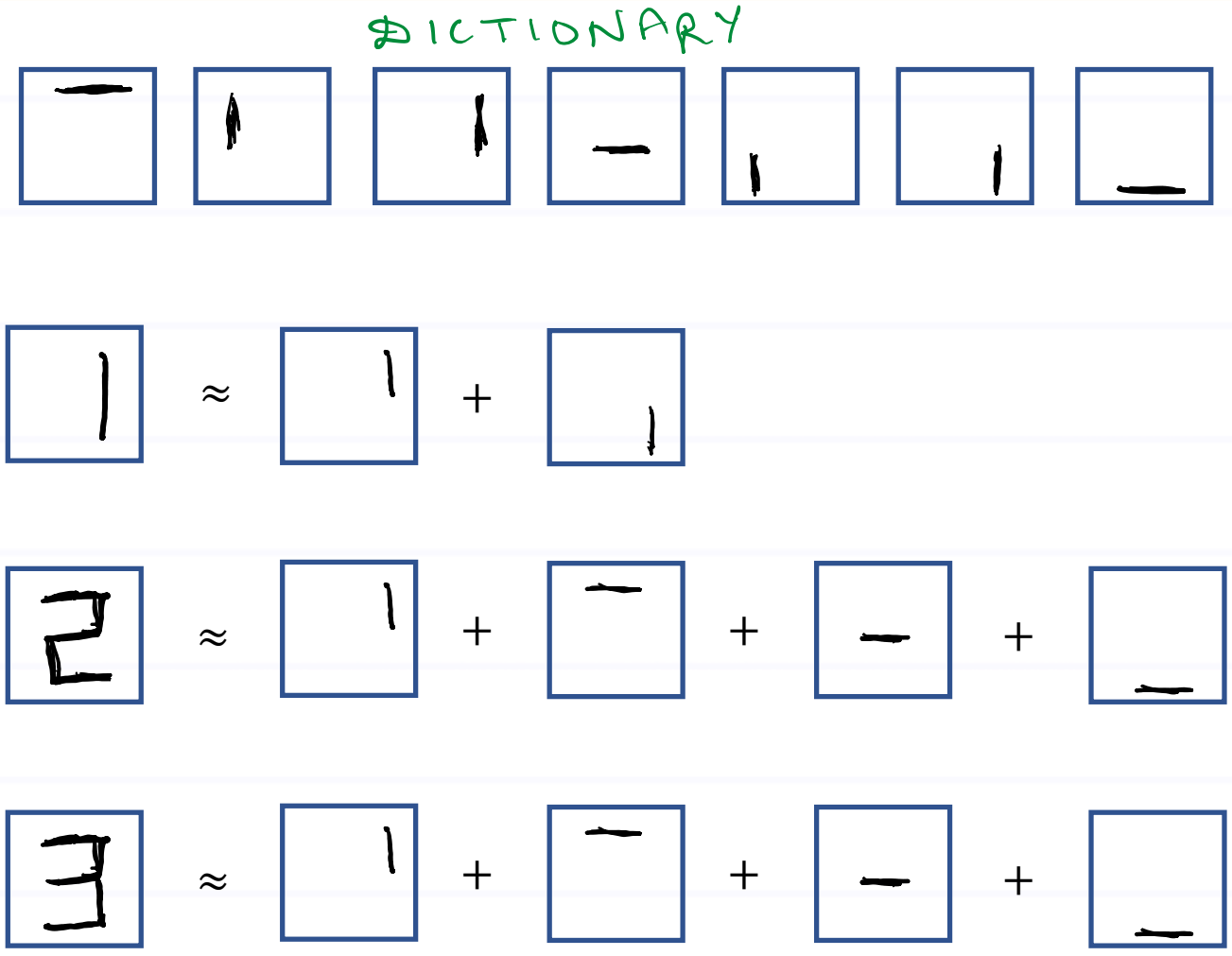
Non-negative Matrix Factorization III

EE698V - Machine Learning for Signal Processing

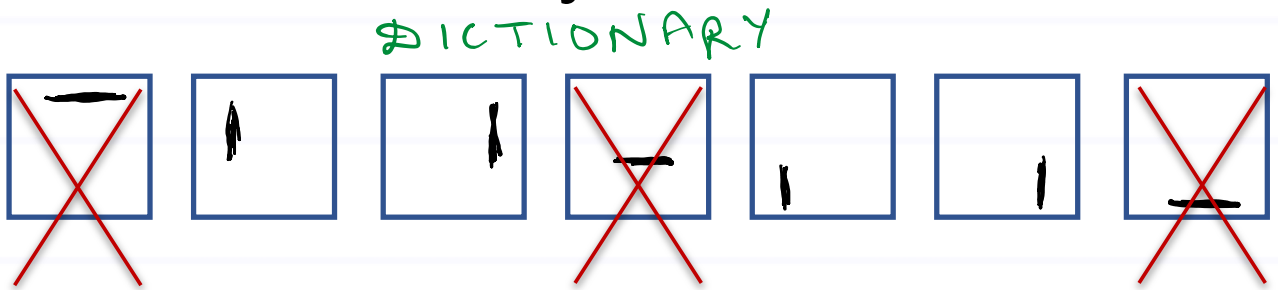
Vipul Arora

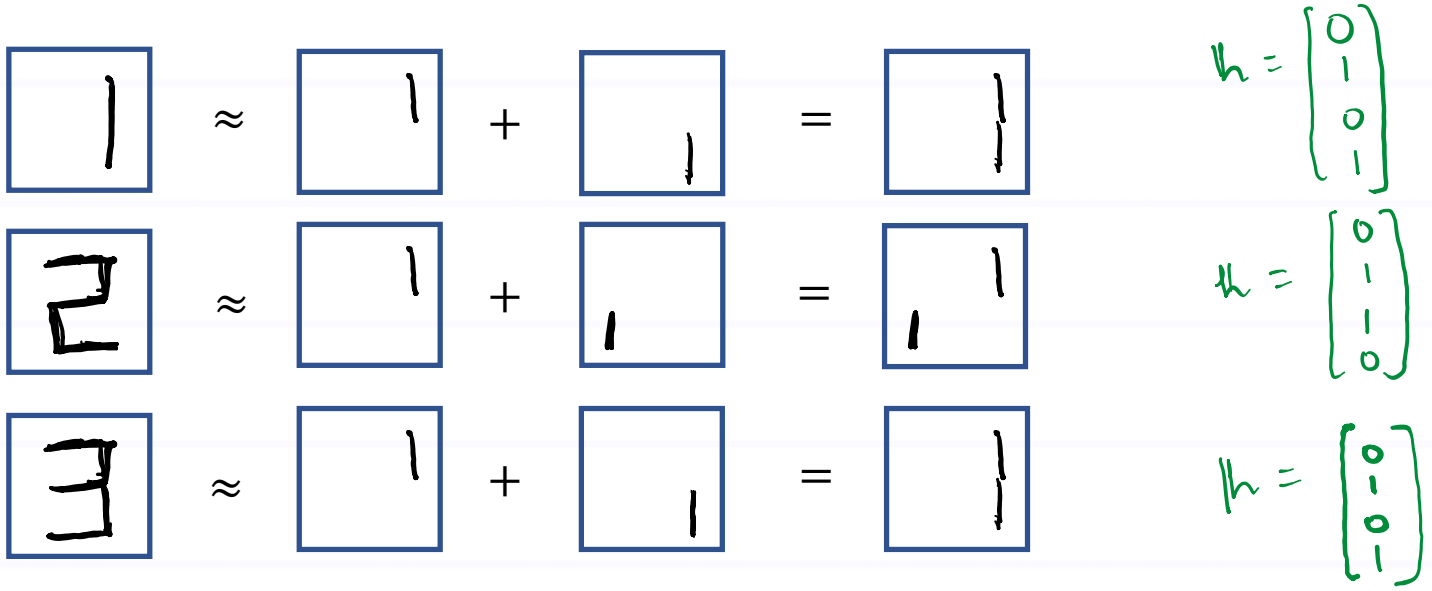
NMF: for Reconstruction



NMF: for Controlled Reconstruction

