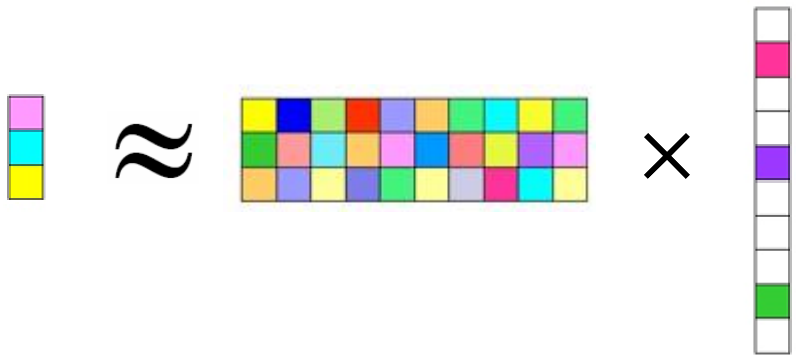
Sparse Dictionary Learning:
Algorithm Comparison and Challenges

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}$$



Kerry Sun
sunx0486@umn.edu
05/06/2019

Outline

- Background
  - Problem set-up
  - Motivation
  - Math
- Algorithms
- Experiments
- Limitations/challenges

What is sparse Dictionary Learning?

Sparse vector

Input data vector

Dictionary

$$\boldsymbol{D}\boldsymbol{w} = \boldsymbol{x}, \boldsymbol{D} \in \mathbb{R}^{N \times K},$$

$$\textbf{\textit{where}}\ K \gg \mathbf{N}\ \textbf{and}\ \|\boldsymbol{w}\|_0^0 = \mathbf{k_0} \ll \mathbf{N}$$

$\boldsymbol{x} = \sum_{j=1}^{K} w_j \boldsymbol{d}_j$ where $\mathbf{D} = \{\boldsymbol{d}_j\}$ of K vector, K > N. D is an overcomplete set

1. Sparse Coding (obtaining a sparse $\boldsymbol{w}$)
2. Learning Dictionary (obtaining a matrix $\boldsymbol{D}$)

What is sparse Dictionary Learning?

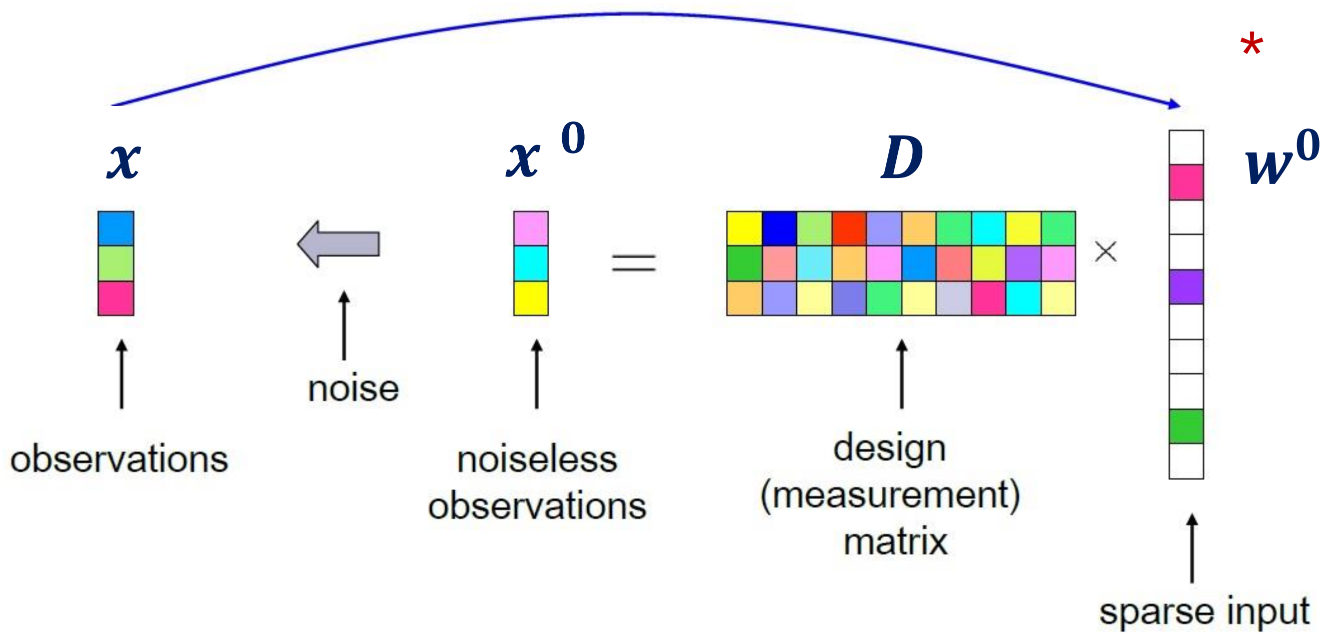
Sparse vector

Data (observation) vector

Dictionary

$$\boldsymbol{Dw} = \boldsymbol{x}, \boldsymbol{D} \in \mathbb{R}^{N\times K},$$

$$where\ K \gg \mathrm{N}\ \mathbf{and}\ \|w\|_0^0 = \mathrm{k}_0 \ll \mathrm{N}$$



* https://sites.google.com/site/yorkyuhuang/home/tutorial/machine-learning

Why sparse Dictionary Learning?

