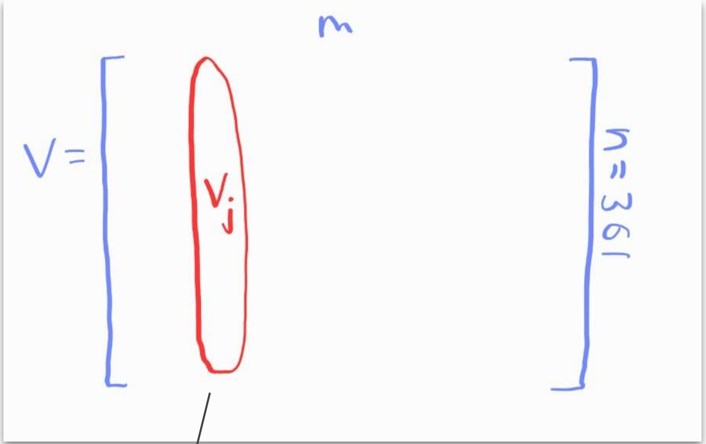
Understanding some Applications

1. NMF applied to face data

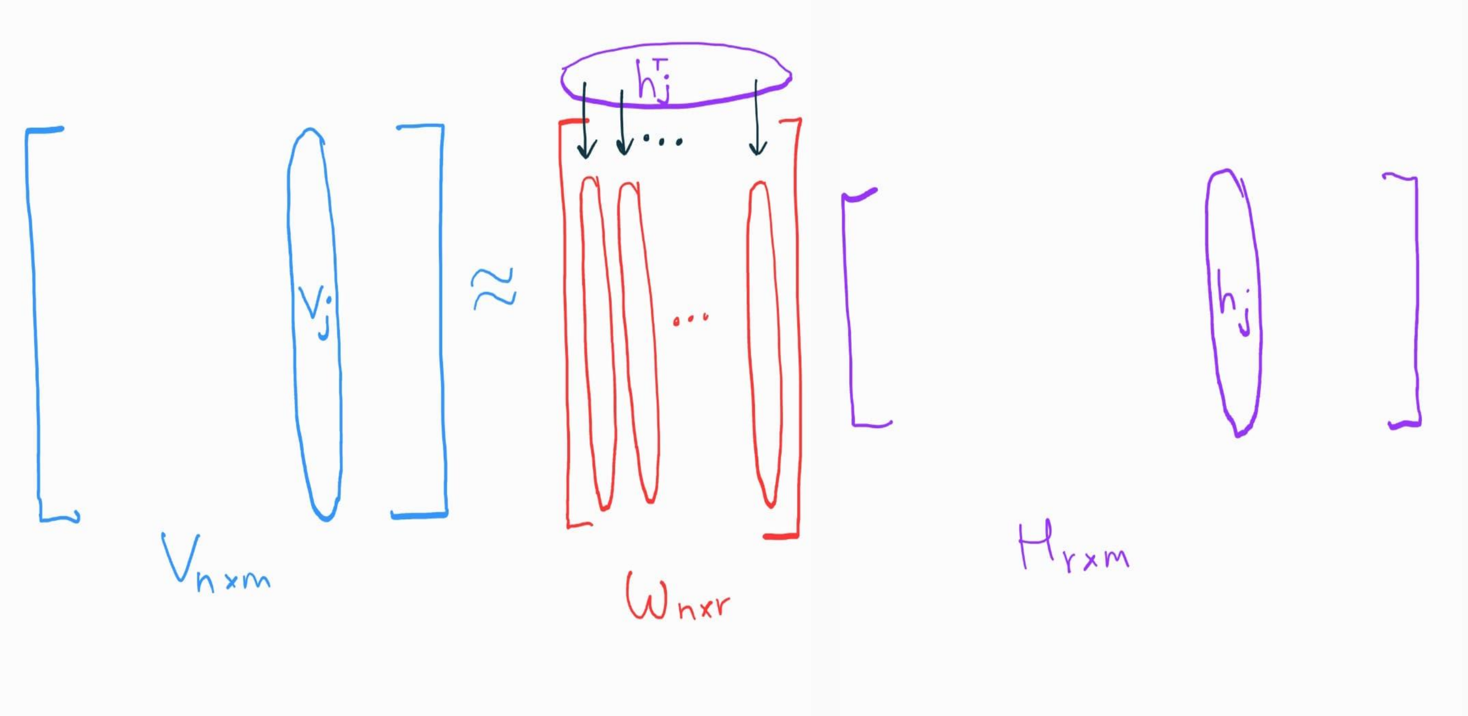
- **Dataset:** m grayscale images of dimensions 19 x 19
- **Construction of V:** V is a [361 x m] matrix. *i.e.* flatten out all the images and place them s.t. each column of V is an image vector.
- We want to decompose V_{nxm} into factors W_{nxr} and H_{rxm}
- $(n + m)*r < n*m$
- Options like NMF, PCA and VQ



j^{th} Image vector

Understanding some Applications

- All the 3 techniques try to approximate the matrix V by WH, but there are some key differences