- Construct only the vertical lines of digits

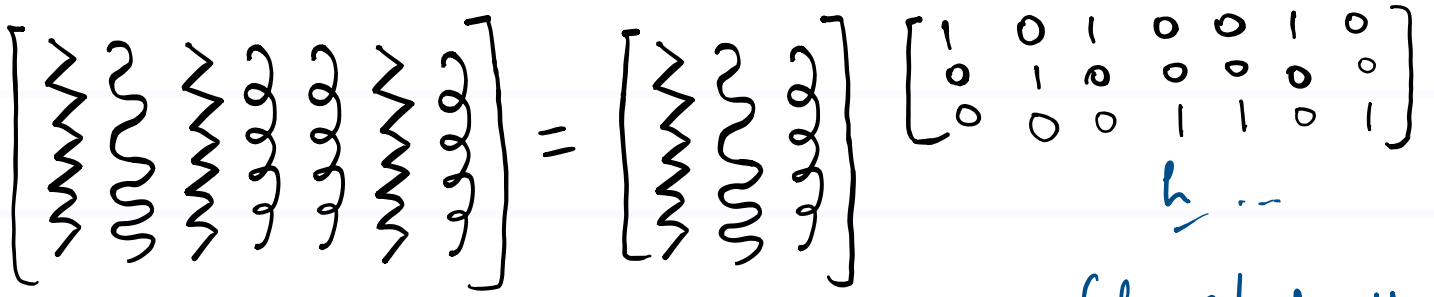




NMF: for Clustering

- Similar $\boldsymbol{v}$ have similar activations

$$V = WH$$

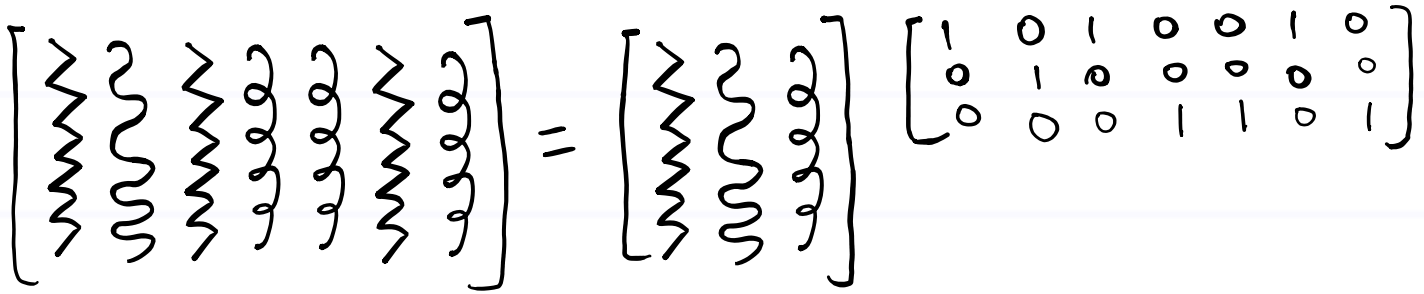


Cluster your H with std. clustering algo.