Sparse vector

Input data vector

Dictionary

$$\boldsymbol{D}\boldsymbol{w} = \boldsymbol{x}, \boldsymbol{D} \in \mathbb{R}^{N \times K},$$

$$\textit{where } K \gg \mathrm{N} \text{ and } \|\boldsymbol{w}\|_0^0 = \mathrm{k}_0 \ll \mathrm{N}$$

1. Not always can find an optimal transform; want to represent input data using as few components as possible
2. Pre-constructed dictionaries (e.g. Fourier, wavelets, DCT*) are usually restricted to signals/images of a certain type
3. Dictionary can be learned from the input data
4. Can be used in many applications such as Compressed sensing, signal recovery, de-noising.

DCT: Discrete cosine transform

Applications

- Analysis

  Given **x**, can we determine the underlying vector $\boldsymbol{w}_0$?

  $\|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{w}_0\|_2 \le \epsilon \quad \rightarrow \quad (P_0^{\epsilon})\text{: } \min_x \|\boldsymbol{w}\|_0 \; subject\ to \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{w}\|_2 \le \epsilon$

- Compression

- De-noising

  $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{v}$, where $\|\boldsymbol{v}\|_2 \le \delta$, $\rightarrow \left(P_0^{\epsilon+\delta}\right)\text{: } \min_x \|\boldsymbol{w}\|_0 \; subject\ to \|\tilde{\boldsymbol{x}} - \boldsymbol{D}\boldsymbol{w}\|_2 \le \epsilon + \delta$

  find approximation by $\boldsymbol{D}\boldsymbol{w}_{\boldsymbol{0}}^{\boldsymbol{\epsilon}+\boldsymbol{\delta}}$

- Compressed-Sensing

- Inverse Problems

- Morphological Component Analysis (MCA)
  - Can we separate 2 sources?

  $\min_{\boldsymbol{w}_1, \boldsymbol{w}_2} \|\boldsymbol{w}_1\|_0 + \|\boldsymbol{w}_2\|_0 \; subject\ to \|x - \boldsymbol{D}_1\boldsymbol{w}_1 - \boldsymbol{D}_2\boldsymbol{w}_2\|_2^2 \le \epsilon_1^2 + \epsilon_2^2$

Application-Denoising

Original clean image

Noisy image, 20.1721dB

Clean Image by Global Trained dictionary, 27.5672dB *

*

Inpainting result using the local-K-SVD

Left: the original image, Center: the degraded image with red text representing missing pixels, right: the recovered image (PSNR= 32.45dB)

* Michael Elad. 2010. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing (1st ed.). Springer Publishing Company, Incorporated.

Application-Image Compression

Original | JPEG (15.81) | JPEG2000 (13.89) | PCA (10.66) | K-SVD (6.67)

Original | JPEG (14.67) | JPEG2000 (12.41) | PCA (9.44) | K-SVD (5.63)

Original | JPEG (15.3) | JPEG2000 (12.57) | PCA (10.27) | K-SVD (6.45)

*

Values in () is RMSE

Michael Elad. 2010. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing (1st ed.). Springer Publishing Company, Incorporated.

Math Background

– the l_2 norm. $\|\alpha\|_2^2 \triangleq \sum_{i=1}^{m} \alpha_i^2$

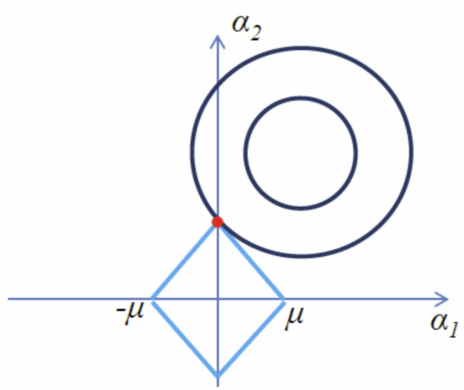
– the l_0 norm. $\|\alpha\|_0 \triangleq \#\{i \mid a_i \neq 0\}$

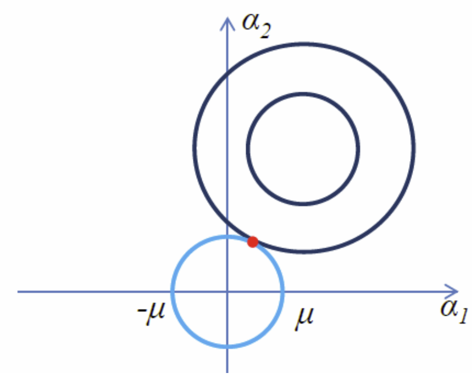
– the l_1 norm. $\|\alpha\|_1 \triangleq \sum_{i=1}^{m} |\alpha_i|$