Understanding the problem

NMF

- All the elements of W and H must be ≥ 0
- Tries to approx. each v_j by a non-negative linear combination of columns of w
- Gives a part-based representation of each face image
- W and H are sparse

Principle Component Analysis

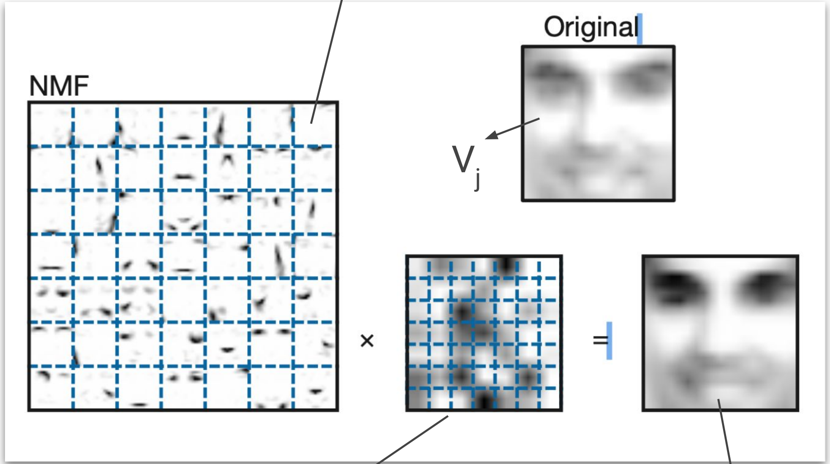
- No constraints of non-negativity on W or H
- But W must be orthonormal and rows of H orthogonal
- Tries to approx. each v_j by a linear combination of columns of w
- Gives a holistic representation of the face images
- Basis vectors containing -ve elements are non-intuitive
- So are the subtractive combinations of the basis vectors

Vector Quantization

- Constraint: Each column of H must be unary vector
- Ex: [0 0 0 1 0 0 0 0 0 0]
- Tries to approx. each v_j by one of the columns of W
- Clusters all the vectors v_j into r clusters given by w_j's
- Each basis vector is an entire face

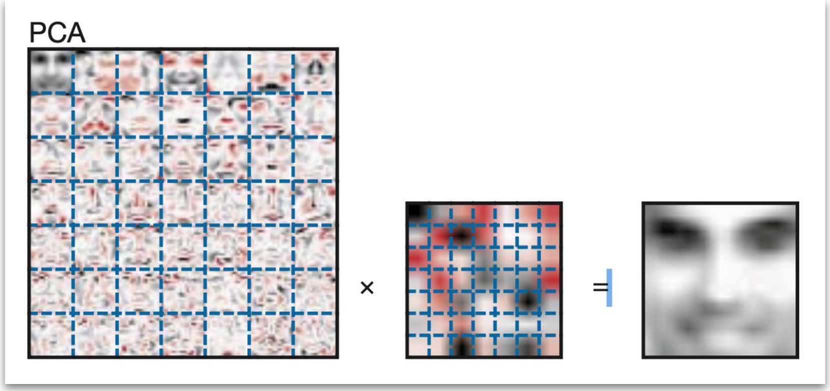
Understanding some Applications

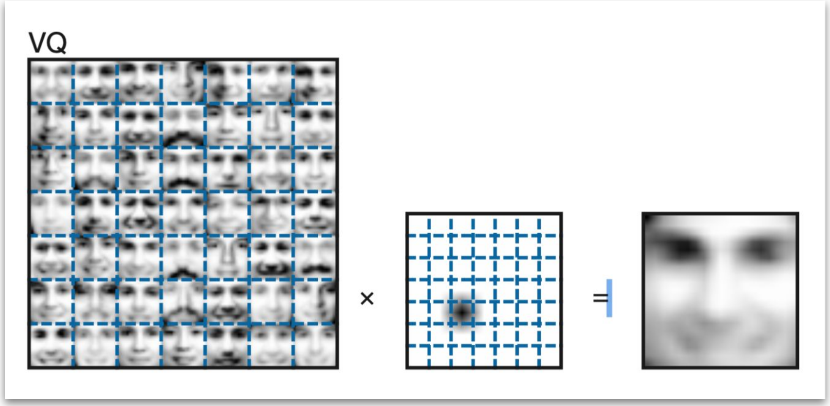
Each grid is a basis vector
i.e. $r = 49$



This entire matrix
represents the vector h_j

$(WH)_j$





letters to nature

larvae collected randomly in the field (2° 48.12′ N, 41° 40.33′ E) by SCUBA. Between 5 and 10 juveniles were recruited successfully in each of 15, 1 l polystyrene containers ($n = 15$), the bottom of which was covered with an acetate sheet that served as substratum for sponge attachment. Containers were then randomly distributed in 3 groups, and sponges in each group were reared for 14 weeks in 3 different concentrations of $Si(OH)_4$: 0.741 ± 0.133, 30.235 ± 0.287 and 100.041 ± 0.760 μM (mean ± s.e.). All cultures were prepared using 0.22 μm polycarbonate-filtered seawater, which was collected from the sponge habitat, handled according to standard methods to prevent Si contamination[29] and enriched in dissolved silica, when treatments required, by using Na_2SiF_6. During the experiment, all sponges were fed by weekly addition of 2 ml of a bacterial culture ($40–60 \times 10^6$ bacteria ml^{-1}) to each container[30]. The seawater was replaced weekly, with regeneration of initial food and $Si(OH)_4$ levels. The concentration of $Si(OH)_4$ in cultures was determined on 3 replicates of 1 ml seawater samples per container by using a Bran-Luebbe TRAACS 2000 nutrient autoanalyser. After week 5, the accidental contamination of some culture containers by diatoms rendered subsequent estimates of Si uptake by sponges unreliable, so we discarded them for the study.

