Matrix Factorization

EE698V - Machine Learning for Signal Processing

Vipul Arora

References (highly recommended)

- Lee, Daniel D., and H. Sebastian Seung, 'Learning the Parts of Objects by Non-Negative Matrix Factorization', *Nature*, 401 (1999), 788–91
- Lee, D D, and H S Seung, 'Algorithms for Non-Negative Matrix Factorization', in *Proc. Conf. Adv. Neural Info. Process. Syst. (NIPS)* (MIT Press, 2001), XIII, 556–62

Matrix Factorization

1.00	0.0	1.00	0.00	0.00	1.00	0.00	0.0
0.75	0.0	0.75	0.25	0.25	0.75	0.25	0.0
0.50	0.0	0.50	0.50	0.50	0.50	0.50	0.0
0.25	0.0	0.25	0.75	0.75	0.25	0.75	0.0
0.00	0.0	0.00	1.00	1.00	0.00	1.00	0.0

Matrix Factorization

1.00	0.0	1.00	0.00	0.00	1.00	0.00	0.0
0.75	0.0	0.75	0.25	0.25	0.75	0.25	0.0
0.50	0.0	0.50	0.50	0.50	0.50	0.50	0.0
0.25	0.0	0.25	0.75	0.75	0.25	0.75	0.0
0.00	0.0	0.00	1.00	1.00	0.00	1.00	0.0

V

$V = WH$

=

0.00	1.00
0.25	0.75
0.50	0.50
0.75	0.25
1.00	0.00

W
(dictionary)

x

0	0	0	1	1	0	1	0
1	0	1	0	0	1	0	0

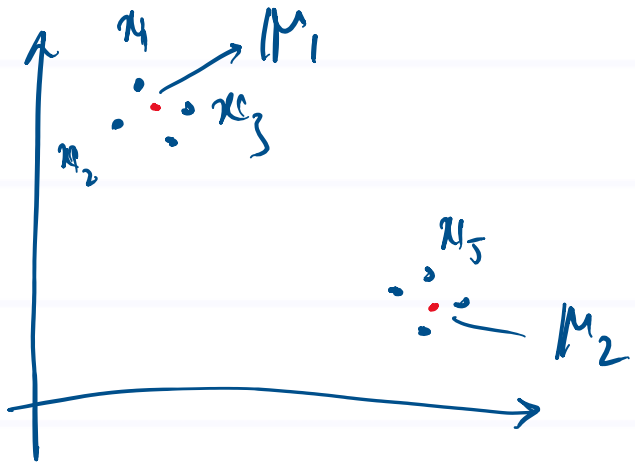
H
(activations)

Matrix Factorization: Advantages

- Reduction in size:
  - $\boldsymbol{V}$: (M, N)
  - $\boldsymbol{W}$: (M, R)
  - $\boldsymbol{H}$: (R, N)
  - If R is small, then MR+RN << MN
- Useful for analysis
  - Dimension reduction
  - Classification
- Useful for reconstruction

K-means or Vector Quantization

1.22	0.20	1.00	0.00	0.00	1.05	0.00	0.00
0.83	0.04	0.67	0.35	0.31	0.66	0.34	0.11
0.42	0.04	0.60	0.60	0.60	0.41	0.63	0.00
0.39	0.03	0.45	0.73	0.84	0.21	0.78	0.07
0.20	0.08	0.01	0.69	1.06	0.07	1.11	0.00



- Winner takes all

$x_1 \rightarrow \mu_1$

$x_2 \rightarrow \mu_1$

$x_5 \rightarrow \mu_2$

K-means or Vector Quantization

1.22	0.20	1.00	0.00	0.00	1.05	0.00	0.00
0.83	0.04	0.67	0.35	0.31	0.66	0.34	0.11
0.42	0.04	0.60	0.60	0.60	0.41	0.63	0.00
0.39	0.03	0.45	0.73	0.84	0.21	0.78	0.07
0.20	0.08	0.01	0.69	1.06	0.07	1.11	0.00