Sparsity inducing

$$\min_{\alpha \in \mathbb{R}} \frac{1}{2}\|x-\alpha\|_2^2 + \lambda\|\alpha\|_1$$
$$\Leftrightarrow \min_{\alpha \in \mathbb{R}} \frac{1}{2}\|x-\alpha\|_2^2 \text{ s.t. } \|\alpha\|_1 \leq \mu$$

$$\min_{\alpha \in \mathbb{R}} \frac{1}{2}\|x-\alpha\|_2^2 + \lambda\|\alpha\|_2^2$$
$$\Leftrightarrow \min_{\alpha \in \mathbb{R}} \frac{1}{2}\|x-\alpha\|_2^2 \text{ s.t. } \|\alpha\|_2 \leq \mu$$





l_p $norm$ where $l_p \in [0,1]$ induce sparsity

http://www.math.nus.edu.sg/~matzuows/BJQS1.pdf

Problem Definition (Setup)

Given: the input dataset $\boldsymbol{X} = [x_1, \dots, x_l], x_l \in \mathbb{R}^N$
Objective: find

1) a dictionary $\boldsymbol{D} \in \mathbb{R}^{N \times K}$
2) a coefficient set $\boldsymbol{W} = [w_1, \dots, w_l], w_l \in \mathbb{R}^K$

Such that both $\|\boldsymbol{X} - \boldsymbol{DW}\|_F^2$ is minimized* and w_l are sparse(N<<K)

$$\underset{\boldsymbol{D} \in \mathcal{C}, \boldsymbol{w}_l \in \mathbb{R}^K}{\arg\min} \sum_{l=1}^{L} \|\boldsymbol{x}_l - \boldsymbol{D}\boldsymbol{w}_l\|_2^2 + \lambda \|\boldsymbol{w}_l\|_p, \; p \in [0, 1]$$

$$\mathcal{C} \equiv \{\boldsymbol{D} \in \mathbb{R}^{N \times K} : \|\boldsymbol{d_i}\|_2 \leq 1 \, \forall 1, \dots, K\}$$

$$\lambda > 0$$

$$\|\boldsymbol{X} - \boldsymbol{DW}\|_F^2 = \sum_l \|\boldsymbol{x}_l - \boldsymbol{D}\boldsymbol{w}_l\|_2^2$$

Algorithms

Objective: Find an optimal sparse coding $\boldsymbol{W}$ and a dictionary $\mathbf{D}$

General strategy: Split the problem into 2 parts:

1) Keep $\mathbf{D}$ fixed, find $\boldsymbol{W}$ (sparse coding)
2) Keep $\boldsymbol{W}$ fixed, find $\mathbf{D}$ (dictionary learning)

Popular Algorithms:

(combine sparse coding and dictionary learning together)

- MOD or ILS-DLA (1999)
- K-SVD – Compared to K-means (2006)
- RLS-DLA (2010)
- ODL (LASSO) (2010)

Spare Coding

Keeping $\boldsymbol{D}$ fixed and find $\boldsymbol{W}$: L independent problem

$$\underset{\boldsymbol{D}\in\mathcal{C},\boldsymbol{w}_l\in\mathbb{R}^K}{\arg\min} \sum_{l=1}^{L} \|\boldsymbol{x}_l - \boldsymbol{D}\boldsymbol{w}_l\|_2^2 + \lambda \|\boldsymbol{w}_l\|_p \,,\ p \in [0,1]$$

Strategy: Vector selection algorithm

- Pursuit algorithm (e.g. Matching Pursuit (MP) and OMP based on l_0 $norm$)
- LARS and LASSO algorithm (based on l_1 $norm$)
- IRLS algorithm (based on l_p $norm$)
- Dantzig-Selector algorithm (based on l_∞ $norm$)

Spare Coding - MP

What is matching pursuit (MP) algorithm?

it is a sparse approximation algorithm which finds the "best matching" projections of data onto of $\boldsymbol{D}$

Idea: suppose optimal $S = 1$, so $\boldsymbol{x} = c \cdot \boldsymbol{w}$

$$\epsilon(k) = \min_{w_k} \|\boldsymbol{d}_k w_k - \boldsymbol{x}\|_2^2 , \; w_k^* = \boldsymbol{d}_k^T \boldsymbol{x} / \|\boldsymbol{d}_k\|_2^2$$

A series of locally optimal single-term updates from $\boldsymbol{w}^{\boldsymbol{0}} = \boldsymbol{0}$, and iteratively construct a k-term $\boldsymbol{w}^{\boldsymbol{k}}$