For the study of the skeleton, sponges were treated according to standard methods[30] and examined in a Hitachi S-2300 scanning electron microscope (SEM).

Received 21 April; accepted 16 August 1999.

1. Hartman, W. D., Wendt, J. W. & Wiedenmayer, F. Living and fossil sponges. Notes for a short course. *Sedimentia* **8,** 1–274 (1980).
2. Ghiold, J. The sponges that spanned Europe. *New Scient.* **129,** 58–62 (1991).
3. Leinfelder, R. R. Upper Jurassic reef types and controlling factors. *Profil* **5,** 1–45 (1993).
4. Wiedenmayer, F. Contributions to the knowledge of post-Paleozoic neritic and archibental sponges (Porifera). *Schweiz. Paläont. Abh.* **116,** 1–147 (1994).
5. Lévi, C. in *Fossil and Recent Sponges* (eds Reitner, J. & Keupp, H.) 72–82 (Springer, New York, 1991).
6. Moret, L. Contribution à l'étude des spongiaires siliceux du Miocene de l'Algerie. *Mém. Soc. Géol. Fr.* **1,** 1-27 (1924).
7. Vacelet, J. Indications de profondeur donnés par les Spongiaires dans les milieux benthiques actuels. *Géol. Méditerr.* **15,** 13–26 (1988).
8. Maldonado, M. & Young, C. M. Bathymetric patterns of sponge distribution on the Bahamian slope. *Deep-Sea Res. I* **43,** 897–915 (1996).
9. Lowenstam, H. A. & Weiner, S. *On Biomineralization* (Oxford Univ., Oxford, 1989).
10. Maliva, R. G., Knoll, A. H. & Siever, R. Secular change in chert distribution: a reflection of evolving biological participation in the silica cycle. *Palaios* **4,** 519–532 (1989).
11. Nelson, D. M., Tréguer, P., Brzezinski, M. A., Leynaert, A. & Quéguiner, B. Production and dissolution

Learning the parts of objects by non-negative matrix factorization

Daniel D. Lee* & H. Sebastian Seung*†

* *Bell Laboratories, Lucent Technologies, Murray Hill, New Jersey 07974, USA*
† *Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

Is perception of the whole based on perception of its parts? There is psychological[1] and physiological[2,3] evidence for parts-based representations in the brain, and certain computational theories of object recognition rely on such representations[4,5]. But little is known about how brains or computers might learn the parts of objects. Here we demonstrate an algorithm for non-negative matrix factorization that is able to learn parts of faces and semantic features of text. This is in contrast to other methods, such as principal components analysis and vector quantization, that learn holistic, not parts-based, representations. Non-negative matrix factorization is distinguished from the other methods by its use of non-negativity constraints. These constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations. When non-negative matrix factorization is implemented as a neural network, parts-based representations emerge by virtue of two properties: the firing rates of neurons are never negative and synaptic strengths do not change sign.

The foundational paper in NMF

http://www.cs.columbia.edu/~blei/fogm/2020F/readings/LeeSeung1999.pdf

Overview of the paper

The algorithms proposed for NMF are broadly from **two schools of thought.**
Lee and Seung gave a probability based approach.

Problem : We want to find H,W such that V≅WH

Objective Function

$$F = \sum_{i=1}^{n} \sum_{\mu=1}^{m} [V_{i\mu} \log(WH)_{i\mu} - (WH)_{i\mu}]$$

Update Step

$$W_{ia} \leftarrow W_{ia} \sum_{\mu} \frac{V_{i\mu}}{(WH)_{i\mu}} H_{a\mu}$$

$$W_{ia} \leftarrow \frac{W_{ia}}{\sum_j W_{ja}}$$

$$H_{a\mu} \leftarrow H_{a\mu} \sum_i W_{ia} \frac{V_{i\mu}}{(WH)_{i\mu}}$$

Motivating the Algorithm

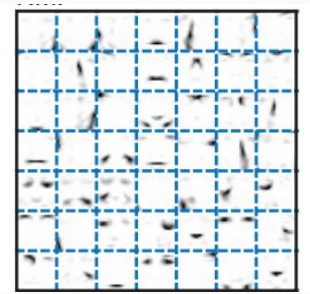
Problem : We want to find H,W such that V≅WH and W, H >=0

For images and learning a part based representation we have :

V

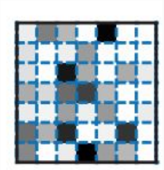
=

W



X

H



Each pixel is taken to comes from some poisson noise added to WH.