=

0.00	1.00
0.25	0.75
0.50	0.50
0.75	0.25
1.00	0.00

W

x

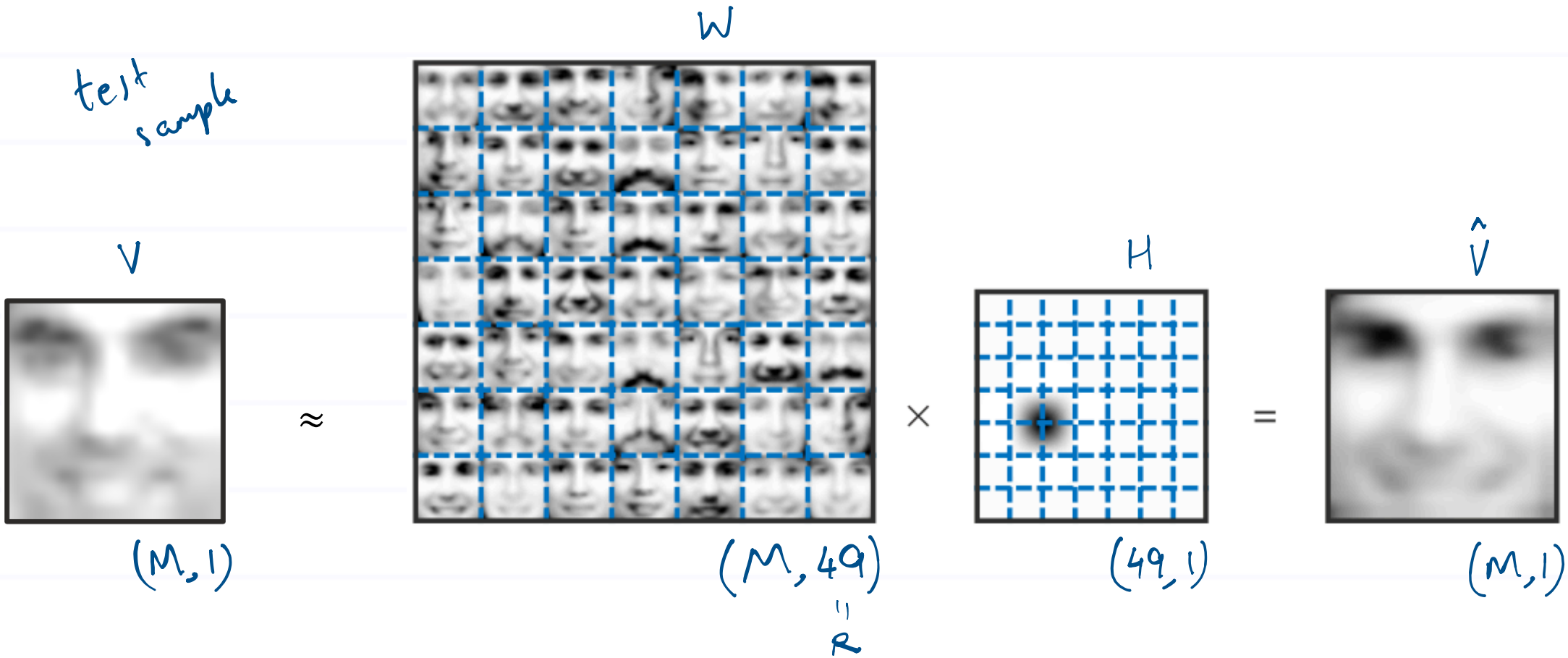
0	0	0	1	1	0	1	0
1	0	1	0	0	1	0	0

H

H → S

Vector Quantization: Example

learnt from 1600 samples



PCA as Matrix Factorization

dict activ.

$$y_n = U^T x_n \quad \forall\ n$$

$$U = [u_1\ u_2 \cdots\ u_M]$$

(eigenvecs of $S = \frac{1}{N}\sum_n x_n x_n^T$)