Spare Coding - MP

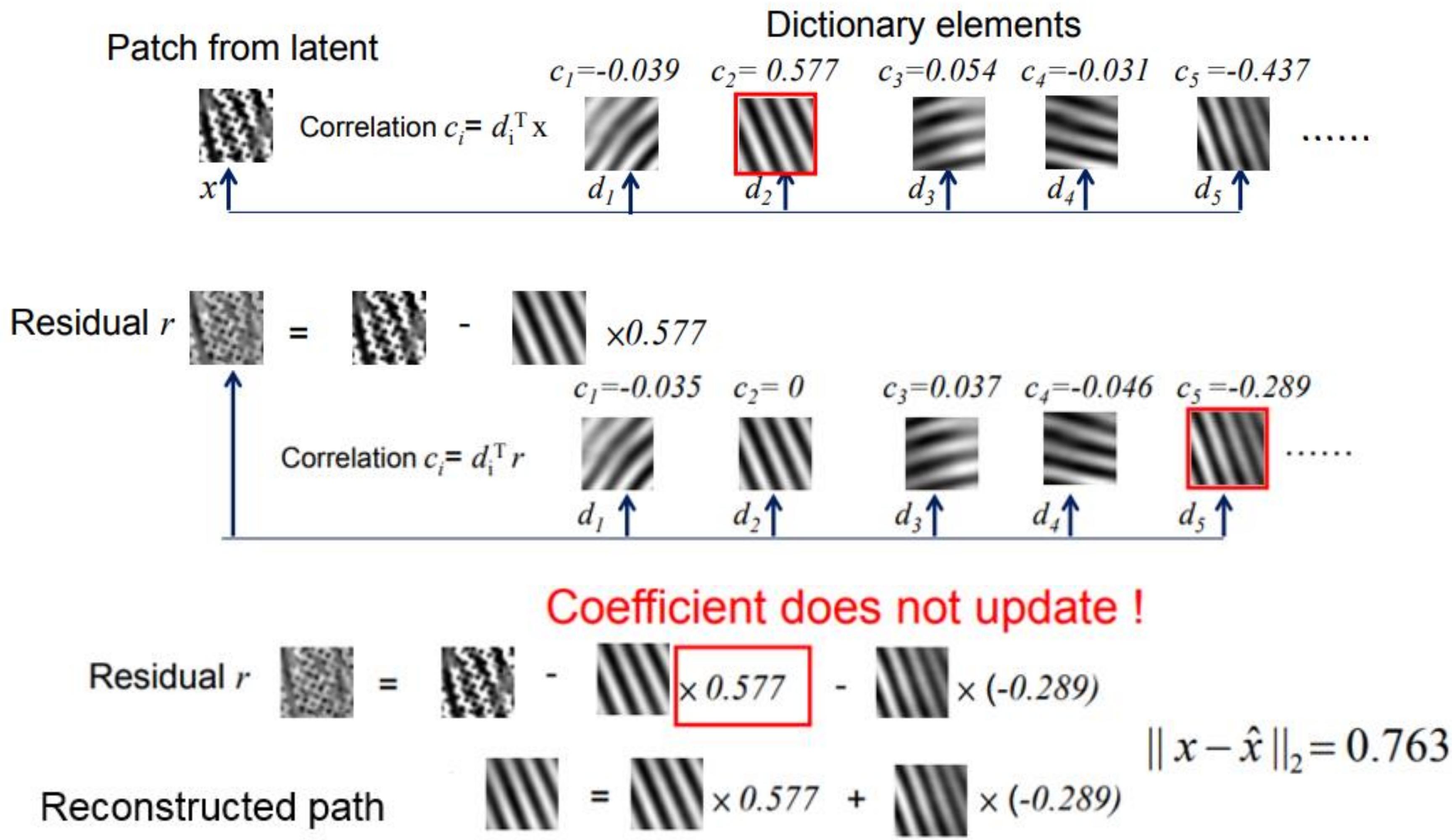
What is matching pursuit (MP) algorithm?

$$\min_{\boldsymbol{w}\in\mathbb{R}^K} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{D}\boldsymbol{w}\|_2^2 \quad s.t. \quad \|\boldsymbol{w}\|_0 \leq S$$

Algorithm 1 Batch Matching Pursuit algorithm

Input : Signal $\boldsymbol{x}$ and a normalized dictionary $\boldsymbol{D}$
Output: List of coefficients $(w_k)_{k=1}^K$ and indices for corresponding atoms $(\gamma_k)_{k=1}^K$
Initialization :
$\boldsymbol{r} \leftarrow \boldsymbol{x}, \ \boldsymbol{w} \leftarrow \boldsymbol{0}$
while not Finished:
$\boldsymbol{c} \leftarrow \boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{r}$ (inner product)
find k : $\text{argmax}_k |c_k|$ ⟵ Find the index of biggest c_k
$w_k \leftarrow w_k + c_k$
$\boldsymbol{r} \leftarrow \boldsymbol{r} - c_k \cdot \boldsymbol{d_k}$
Finished: $|\boldsymbol{r}| \leq$ some limit
Finished: s non-zero entries in $\boldsymbol{w}$
end

Spare Coding - MP



Algorithm – Method of Optimal Direction (MOD)

Initialize $\boldsymbol{D}$

Keeping $\boldsymbol{D}$ fixed and find $\boldsymbol{W}$: L independent problem

Strategy: Vector selection algorithm (e.g. Matching Pursuit (MP))

Keeping $\boldsymbol{W}$ fixed and find $\boldsymbol{D}$ using least square solution

Strategy: by minimizing $\|\boldsymbol{X}-\boldsymbol{D}\boldsymbol{W}\|_F^2 \Rightarrow (\boldsymbol{X}-\boldsymbol{D}\boldsymbol{W})\boldsymbol{W}^T = 0$

$\Rightarrow$ Least square $\boldsymbol{D} = (\boldsymbol{X}\boldsymbol{W}^T)(\boldsymbol{W}\boldsymbol{W}^T)^{-1}$

Issues:

- Can be slow when having large number of dictionary column because of the inversion
- Dictionary is updated before turning to re-evaluate the coefficients, which can inflict a server limitation on the training speed.