The distance measure measure used for comparing V and WH is :

Generalized Kullback-Leibler Divergence

$$D(A||B) = \sum_{ij}\left(A_{ij}\log\frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij}\right)$$

When $\sum_{ij} A_{ij} = \sum_{ij} B_{ij} = 1$

It reduces to the KL Divergence we know for probability distributions

Lee D., Seung H. S., “Algorithms for Non-negative Matrix Factorization”, Advances in Neural Information Processing, Vol. 13,

Motivating the Algorithm

Problem : We want to find H,W such that V≅WH and W, H >=0

We want to decrease this measure to get as close to V :

$$\boxed{D(A||B) = \sum_{ij}\left(A_{ij}\log\frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij}\right)}$$

For V and WH the formulation is :

$$D(V||WH) = \sum_{ij}\left(V_{ij}\log\frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij}\right)$$

$$D(V||WH) = \sum_{ij}\left(V_{ij}\log V_{ij} - V_{ij}\log(WH)_{ij} - V_{ij} + (WH)_{ij}\right)$$

function of V

Motivating the Algorithm

Problem : We want to find H,W such that V≅WH and W, H >=0

So we can ignore the V related terms

$$\min \left(\sum -v_{ij} \log (WH)_{ij} + (WH)_{ij} \right)$$

ss

$$\max \left(\sum v_{ij} \log (WH)_{ij} - (WH)_{ij} \right)$$

Final Objective

$$F = \sum_{i=1}^{n} \sum_{\mu=1}^{m} [V_{i\mu} \log(WH)_{i\mu} - (WH)_{i\mu}]$$

Max

Motivating the Algorithm

Problem : We want to find H,W such that V≅WH and W, H >=0

We want to maximise the objective
So we use the **GRADIENT DESCENT** method

But we have some problems :

1. We have to **maximise with respect to both W and H** both of which are unknown

2. We want to ensure the most important part of NMF
H >= 0
W >= 0

Final Objective

$$F = \sum_{i=1}^{n} \sum_{\mu=1}^{m} [V_{i\mu} \log(WH)_{i\mu} - (WH)_{i\mu}]$$

Constraint
H >= 0   W >= 0

Motivating the Algorithm

Problem : We want to find H,W such that V≅WH and W, H >=0

Final Objective

$$F = \sum_{i=1}^{n}\sum_{\mu=1}^{m}[V_{i\mu}\log(WH)_{i\mu} - (WH)_{i\mu}]$$

Constraint

H >= 0   W >= 0

The Objective is non convex in both (W,H)
But in each it is convex

For 1] We solve iteratively with respect to H using gradient descent and then solve using the H for W

$$H_{a\mu} \leftarrow H_{a\mu} + \boxed{\eta_{a\mu}\left[\sum_i W_{ia}\frac{V_{i\mu}}{(WH)_{i\mu}} - \sum_i W_{ia}\right]}$$

This term can become negative :(

Motivating the Algorithm

Problem : We want to find H,W such that V≅WH and W, H >=0

For 2]

We **use a specific step size** to remove any kind of subtractive term and maintain the non-negative constraint

$$\eta_{a\mu} = \frac{H_{a\mu}}{\sum_i W_{ia}},$$

Final Objective

$$F = \sum_{i=1}^{n}\sum_{\mu=1}^{m}[V_{i\mu}\log(WH)_{i\mu} - (WH)_{i\mu}]$$

Constraint

H >= 0 W >= 0

The GD derived term

$$H_{a\mu} \leftarrow H_{a\mu} + \eta_{a\mu}\left[\sum_i W_{ia}\frac{V_{i\mu}}{(WH)_{i\mu}} - \sum_i W_{ia}\right]$$

Motivating the Algorithm

Problem : We want to find H,W such that V≅WH and W, H >=0

$$H_{a\mu} = H_{a\mu} + \frac{H_{a\mu}}{\left(\sum_i W_{ai}\right)}\left(\left(X\right) - \sum W_{ai}\right)$$

$$H_{a\mu} = \cancel{H_{a\mu}} - \cancel{H_{a\mu}} + \frac{H_{a\mu}}{\sum_i W_{ai}} \cdot X$$

(Chumay!)

$$H_{a\mu} = \frac{H_{a\mu}(X)}{\sum_i W_{ai}}$$

$$\boxed{H_{a\mu} = H_{a\mu} \cdot \frac{1}{\sum W_{ia}}\left(\sum_i \frac{W_{ia}\cdot V_{i\mu}}{(WH)_{i\mu}}\right)}$$

update rule

And we do the same thing for W

Final Objective

$$F = \sum_{i=1}^{n}\sum_{\mu=1}^{m}\left[V_{i\mu}\log(WH)_{i\mu} - (WH)_{i\mu}\right]$$

Constraint

H >= 0 W >= 0

Step size in GD

$$\eta_{a\mu} = \frac{H_{a\mu}}{\sum_i W_{ia}},$$

The GD derived term

$$H_{a\mu} \leftarrow H_{a\mu} + \eta_{a\mu}\left[\sum_i W_{ia}\frac{V_{i\mu}}{(WH)_{i\mu}} - \sum_i W_{ia}\right]$$