In matrix form: $[y_1\ y_2 \cdots\ y_N] = U^T [x_1\ x_2 \cdots\ x_N]$

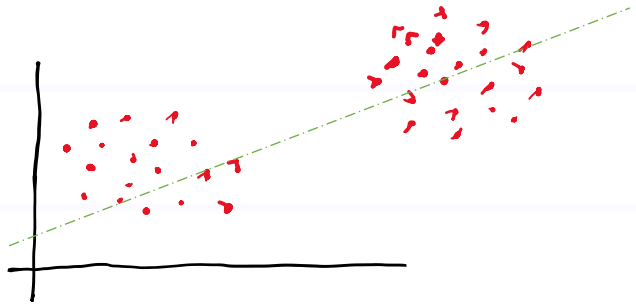
$$Y = U^T X$$

$M \times N$ $\quad$ $M \times 784$ $\quad$ $784 \times N$

- This is a way of matrix factorization
- Here, Y, U, X can have any real values
- But for images, negative values do not make sense.
- Can we constrain U to have only non-negative values? Then Y will also be non-negative

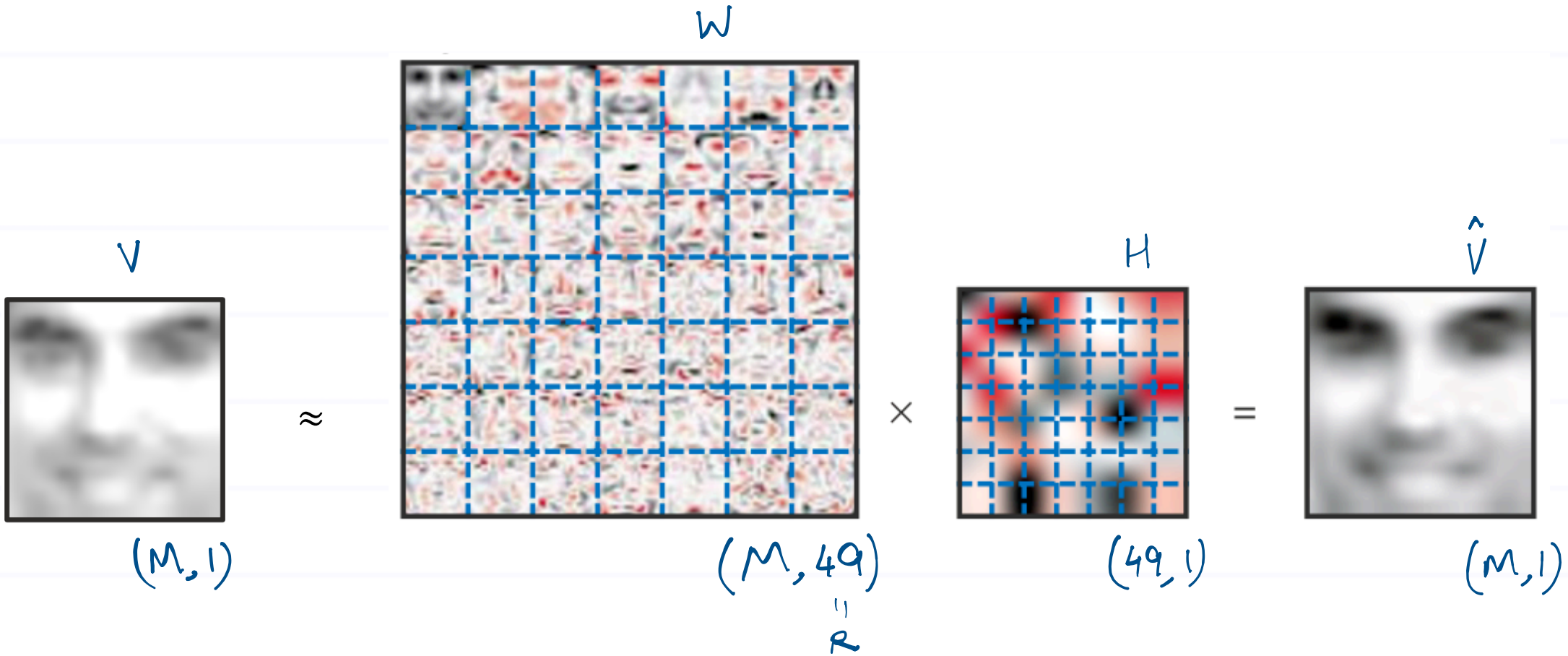