NMF: for Dimensionality Reduction

- Learn W from a huge database

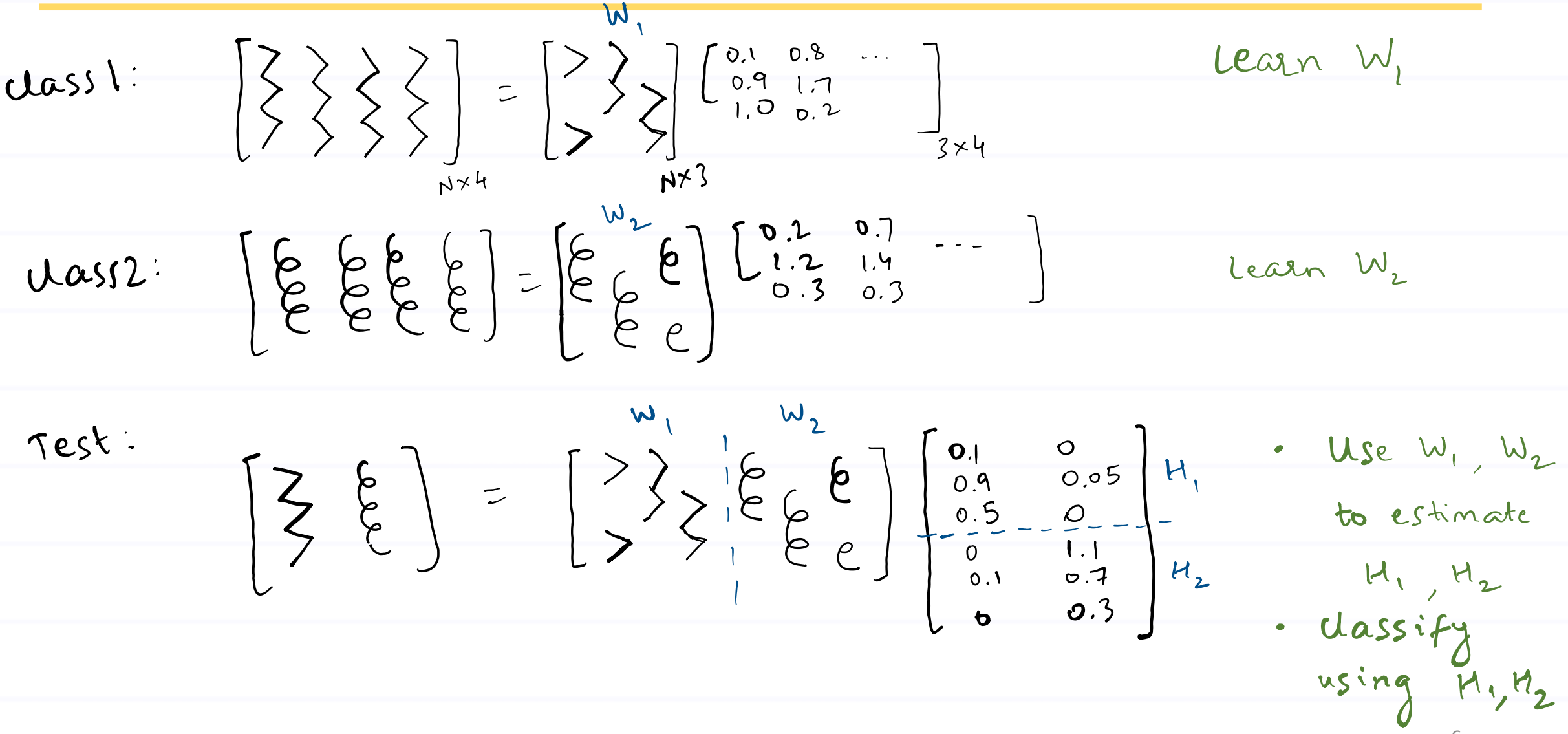
$$V = WH$$



- Use W to convert a new vector $\boldsymbol{v}$ into $\boldsymbol{h}$

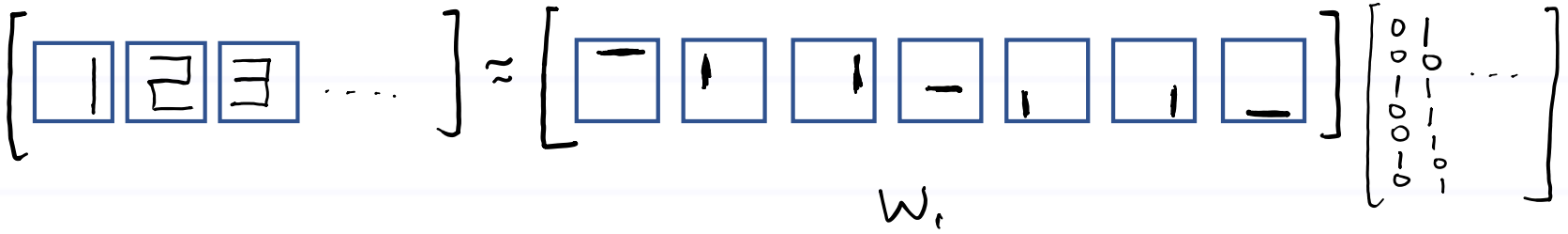
$$\boldsymbol{v} = W\boldsymbol{h}$$

NMF: for Classification



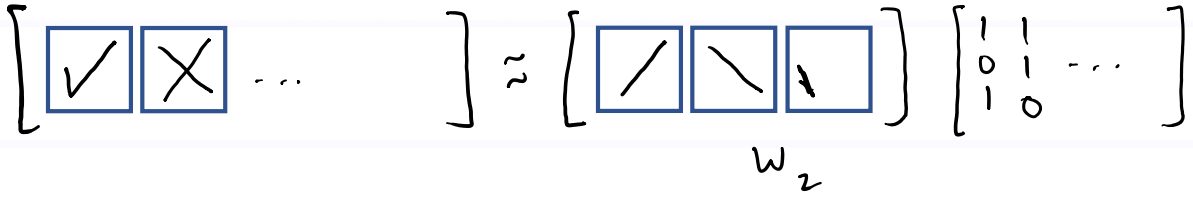
NMF: for Separation