Motivating the Algorithm

Problem : We want to find H,W such that V≅WH and W, H >=0

Update Rules

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} V_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}}$$

$$W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} V_{i\mu}/(WH)_{i\mu}}{\sum_\nu H_{a\nu}}$$

Final Objective

$$F = \sum_{i=1}^{n} \sum_{\mu=1}^{m} [V_{i\mu} \log(WH)_{i\mu} - (WH)_{i\mu}]$$

Constraint

H >= 0   W >= 0

Step size in GD

$$\eta_{a\mu} = \frac{H_{a\mu}}{\sum_i W_{ia}},$$

Motivating the Algorithm

Problem : We want to find H,W such that V≅WH and W, H >=0

Algorithm

H, W random initialise
while not converged :

Update H
Update W

end while

Update Rules

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} V_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}}$$

$$W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} V_{i\mu}/(WH)_{i\mu}}{\sum_\nu H_{a\nu}}$$

Final Objective

$$F = \sum_{i=1}^{n} \sum_{\mu=1}^{m} [V_{i\mu} \log(WH)_{i\mu} - (WH)_{i\mu}]$$

Constraint

H >= 0 W >= 0

Step size in GD

$$\eta_{a\mu} = \frac{H_{a\mu}}{\sum_i W_{ia}},$$

Convergence of the Algorithm

Convergence Proof :

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

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

are satisfied.

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

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

Lemma 2 : Here **we actually have a G** which is **proved to be an auxiliary function for F where F is KLD from which we obtained our objective**

Convergence of the Algorithm

What we want to show to say the algorithm converges?

Our update step $\longrightarrow$ Makes update such that with each update $\boxed{\text{KLD is nonincreasing}}$

$H \leftarrow \text{update}(H)$

$W \leftarrow \text{update}(W)$

if we have a G

s.t. $\begin{cases} G(h, h^t) \geq F(h) \\ \text{and} \\ G(h, h) = F(h) \end{cases}$

(KLD term)

then using a

$$h^{t+1} = \underset{h}{\operatorname{argmin}}\ G(h, h^t)$$

$\downarrow$

It is a therorem that F(h) will non increasing when h^{t+1} is the update

Comparison: Frobenius vs KLD

Problem : We want to find H,W such that V≅WH and W, H >=0

We mentioned in the start that there are 2 schools of thought we present it below :

Objective Function

KLD

$$F = \sum_{i=1}^{n} \sum_{\mu=1}^{m} [V_{i\mu} \log(WH)_{i\mu} - (WH)_{i\mu}]$$

- Probabilistic way of looking at the NMF

Frobenius Norm

$$\min_{W \geq 0, H \geq 0} f(W,H) = \|A - WH\|_F^2.$$

- Pure linear algebra way of going about the NMF

The Objective is different!!

Comparison: Frobenius vs KLD

Problem : We want to find H,W such that V≅WH and W, H >=0

Algorithm

KLD

Theorem 2 *The divergence $D(V\|WH)$ is nonincreasing under the update rules*

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} V_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}} \qquad W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} V_{i\mu}/(WH)_{i\mu}}{\sum_\nu H_{a\nu}} \tag{5}$$

The divergence is invariant under these updates if and only if W and H are at a stationary point of the divergence.

Frobenius

Algorithm 1 The BCD framework for solving NMF: $\min_{W,H\geq 0} \|A - WH\|_F^2$

1: Input: Matrix $A \in \mathbb{R}^{m\times n}$, tolerance parameter $0 < \varepsilon << 1$, upper limit of the number of iterations T
2: Initialize H
3: **repeat**
4: Obtain the optimal solution of subproblem (8a)
5: Obtain the optimal solution of subproblem (8b)
6: **until** A particular stopping criterion based on W, H, ε is satisfied *or* the number of iterations reaches upper limit T
7: Output: W, H

$$W \leftarrow \arg\min_{W\geq 0} f(W,H), \tag{7a}$$

$$H \leftarrow \arg\min_{H\geq 0} f(W,H). \tag{7b}$$

These subproblems can be written as

$$\min_{W\geq 0} \|H^T W^T - A^T\|_F^2, \tag{8a}$$

$$\min_{H\geq 0} \|WH - A\|_F^2. \tag{8b}$$

Comparison: Frobenius vs KLD

Problem : We want to find H,W such that V≅WH and W, H >=0

Discussion on some more point

KLD

Monotonic convergence can be proven using techniques similar to those used in proving the convergence of the EM algorithm
We do not have the overhead of finding the step-size as we have to do it for the frobenius algo.