all u_i are orthonormal

PCA as Matrix Factorization



- Dimensionality reduction: R eigen vectors

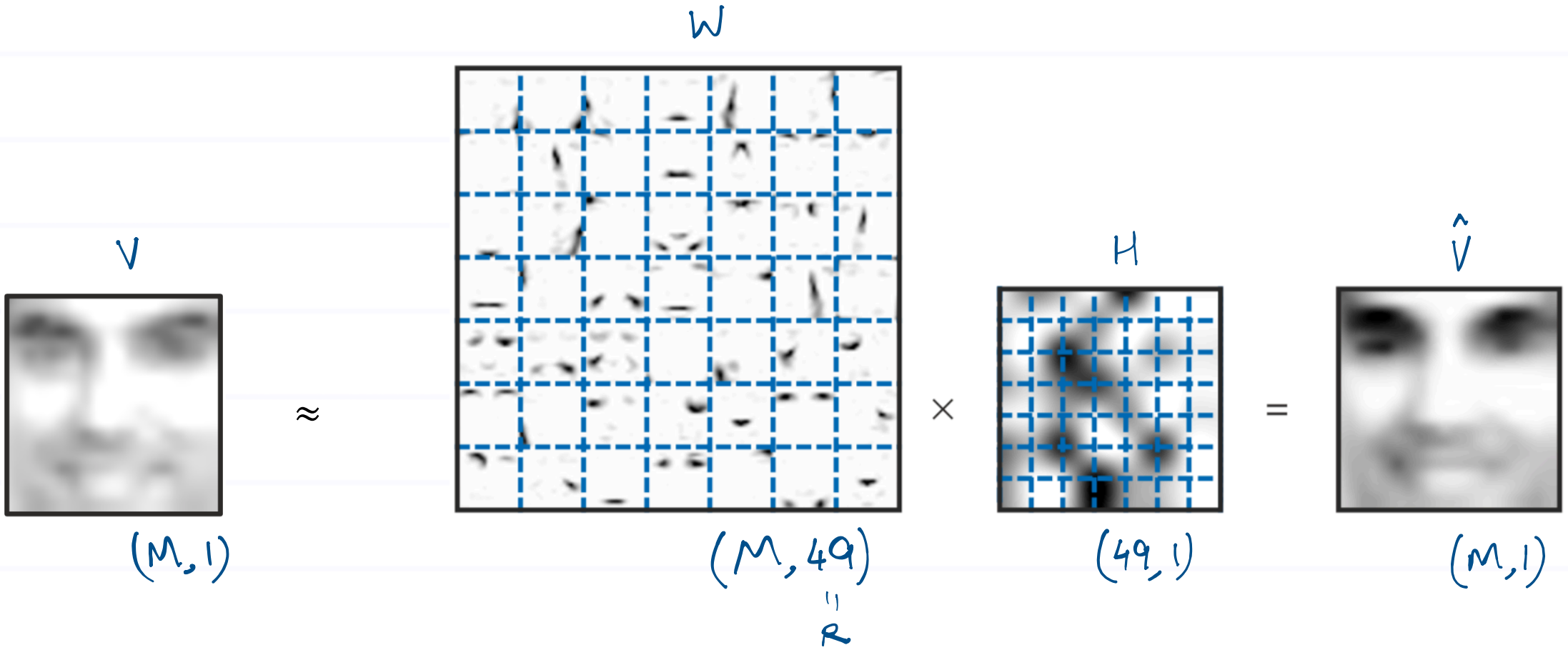
PCA: Example



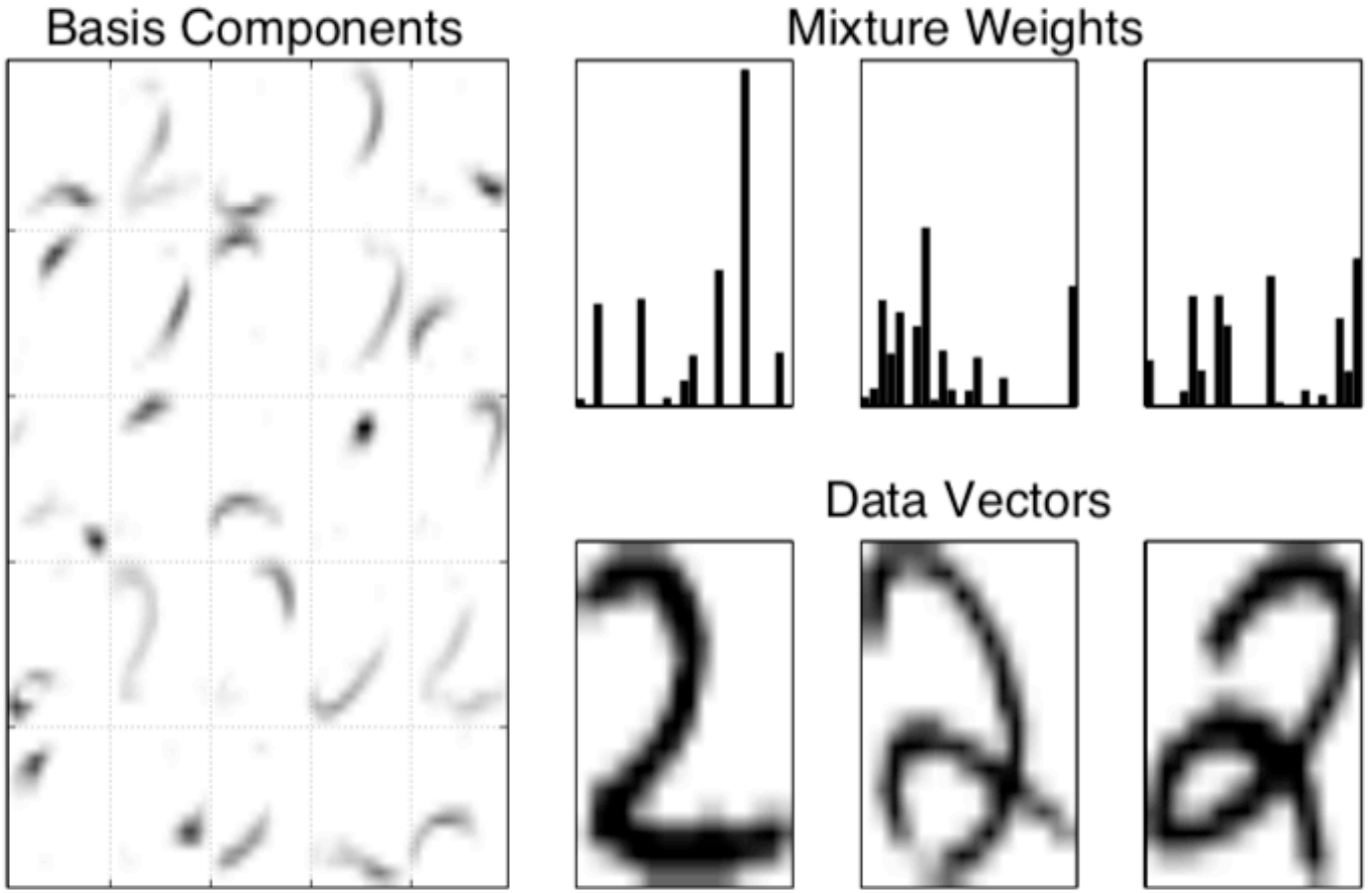
Non Negative Matrix Factorization (NMF)