- Linearly added vectors from different classes can be separated



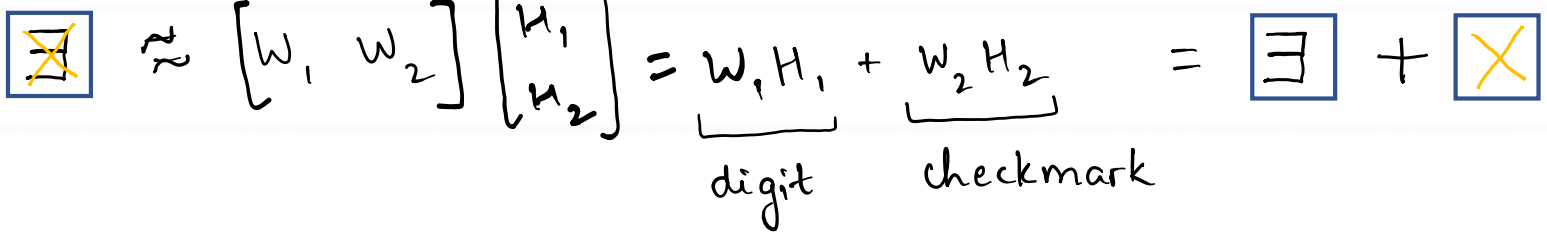
W_1

· Learn W_1



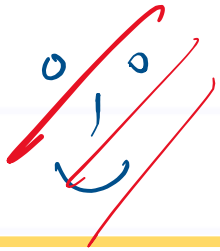
W_2

· Learn W_2

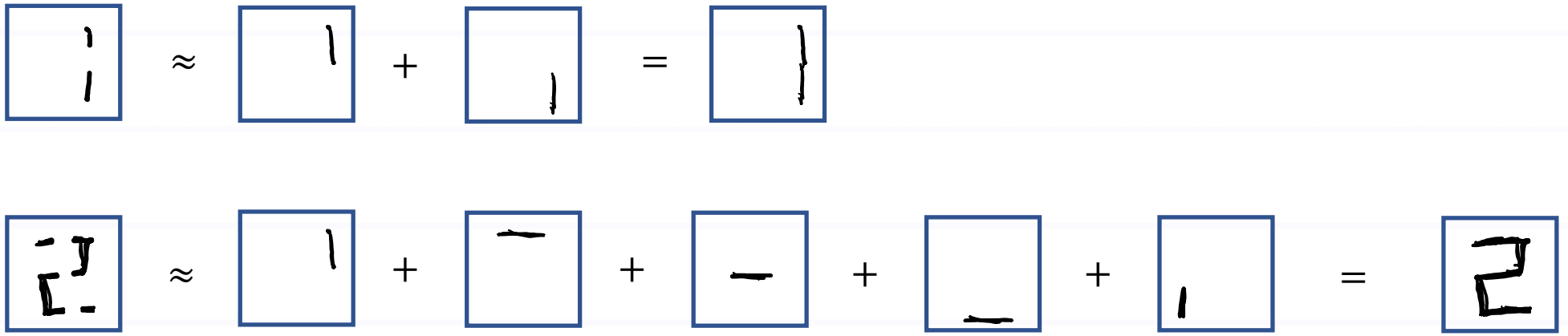


· use $[W_1, W_2]$ to estimate H_1, H_2 & reconstruct

NMF: for In-painting



- Missing parts could be refilled or reconstructed using dictionary learnt beforehand

