Matrix Factorization

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

$$X = WH + \epsilon$$

Laplace Distribution

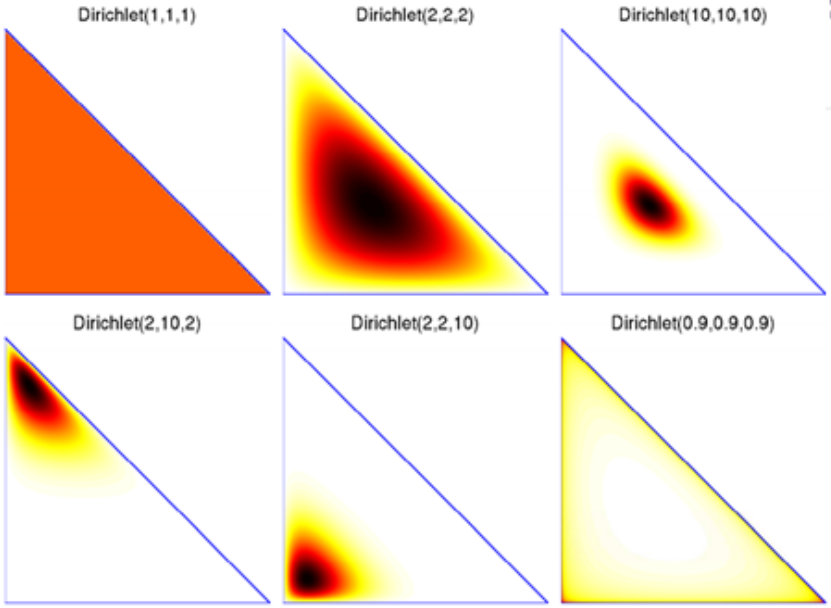
- Laplace distribution offers the l_1 penalty in the likelihood function



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 distributions, i.e Gaussian, Laplace and Dirichlet distributions

$$X = WH + \epsilon$$

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

Variational Inference (VI) for GLAD

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

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

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

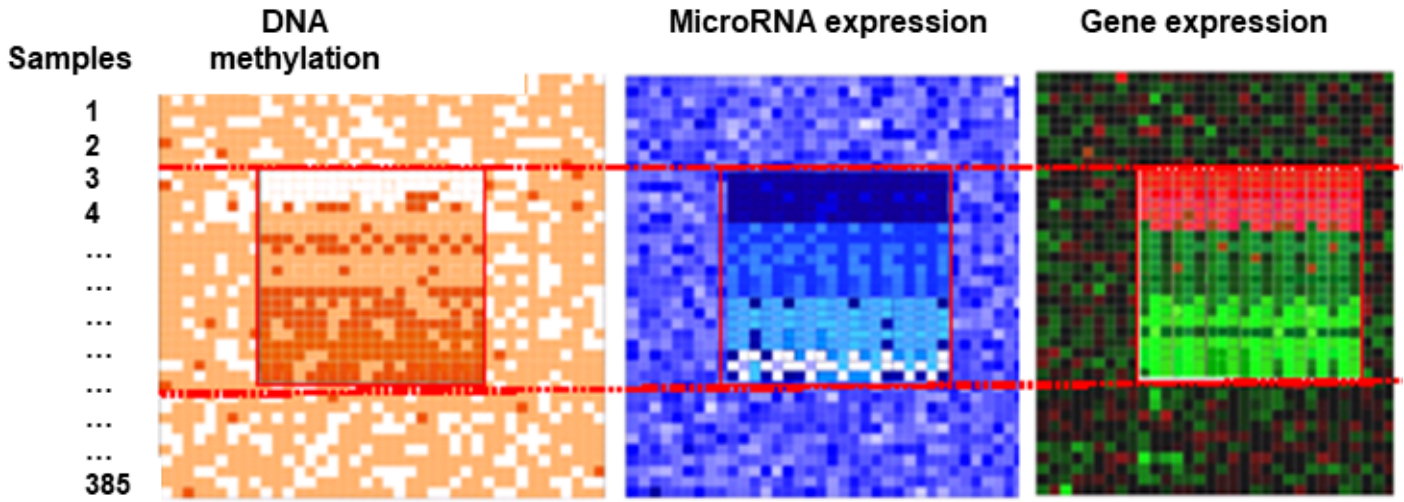
- If $\mathcal{L}$ can be computed analytically, VI turns the inference to an optimization problem
- GLAD further approximates $\mathcal{L}$ by the Laplace approximation
  - Computationally expensive, unrealistic for real-world data