- Many times matrices are non-negative
  - Images
  - Spectral magnitudes
  - Energies
  - Counts (NLP)
- Non-negative dictionary and activations give interpretability and dexterity; dictionary learns different parts
- $\boldsymbol{V} \geq 0, \boldsymbol{W} \geq 0, \boldsymbol{H} \geq 0$

NMF: Example



NMF: Another Example



NMF

- V is known
- W, H are unknowns
- Estimate them in tandem
  - Given V, initialize W
  - Estimate H, given W
  - Estimate W, given H
  - Iterate until convergence

NMF: parameter estimation

$$E = |V - \hat{V}|^2$$

$$= |V - WH|^2 \qquad \text{constraint: } W \geq 0, \; H \geq 0$$

$$\text{or} \quad E = \sum_{i,j} \left| V_{ij} - \sum_r W_{ir} H_{rj} \right|^2 \quad ; \quad W_{ir} \geq 0, H_{rj} \geq 0 \quad \forall i, r, j$$

To optimize:

- $\frac{\partial E}{\partial W (\text{or } H)} = 0$
- grad. desc.

NMF: parameter estimation

- $W_{ir} \leftarrow W_{ir} \frac{(VH^T)_{ir}}{(WHH^T)_{ir}}$

- $H_{rj} \leftarrow H_{rj} \frac{(W^T V)_{rj}}{(W^T WH)_{rj}}$

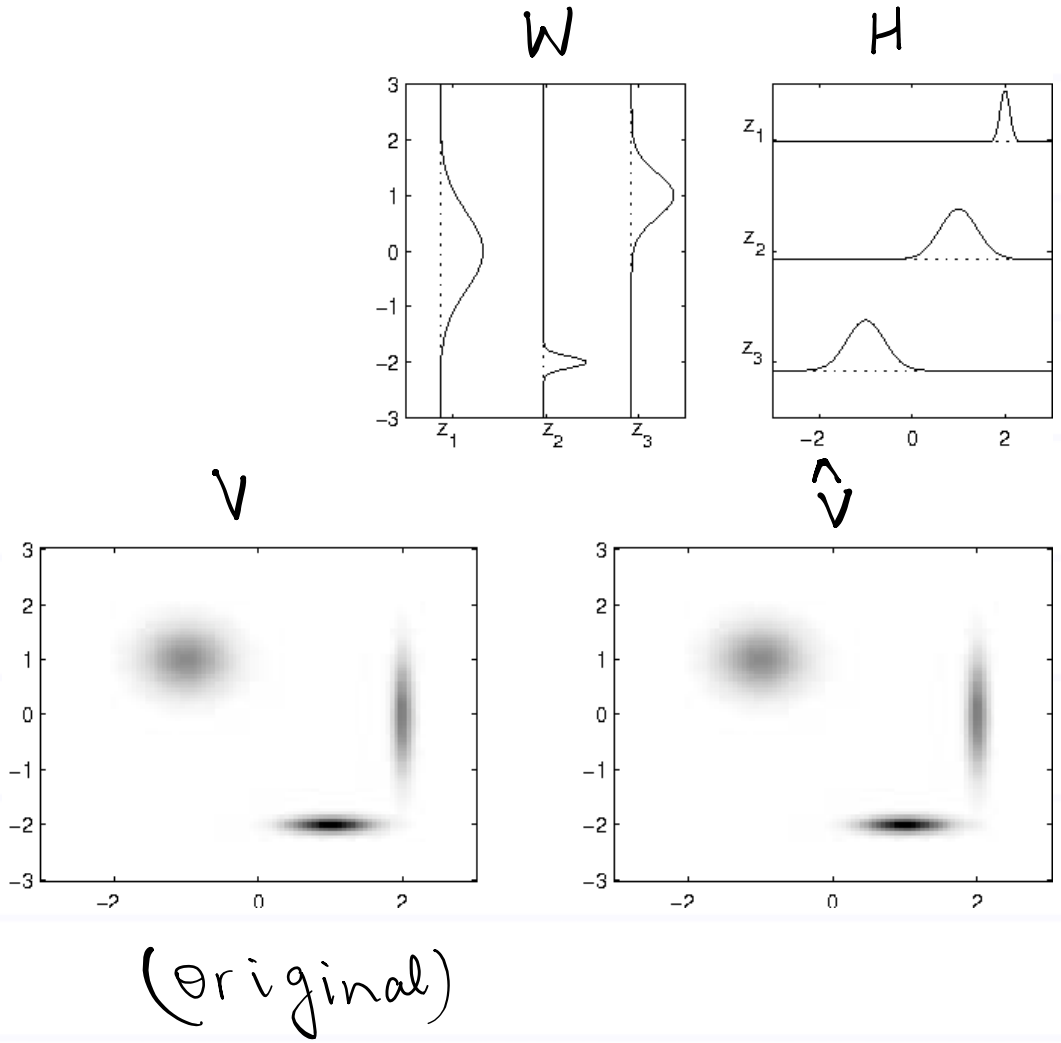
- See derivation in Reference: Lee, D D, and H S Seung, 'Algorithms for Non-Negative Matrix Factorization', in *Proc. Conf. Adv. Neural Info. Process. Syst. (NIPS)* (MIT Press, 2001), XIII, 556–62