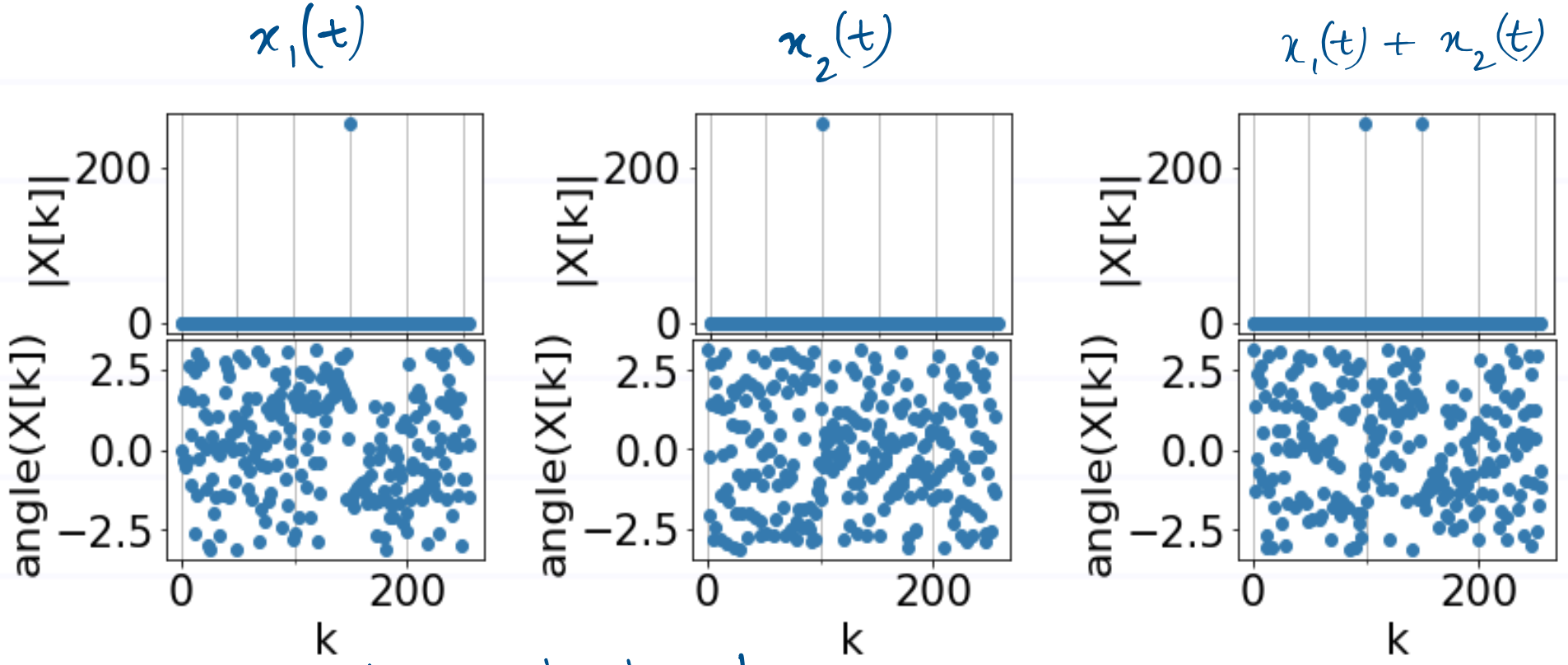


Time Series

$$X[k] \propto \sum_n \sin\left(2\pi f \frac{n}{F_s}\right) e^{-j 2\pi k \frac{n}{N}}$$

$$\frac{f}{F_s} = \frac{k}{N}$$

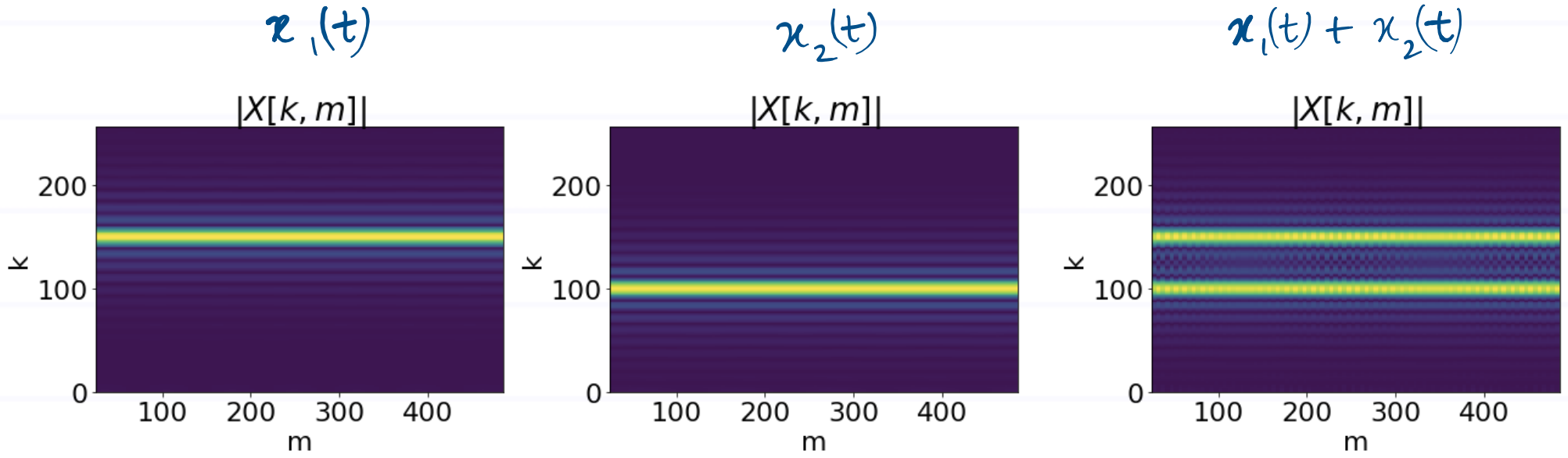
- Linear addition of periodic signals



Notice that: $|X_1[k] + X_2[k]| \approx |X_1[k]| + |X_2[k]|$

Time Series

- Note that $|X[k]|$ can be assumed to add linearly
- Same is observed with $|X[k,m]|$ as well