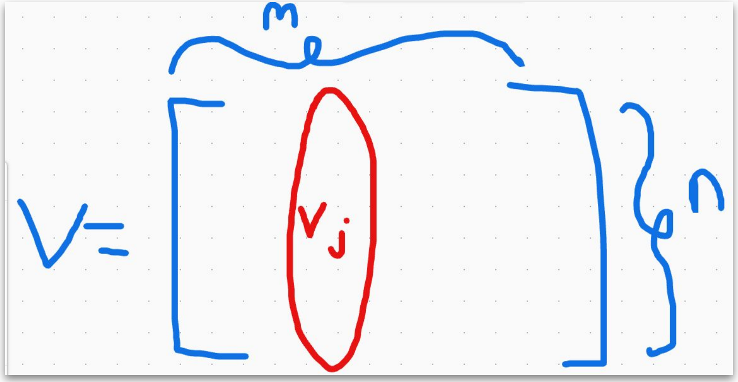
Frobenius

Algorithms for solving NMF with KL-divergence are typically much slower than those for solving NMF based on the Frobenius norm

Understanding some Applications

NMF applied to Text data (Training)

- **Dataset:** m documents with a fixed vocabulary of size n.
- **Construction of V:** V is a [n x m] matrix. *i.e.* column v_j is a representation of the j^{th} document.
- Preprocessing changes V.
- We decompose V into factors W and H using NMF algorithm



Preprocessing

NEED to reduce the vocabulary size

- Tokenization
- Lower Casing
- Stemming
- Removing non-essentials.
    - Punctuations
    - Numbers
    - Stop words
    - Single Character

Before

‘In the new system “Canton becomes Guangzhou and Tientsin becomes Tianjin.” Most importantly, the newspaper would now refer to the country’s capital as Beijing, not Peking. This was a step too far for some American publications. In an article on Pinyin around this time, the Chicago Tribune said that while it would be adopting the system for most Chinese words, some names had “become so ingrained’

After

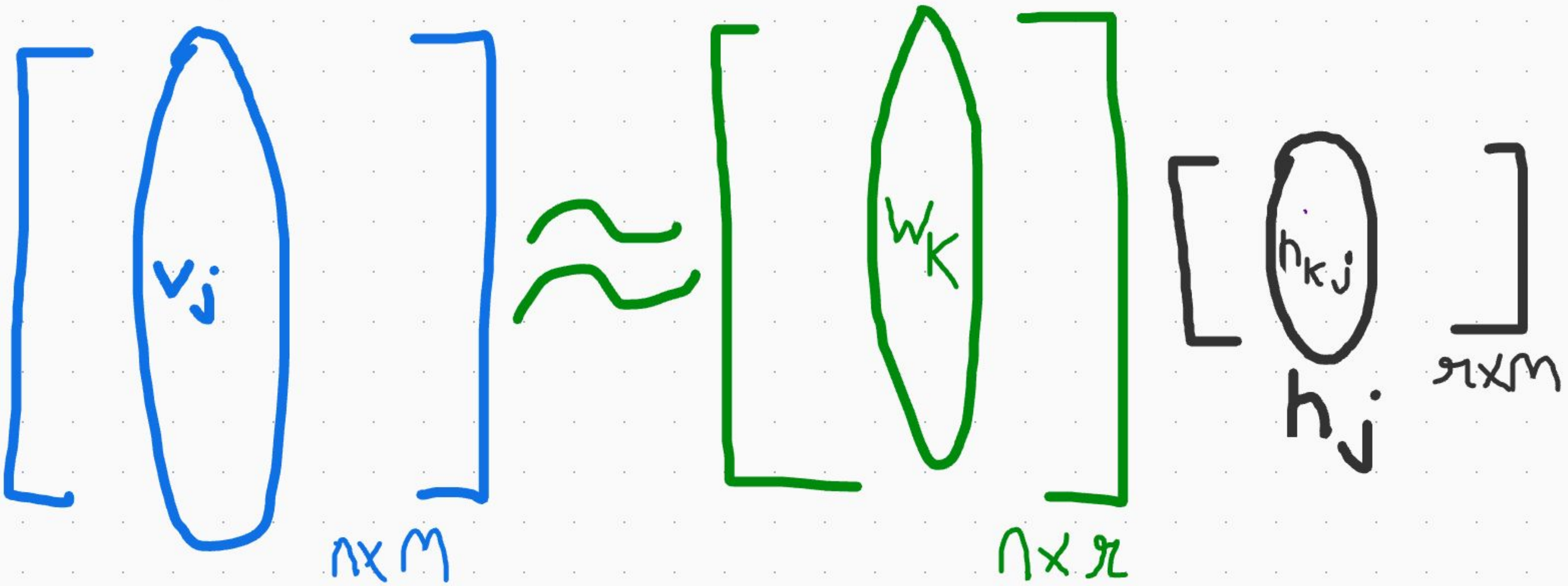
‘new canton becom guangzhou tientsin becom tianjin import newspap refer countri capit beij peke step far american public articl pinyin time chicago tribun adopt chines word becom ingrain‘

Preprocessing

Constructing V.

- Converting Text into numbers.
  - Bag of words
  - Tf-idf
  - Word vectors
- Feature Selection
  - Remove words with extreme documents count.
  - Choose the best feature set

Understanding some Applications



Understanding some Applications

NMF applied to Text data (Prediction)

- We have the W Matrix.
- **Construction of v:** Use the same preprocessing to construct a column vector v from the document.
- Solve for h!!

Voila You Have your Prediction !!!!