K-SVD

$$\min_{\boldsymbol{D},\boldsymbol{W}} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{W}\|_F^2 \text{ subject to } \forall i, \ \|\boldsymbol{w}_i\|_0 \leq T_0$$

Initialize $\boldsymbol{D}$

Keeping $\boldsymbol{D}$ fixed and find $\boldsymbol{W}$: L independent problem

Strategy: Orthogonal Matching Pursuit (OMP)

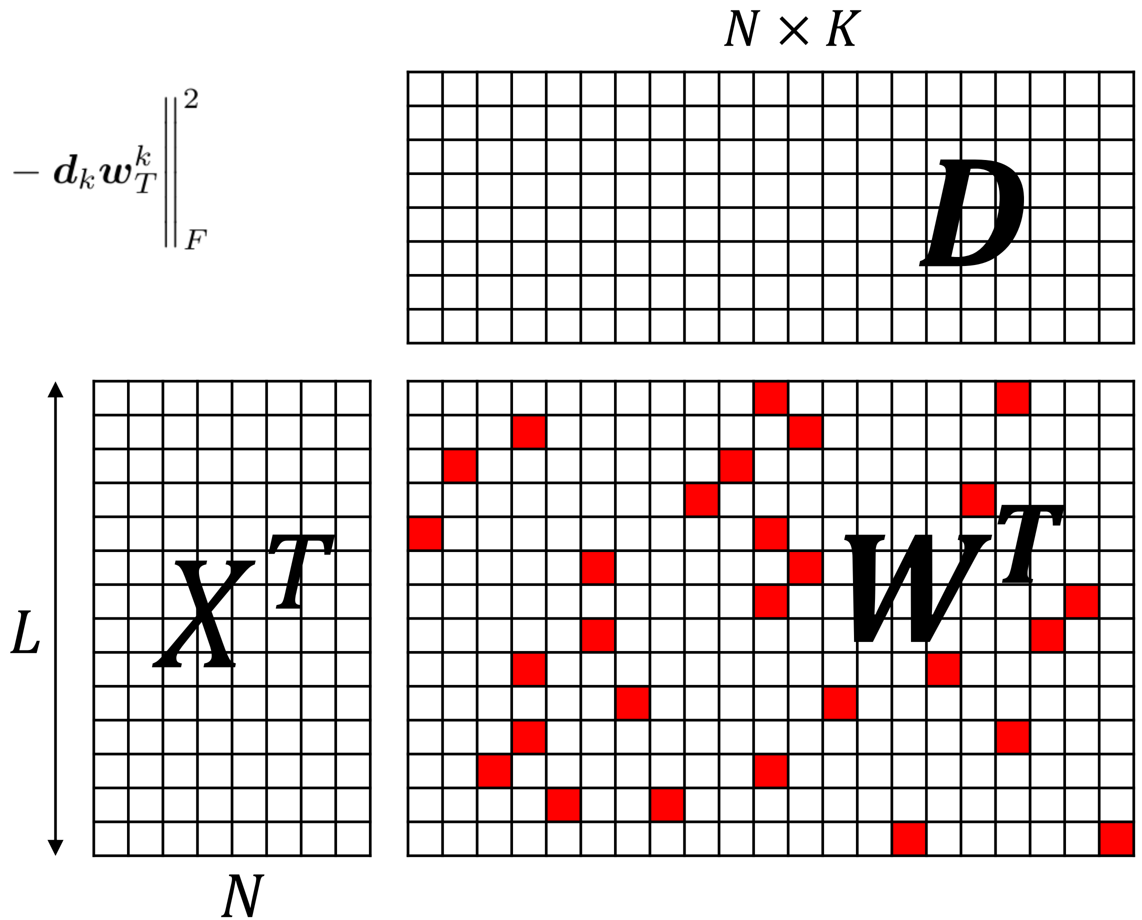
$$\boldsymbol{D}_{\mathcal{S}^k}^T(\boldsymbol{D}_{\mathcal{S}^k}\boldsymbol{w}_{\mathcal{S}^k} - \boldsymbol{x}) = -\boldsymbol{D}_{\mathcal{S}^k}^T\boldsymbol{r}^k = 0$$

Keeping nonzero positions in $\boldsymbol{W}$ fixed and find $\boldsymbol{D}$ and $\boldsymbol{W}$ using SVD decomposition.