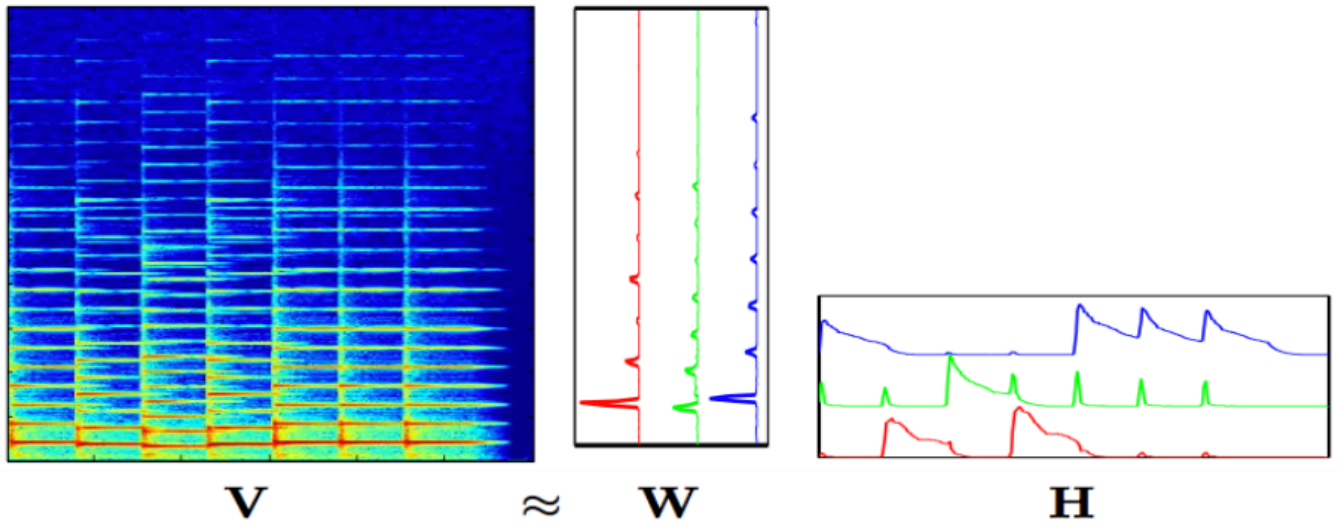
NMF for time series

- We can use this assumption to use NMF for time series analysis
- A very popular application is audio source separation

NMF for Audio

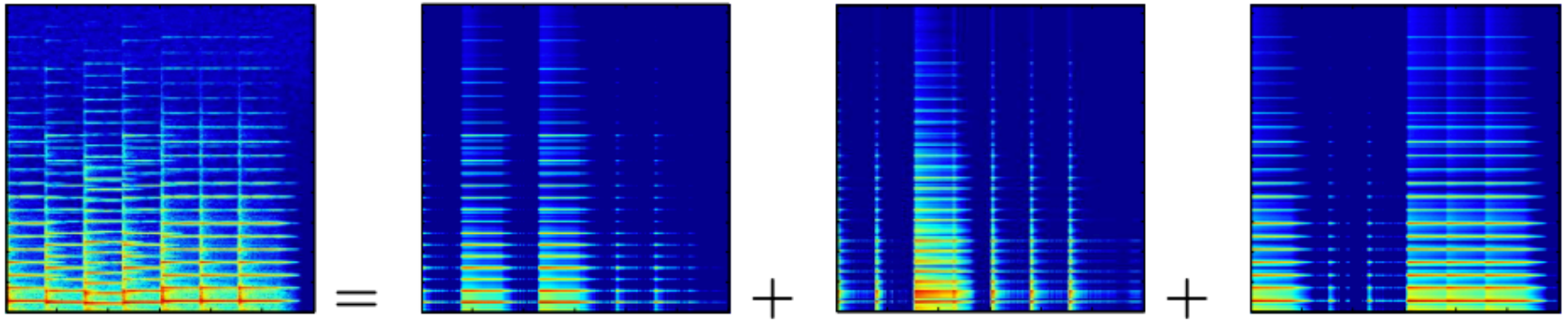
- Given an audio recording $x[n]$ sampled at F_s
- Obtain $|X[k,m]|$

NMF on STFT: |X[k]| as a feature vector



Courtesy: https://ccrma.stanford.edu/~njb/teaching/sstutorial/part2.pdf

Reconstruct each pitch separately



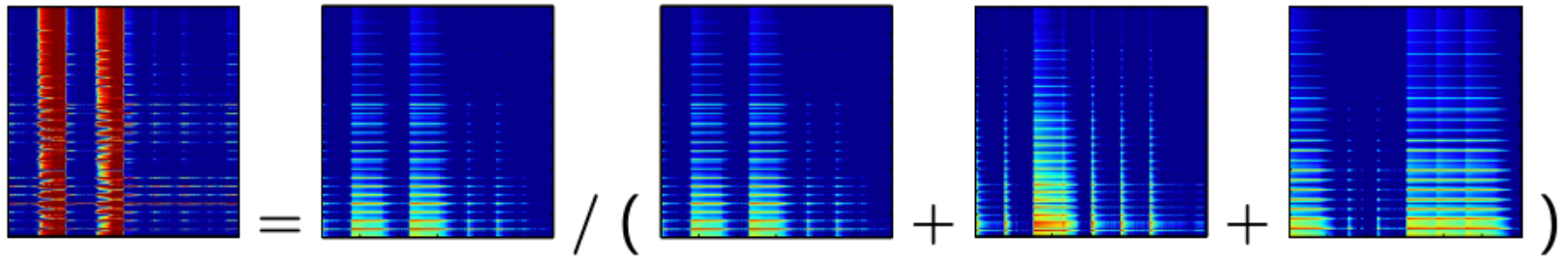
$$\mathbf{V} \approx \mathbf{w}_1 \mathbf{h}_1^{\mathrm{T}} + \mathbf{w}_2 \mathbf{h}_2^{\mathrm{T}} + \mathbf{w}_3 \mathbf{h}_3^{\mathrm{T}}$$