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

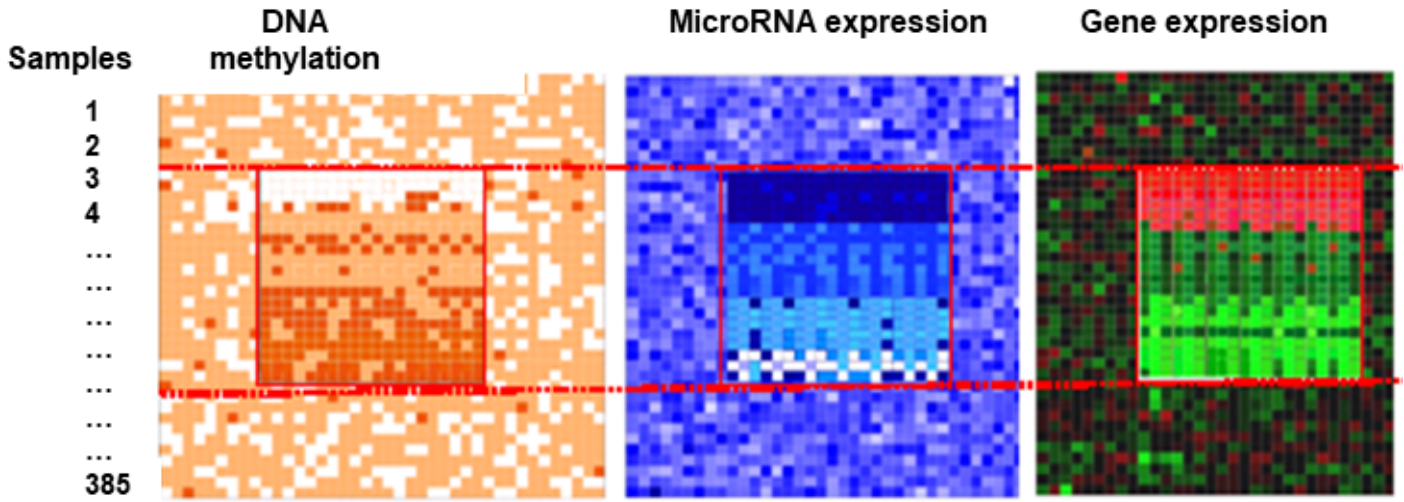
Mutli-view Data

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

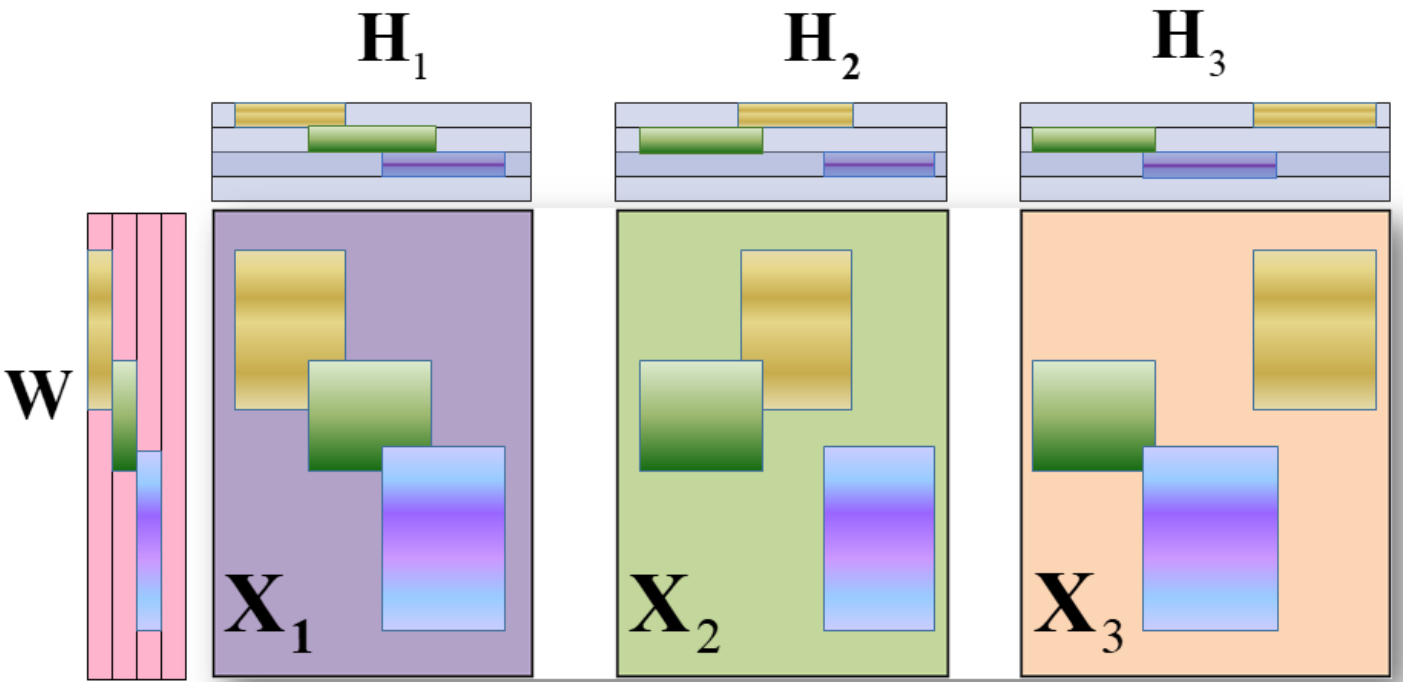
Problem

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



Joint NMF (jNMF)

- Goal: Identify multi-dimensional modules across multiple types of genomic data [**?**]



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

Extensions of jNMF

jNMF is a powerful tool for integrative analysis of multi-view data. Data from different sources share the same basis matrix [Zhang et al., 2012].

Extensions of jNMF

jNMF is a powerful tool for integrative analysis of multi-view data.
Data from different sources share the same basis matrix
[Zhang et al., 2012].

Motivation of Integrative NMF (iNMF)

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

iNMF was proposed to address this challenge
[Yang and Michailidis, 2015]

Extensions of jNMF

jNMF is a powerful tool for integrative analysis of multi-view data. Data from different sources share the same basis matrix [Zhang et al., 2012].

Motivation of Integrative NMF (iNMF)

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

iNMF was proposed to address this challenge [Yang and Michailidis, 2015]

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.