Generalization of K-means clustering process

K-SVD continued

$$
\begin{aligned}
\|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{W}\|_F^2 &= \left\| \boldsymbol{X} - \sum_{j=1}^{K} \boldsymbol{d}_j \boldsymbol{w}_T^j \right\|_F^2 \\
&= \left\| \left(\boldsymbol{X} - \sum_{j \neq k} \boldsymbol{d}_j \boldsymbol{w}_T^j \right) - \boldsymbol{d}_k \boldsymbol{w}_T^k \right\|_F^2 \\
&= \left\| \boldsymbol{E}_k - \boldsymbol{d}_k \boldsymbol{w}_T^k \right\|_F^2
\end{aligned}
$$



K-SVD continued

$$\|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{W}\|_F^2 = \left\| \boldsymbol{X} - \sum_{j=1}^{K} \boldsymbol{d}_j \boldsymbol{w}_T^j \right\|_F^2$$

$$= \left\| \left(\boldsymbol{X} - \sum_{j \neq k} \boldsymbol{d}_j \boldsymbol{w}_T^j \right) - \boldsymbol{d}_k \boldsymbol{w}_T^k \right\|_F^2$$

$$= \|\boldsymbol{E}_k - \boldsymbol{d}_k \boldsymbol{w}_T^k\|_F^2$$

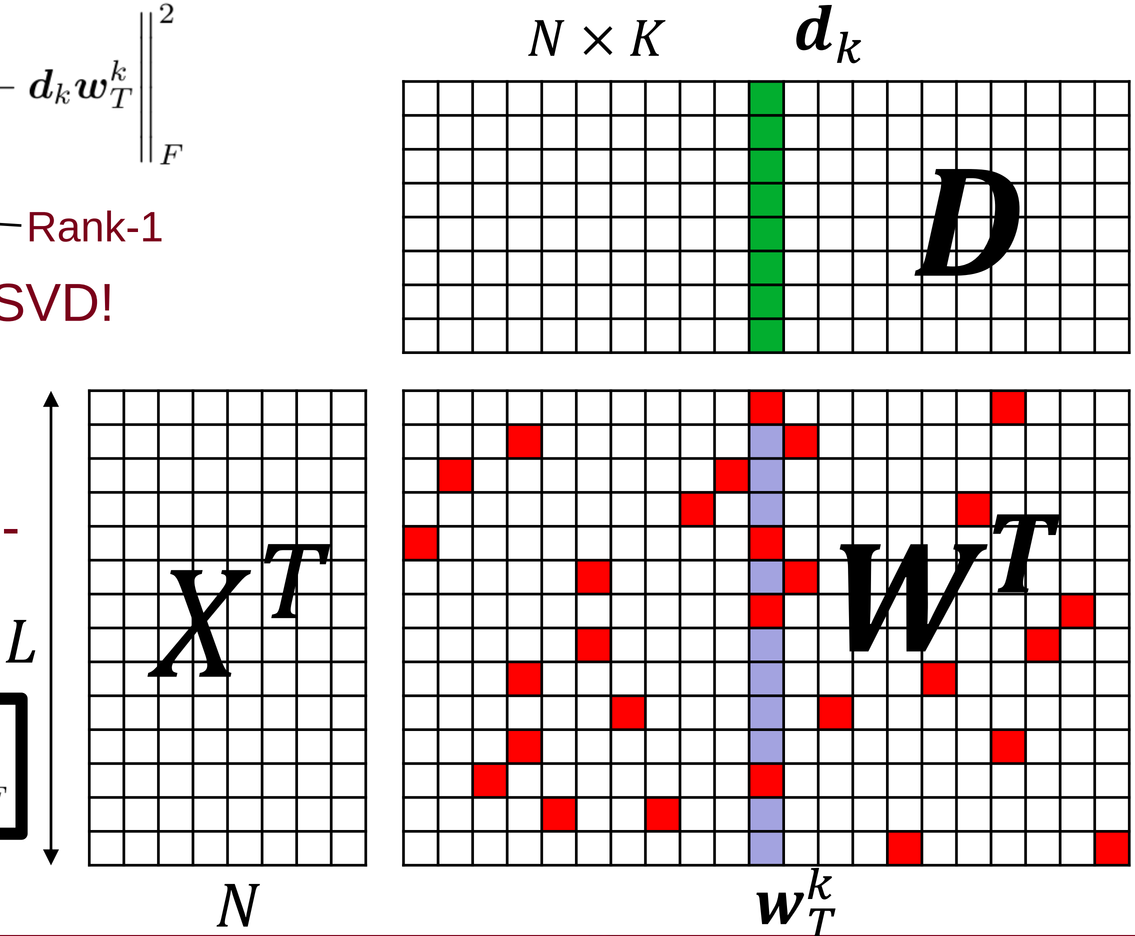
Rank-1

Rank-1 approximation via SVD!