$$|\hat{\mathbf{X}}_s| \approx \mathbf{w}_1 \mathbf{h}_1^{\mathrm{T}}$$

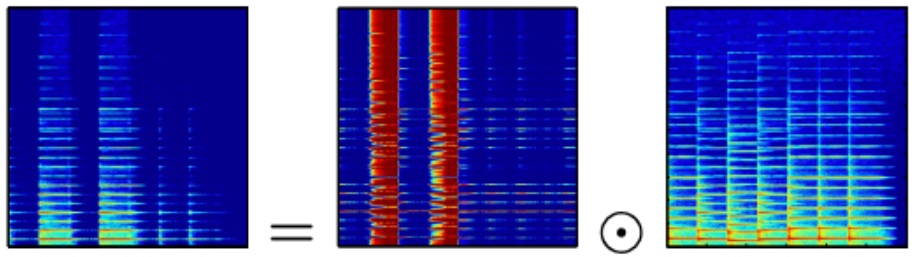
Reconstruction with masking

1. Compute filter $\mathbf{M}_s = \frac{\mathbf{w}_1 \mathbf{h}_1^{\mathrm{T}}}{\sum_{i=1}^{K} \mathbf{w}_i \mathbf{h}_i^{\mathrm{T}}}$, with $\alpha = 1$

$$\mathbf{M}_s = \frac{(\mathbf{W}_s \mathbf{H}_s)^{\alpha}}{\sum_{i=1}^{K} (\mathbf{W}_i \mathbf{H}_i)^{\alpha}} = \frac{|\hat{\mathbf{X}}_s|^{\alpha}}{\sum_{i=1}^{K} |\hat{\mathbf{X}}_i|^{\alpha}}$$



2. Multiply with $|\hat{\mathbf{X}}_s| = \mathbf{M}_s \odot |\mathbf{X}|$



Spectrum to x[n]

- Inverse DFT to get $x[n]$
  - Use $|\hat{X}_s[k]|$ and $\angle X[k]$

Source Separation

- Train W_1 from source 1
- Train W_2 from source 2
- Decompose mixture as $V = [W_1\ W_2]\begin{bmatrix} H_1 \\ H_2 \end{bmatrix}$

References: optional, for advanced study

- Smaragdis, P., & Brown, J. C. (2003, October). Non-negative matrix factorization for polyphonic music transcription. In 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No. 03TH8684) (pp. 177-180). IEEE.
- Smaragdis, P., Raj, B., & Shashanka, M. (2011). Missing data imputation for time-frequency representations of audio signals. Journal of signal processing systems, 65(3), 361-370.
- Wang, Y. X., & Zhang, Y. J. (2011, September). Image inpainting via weighted sparse non-negative matrix factorization. In 2011 18th IEEE International Conference on Image Processing (pp. 3409-3412). IEEE.
- Benzi, K., Kalofolias, V., Bresson, X., & Vandergheynst, P. (2016, March). Song recommendation with non-negative matrix factorization and graph total variation. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2439-2443). Ieee.

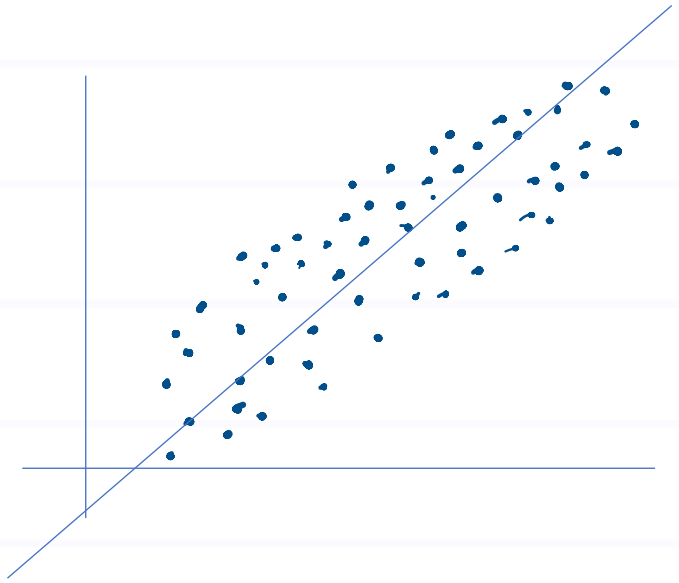
Supervised Dimensionality Reduction

Reference: PRML Section 4.1.4

Linear Transformation

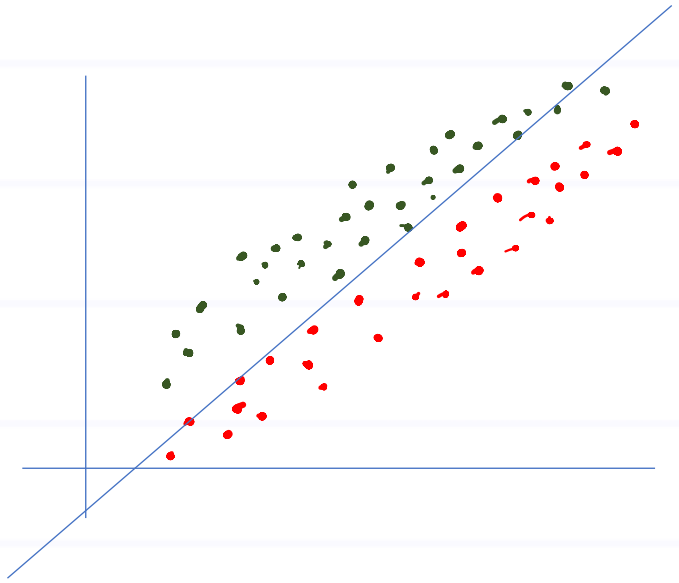
- PCA:
  - Find eigen vectors and project along the eigenvector(s) with maximum eigenvalues
  - $\boldsymbol{y} = U\boldsymbol{x}$