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

Application of iNMF

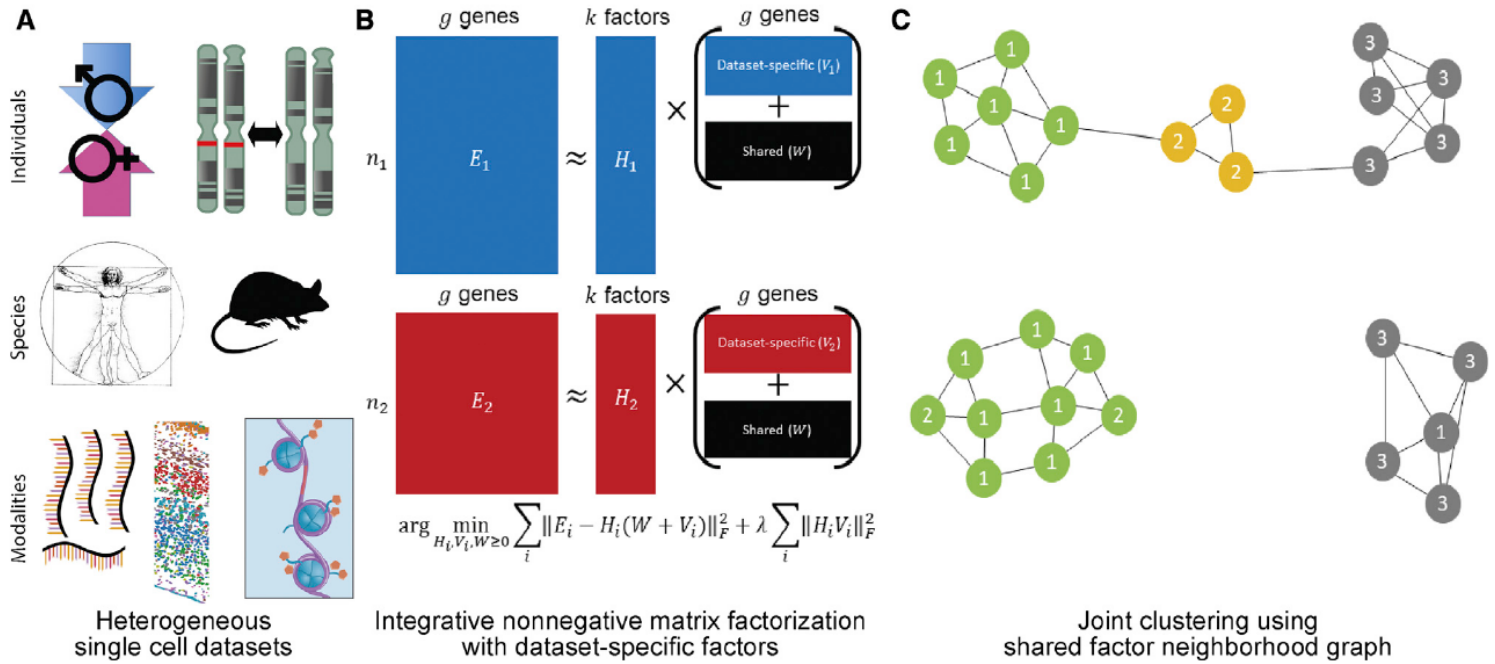


Figure: Discover the common and specific pattern in singe-cell multi-view data [Welch et al., 2019]

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

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

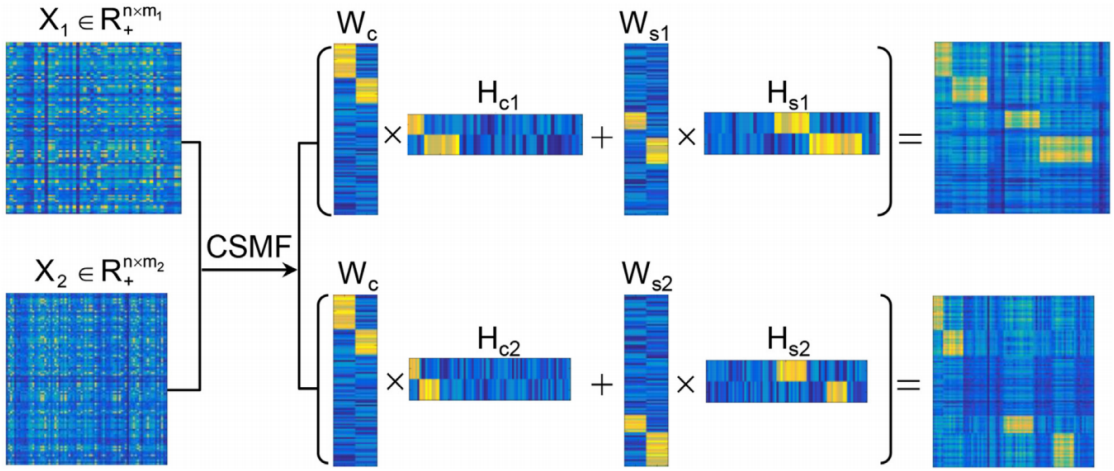


Figure: Illustration of CSMF

A Unified Joint Matrix Factorization Framework

jNMF was extended for integrating prior networked knowledge.