$V \approx \boxed{W} H$ (Train)

$W h = v$ Find h?(Predict)

Comparison with LDA and SVD

What is LDA ?

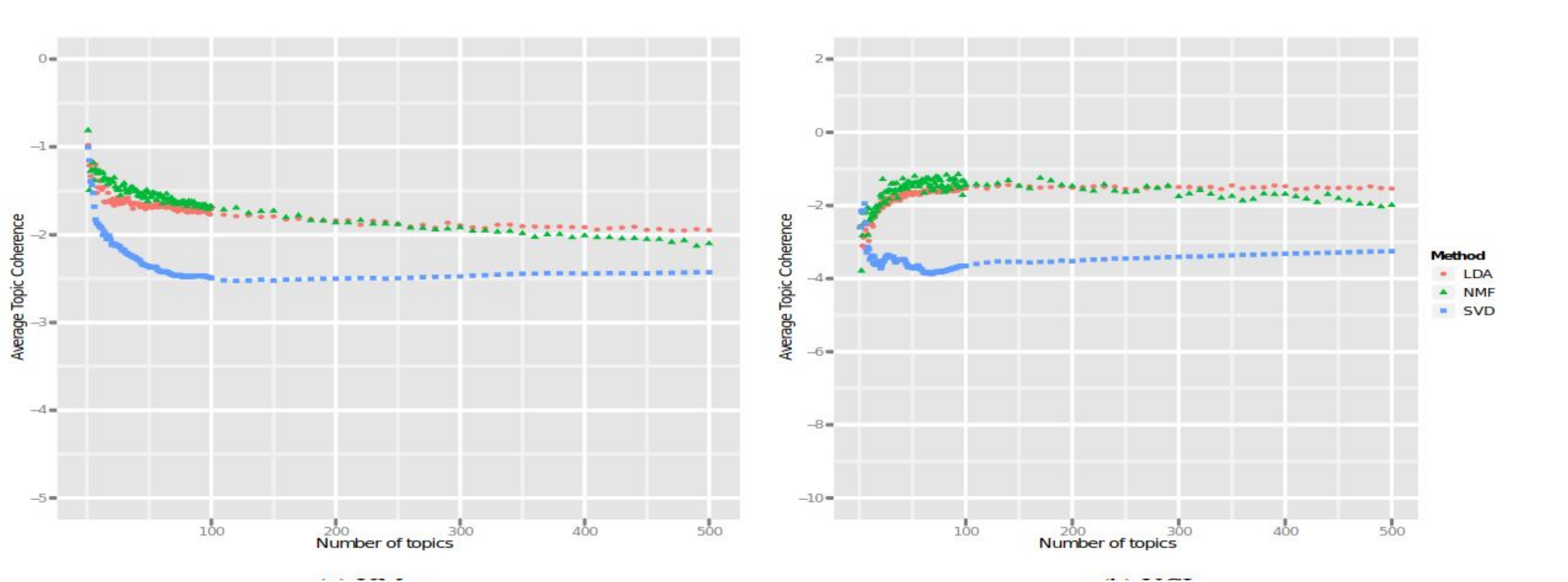
1. Choose $\Theta_i \sim Dir(\alpha)$, a topic distribution for D_i
2. For each word $w_j \in D_i$:
   (a) Select a topic $z_j \sim \Theta_i$
   (b) Select the word $w_j \sim \Phi_{z_j}$