PCA for dim. red.



$$\boldsymbol{y} = U\boldsymbol{x}$$

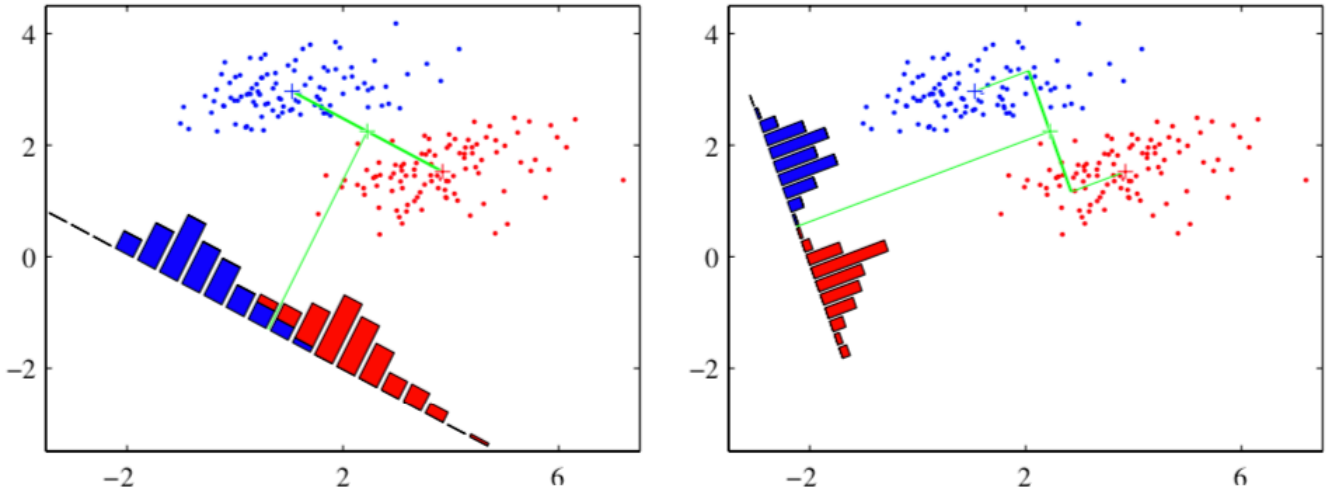
PCA for dim. red.



$$\boldsymbol{y} = U\boldsymbol{x}$$

How to find a good direction

- $y = wx$

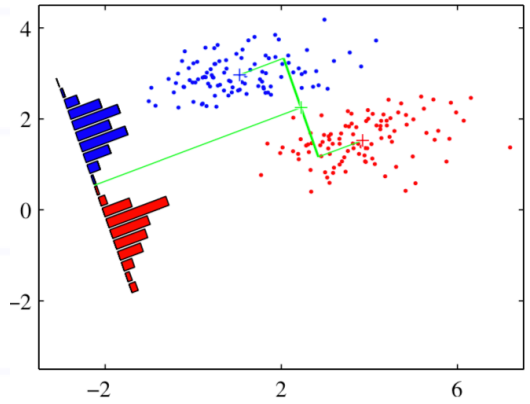


- We should **maximize** the separation between means, but **minimize** the intra-class variance

$y = w^T x$: how to find w

- Mean of each class in x

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} \mathbf{x}_n \qquad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} \mathbf{x}_n$$



- Mean of each class in y

$$m_k = \mathbf{w}^{\mathrm{T}} \mathbf{m}_k \qquad k \in \{1, 2\}$$

- Intra-class variance:

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2$$

$y = w^T x$: how to find w

- Thus, find $\boldsymbol{w}$ which maximizes

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

$$J(w) = \frac{\left(w^T m_2 - w^T m_1\right)\left(w^T m_2 - w^T m_1\right)^T}{\sum_k \sum_{n \in C_k} \left(w^T x_n - w^T m_k\right)\left(w^T x_n - w^T m_k\right)^T} = \frac{w^T S_B w}{w^T S_W w}$$

Magnitude of w is not important; it could be set to 1

where,

$$\mathbf{S}_{\mathrm{B}} = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}}$$

$$\mathbf{S}_{\mathrm{W}} = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^{\mathrm{T}} + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^{\mathrm{T}}$$

Put

$$\frac{\partial J(w)}{\partial w} = 0 \Rightarrow \underbrace{(w^T S_B w)}_{\text{scalar}} S_w w - \underbrace{(w^T S_w w)}_{\text{scalar}} S_B w = 0$$

$$S_w w = (m_2 - m_1)\underbrace{(m_2 - m_1)^T w}_{\text{scalar}} \times \text{Scalar}$$

$$\Rightarrow \boxed{w = S_w^{-1}(m_2 - m_1)}$$

This method is called **Fisher's Linear Discriminant**