To keep cardinalies of $\boldsymbol{W}$, Define $\boldsymbol{\Omega_k}$ ($L \times |\omega_l|$) such that it only pick out the non-zero entries.

$$\boxed{\left\| \boldsymbol{E}_k \boldsymbol{\Omega}_K - \boldsymbol{d}_k \boldsymbol{w}_T^k \boldsymbol{\Omega}_K \right\|_F^2 = \left\| \boldsymbol{E}_k^R - \boldsymbol{d}_k \boldsymbol{w}_R^k \right\|_F^2}$$

$$\boldsymbol{E}_k^R = U \Delta V^T$$



K-SVD continued

$$\|\boldsymbol{X}-\boldsymbol{D}\boldsymbol{W}\|_F^2 = \left\|\boldsymbol{X}-\sum_{j=1}^{K}\boldsymbol{d}_j\boldsymbol{w}_T^j\right\|_F^2$$

$$= \left\|\left(\boldsymbol{X}-\sum_{j\neq k}\boldsymbol{d}_j\boldsymbol{w}_T^j\right)-\boldsymbol{d}_k\boldsymbol{w}_T^k\right\|_F^2$$

$$= \|\boldsymbol{E}_k - \boldsymbol{d}_k\boldsymbol{w}_T^k\|_F^2$$

Rank-1

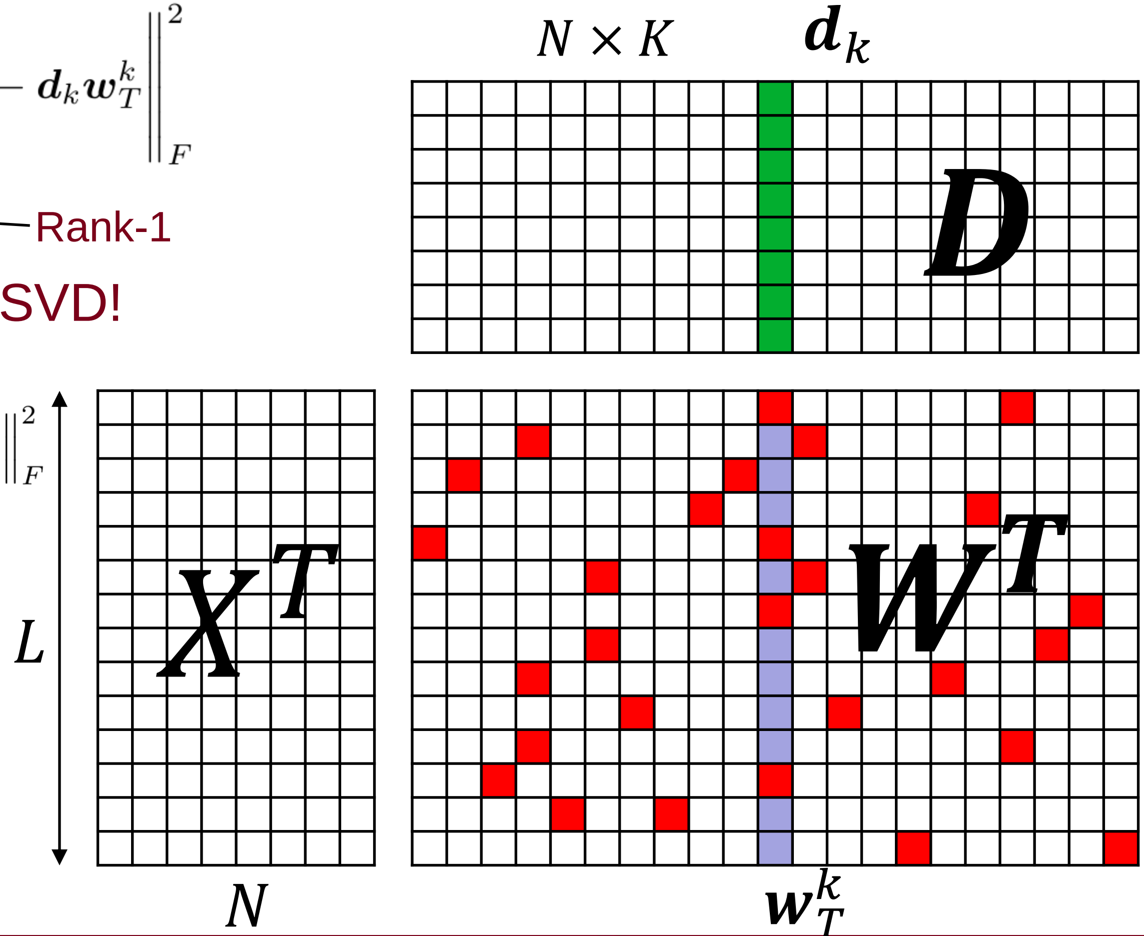
Rank-1 approximation via SVD!

$$\left\|\boldsymbol{E}_k\boldsymbol{\Omega}_K - \boldsymbol{d}_k\boldsymbol{w}_T^k\boldsymbol{\Omega}_K\right\|_F^2 = \left\|\boldsymbol{E}_k^R - \boldsymbol{d}_k\boldsymbol{w}_R^k\right\|_F^2$$

$$\boldsymbol{E}_k^R = U\Delta V^T$$

$$\boldsymbol{d}_k \rightarrow \tilde{\boldsymbol{d}}_k = U(:,1)$$

$$\boldsymbol{w}_R^k \rightarrow V(:,1)\Delta(1,1)$$



Aharon, M, Elad, M, Bruckstein, A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, 2006.