Probabilistic Latent Component Analysis (PLCA):
one kind of NMF

Reference:

Shashanka, M, ‘Latent Variable Framework for Modeling and Separating Single Channel Acoustic Sources’ (Boston University, 2007) **Section 3.3.2**

PLCA



(original)

$V \approx WH$

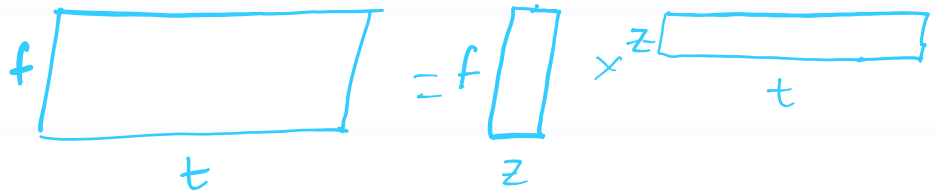
$= \hat{V}$

(approx.)

PLCA

like a histogram

Given V_{ft}



$P(f|z)$ $P(z|t)$

Let $V_{ft} \sim P(f|t)$

We want to approximate $P(f|t)$ with PLCA using latent variable z (discrete r.v.)

$$P(f|t) = \sum_z P(z|t) \underbrace{P(f|z,t)}_{= P(f|z)}$$

independent of t

PLCA

(t is not a r.v. here)

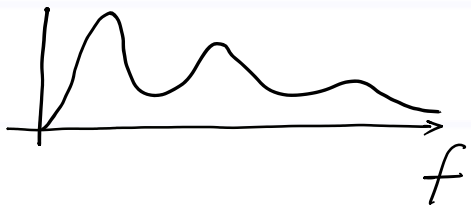
$$V_{ft} \sim P_t(f) = \sum_z P(f|z) \, P_t(z)$$

PLCA

- AIM: Given V_{ft}, we want to estimate the underlying distribution with the help of latent variables
- Analogy:
  - underlying clusters
  - underlying classes
  - parts