What is SVD

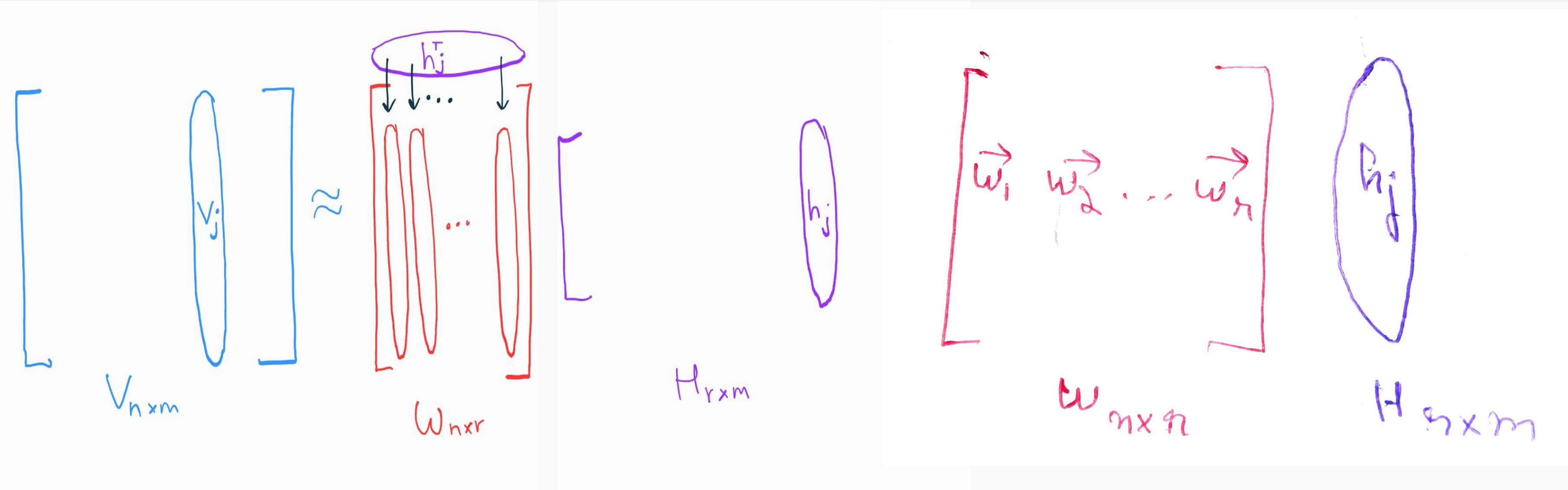
$$M = U\Sigma V^T$$

Model	Label	Top Words	UMass	UCI
High Quality Topics				
LDA	interview	told asked wanted interview people made thought time called knew	-2.52	1.29
	wine	wine wines bottle grapes made winery cabernet grape pinot red	-1.97	1.30
NMF	grilling	grilled sweet spicy fried pork dish shrimp menu dishes sauce	-1.01	1.98
	cloning	embryonic cloned embryo human research stem embryos cell cloning cells	-1.84	1.46
SVD	cooking	sauce food restaurant water oil salt chicken pepper wine cup	-1.87	-1.21
	stocks	fund funds investors weapons stocks mutual stock movie film show	-2.30	-1.88

Coherence Scores



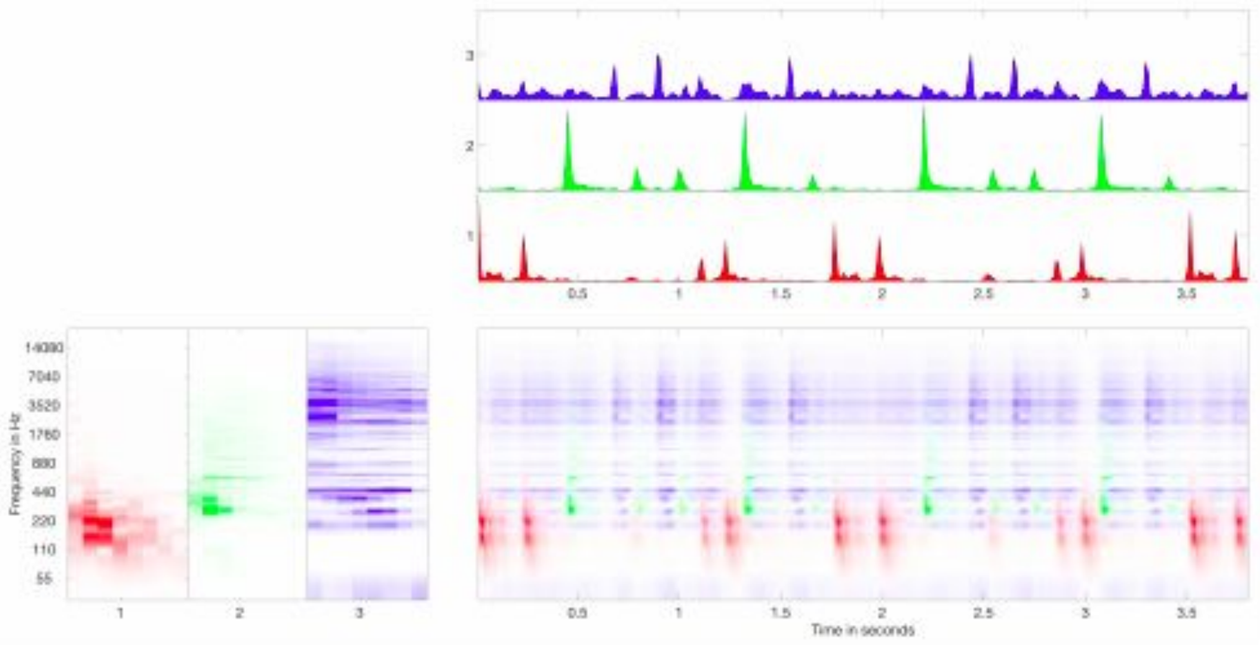
An interesting view of the factorization



$$v_j \approx \vec{w}_1 h_{1j} + \vec{w}_2 h_{2j} + \dots + \vec{w}_r h_{rj}$$

$$\frac{v_j}{\sum_i h_{ij}} \approx \vec{w}_1 c_1 + \vec{w}_2 c_2 + \dots + \vec{w}_r c_r$$

Separating Sound signals using NMF



Separating Sound signals using NMF

$$\begin{bmatrix} x_{1t} \\ x_{2t} \\ \vdots \\ x_{Ft} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{F1} & a_{F2} \end{bmatrix} \bullet \begin{bmatrix} s_{1t} \\ s_{2t} \end{bmatrix}$$

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t$$

$$\mathbf{x}_t = \mathbf{a}_1 s_{1t} + \mathbf{a}_2 s_{2t}$$

Thank You !