Problem

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

- Network regularizers were introduced for multi-view omics data [Zhang et al., 2011]
- More generally, Joint Matrix Factorization (JMF) for data integration of multi-view data and multiple types of networks was proposed [Zhang and Zhang, 2019a]
- 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

Illustration of JMF

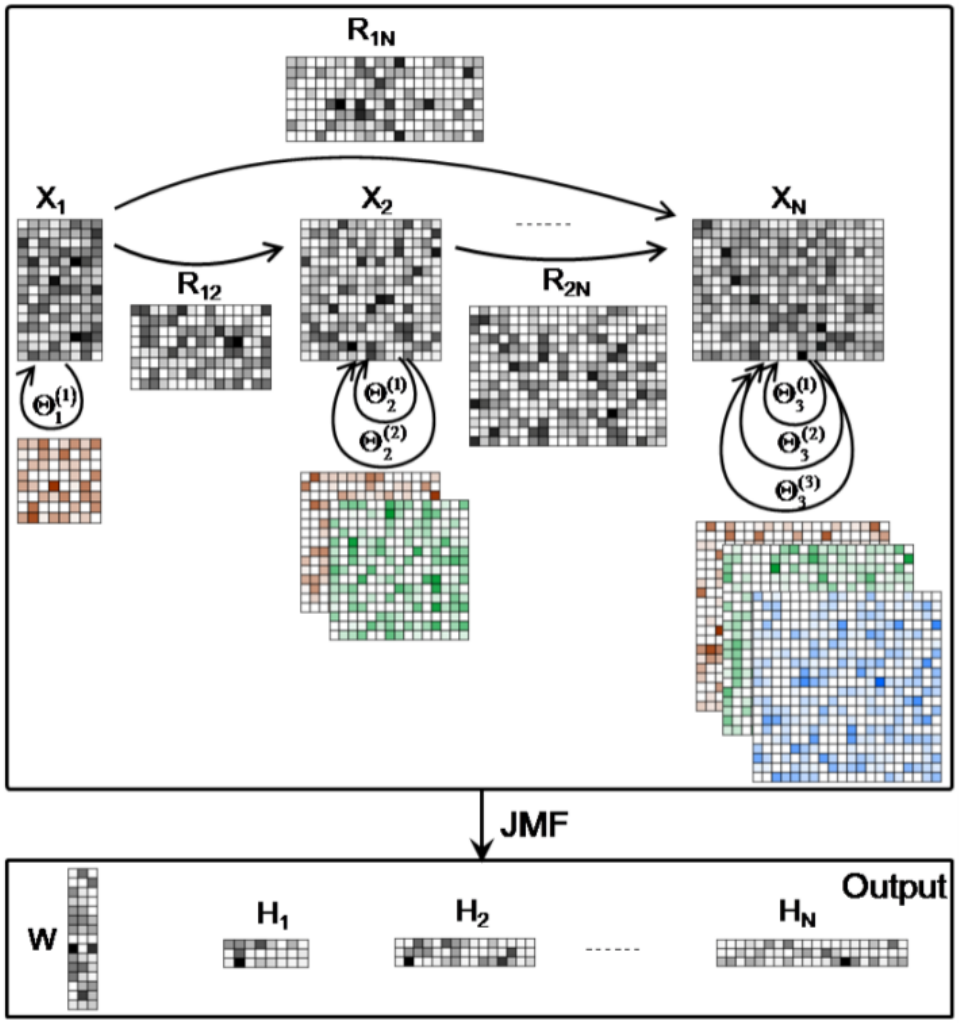


Figure: Illustration of JMF

Bayesian Joint Matrix Decomposition (BJMD)

- 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

Bayesian Joint Matrix Decomposition (BJMD)

- 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

A Bayesian Joint Matrix Decomposition model (BJMD) was proposed to remedy this shortcoming [Zhang and Zhang, 2017]

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

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

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

divergence

regularized term on W
encourage sparsity

regularized term on H
encourage h_{kj}^{c} to be $\alpha_{0k} / \sum_i \alpha_{0i}$

Complete Log Likelihood of BJMD

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

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

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

divergence

regularized term on W
encourage sparsity

regularized term on H
encourage h_{kj}^{c} to be $\alpha_{0k} / \sum_i \alpha_{0i}$

How to solve it?