Maximum likelihood Estimation

v_{ft} histogram



given t

$$\text{likelihood} = \prod_i P_t(f_i)$$

$$\log \text{likelihood} = \sum_i \log P_t(f_i)$$

$$= \sum_f v_{ft} \log P_t(f) \qquad \because \text{counts} \propto v_{ft}$$

f_i & f

MLE

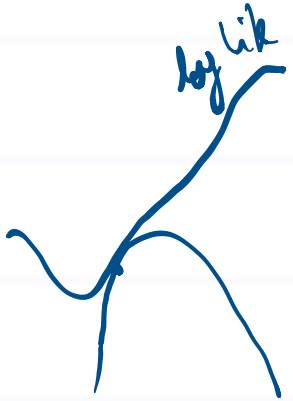
Over all t,

$$\text{log likelihood} = \sum_{f,t} V_{ft} \log P_t(f)$$

$$= \sum_{f,t} V_{ft} \log \left[\sum_z P_t(z) P(f|z) \right]$$

MLE

Over all t,



$$\text{log likelihood} = \sum_{f,t} V_{ft} \log P_t(f)$$

$$= \sum_{f,t} V_{ft} \log \left[\sum_z \frac{P_t(z)\, P(f|z)}{q_t(z|f)} \times q_t(z|f) \right]$$

pdf

$$\geq \sum_{f,t} V_{ft} \sum_z q_t(z|f) \log \left[\frac{P_t(z)\, P(f|z)}{q_t(z|f)} \right]$$

$$\therefore \text{Aux. fn.}, \quad \mathcal{L}(q, \theta) = \sum_{f,t} V_{ft} \sum_z q_t(z|f) \log \left[\frac{P_t(z)\, P(f|z)}{q_t(z|f)} \right]$$

Recall: EM Algorithm

- E-step
  - $q \leftarrow \text{argmin}_q \ \text{KL}\left[q(Z)||p(Z|X,\theta)\right]$ $q(z) = p(z|x,\theta)$
- M-step
  - $\theta \leftarrow \text{argmax}_\theta \ \mathbb{E}_q[\log p(X,Z|\theta)]$ analytically solvable

EM Algorithm

E-step: $q_t(z|f) = \frac{P_t(z)\, P(f|z)}{\sum_z P_t(z)\, P(f|z)}$

M-step: $\theta = \arg\max_\theta \mathcal{L}(q, \theta)$

M-step

$$\mathcal{L}(q, \theta) = \sum_{f,t} V_{ft} \sum_{z} q_t(z|f) \log\left[\frac{P_t(z)\, P(f|z)}{q_t(z|f)}\right]$$

θ includes $\underbrace{P_t(z)}_{H}$ and $\underbrace{P(f|z)}_{W}$

Constraints:

$$\sum_{z} P_t(z) = 1 \quad \& \quad \sum_{f} P(f|z) = 1$$

M-step

$$\mathcal{L}(q,\theta) = \sum_{f,t} V_{ft} \sum_{z} q_t(z|f) \log\left(P_t(z)\, P(f|z)\right) + \sum_t \lambda_t \left(1 - \sum_z P_t(z)\right)$$

$$+ \sum_z \lambda'_z \left(1 - \sum_f P(f|z)\right)$$

$$\frac{\partial \mathcal{L}}{\partial P_{t'}(z')} = \sum_f V_{ft'}\, q_{t'}(z'|f) \cdot \frac{1}{P_{t'}(z')} + \lambda_{t'}(-1) = 0$$

$$P_{t'}(z') = \frac{\sum_f V_{ft'}\, q_{t'}(z'|f)}{\lambda_{t'}}$$

$$\Rightarrow \sum_f V_{ft'}\, q_{t'}(z'|f) = \lambda_{t'}\, P_{t'}(z')$$

$$\underbrace{P_{t'}(z')}_{1}$$

Summing over z' gives $\lambda_{t'}$

M-step

$$\therefore \quad p_t(z) = \frac{\sum_f v_{ft}\, q_t(z|f)}{\sum_{z'} \sum_f v_{ft}\, q_t(z'|f)}$$

Similarly,

$$p(f|z) = \frac{\sum_t v_{ft}\, q_t(z|f)}{\sum_f \sum_t v_{ft}\, q_t(z|f)}$$