- Two effective algorithms was proposed to solve it [Zhang and Zhang, 2017]

Effects of Heterogeneous Noise

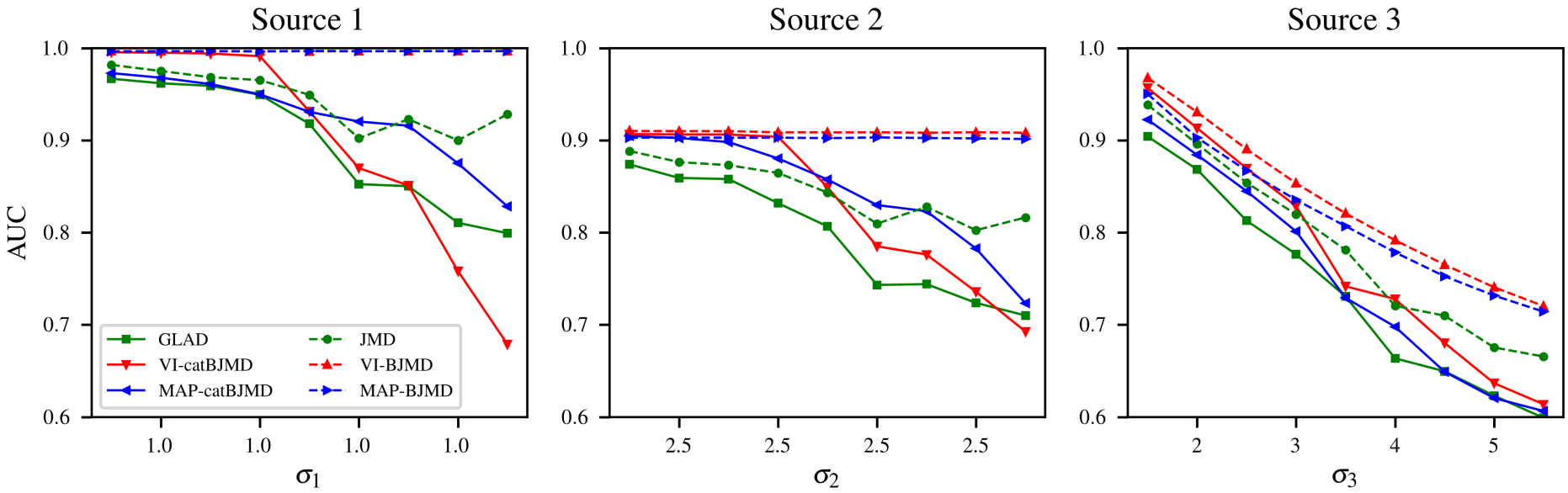


Figure: Performance comparison on simulation data.

Summary

- NMF is a powerful tool in dimension reduction and pattern recognition
- Several classical algorithms were introduced
- Exact NMF is NP-hard, but local minimum is usually good enough for applications
- 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.

Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P., and Plemmons, R. J. (2007).

Algorithms and applications for approximate nonnegative matrix factorization.

Computational statistics & data analysis, 52(1):155–173.

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., 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.

References II

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.

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

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.

References III

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.

Lin, C.-J. (2007).

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

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.

References IV

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.

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.

References V

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.

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.

Zhang, S., Liu, C.-C., Li, W., Shen, H., Laird, P. W., and Zhou, X. J. (2012).

Discovery of multi-dimensional modules by integrative analysis of cancer genomic data.
Nucleic acids research, 40(19):9379–9391.