K-SVD continued

$$\|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{W}\|_F^2 = \left\| \boldsymbol{X} - \sum_{j=1}^{K} \boldsymbol{d}_j \boldsymbol{w}_T^j \right\|_F^2$$

$$= \left\| \left(\boldsymbol{X} - \sum_{j \neq k} \boldsymbol{d}_j \boldsymbol{w}_T^j \right) - \boldsymbol{d}_k \boldsymbol{w}_T^k \right\|_F^2$$

$$= \left\| \boldsymbol{E}_k - \boldsymbol{d}_k \boldsymbol{w}_T^k \right\|_F^2$$

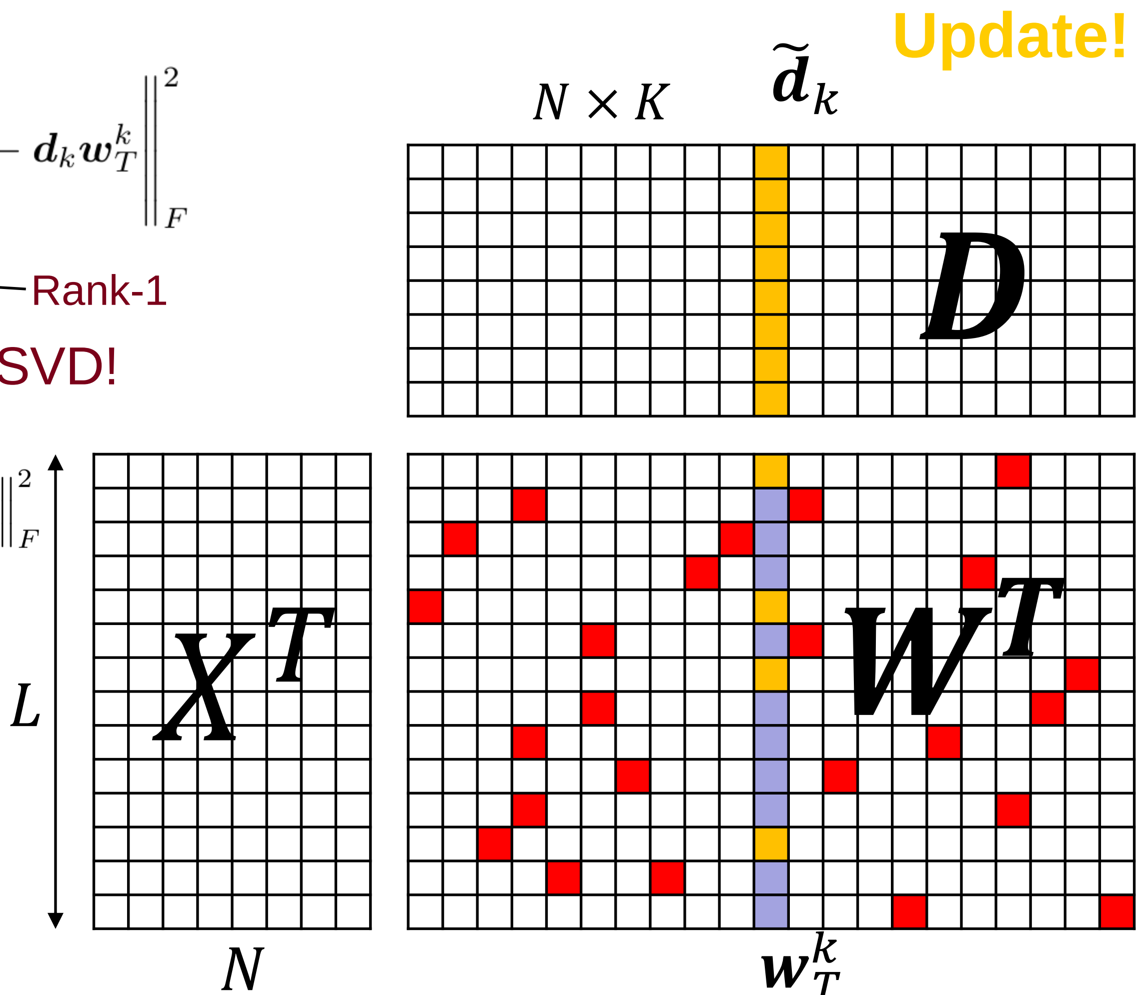
Rank-1

Rank-1 approximation via SVD!

$$\left\| \boldsymbol{E}_k \boldsymbol{\Omega}_K - \boldsymbol{d}_k \boldsymbol{w}_T^k \boldsymbol{\Omega}_K \right\|_F^2 = \left\| \boldsymbol{E}_k^R - \boldsymbol{d}_k \boldsymbol{w}_R^k \right\|_F^2$$

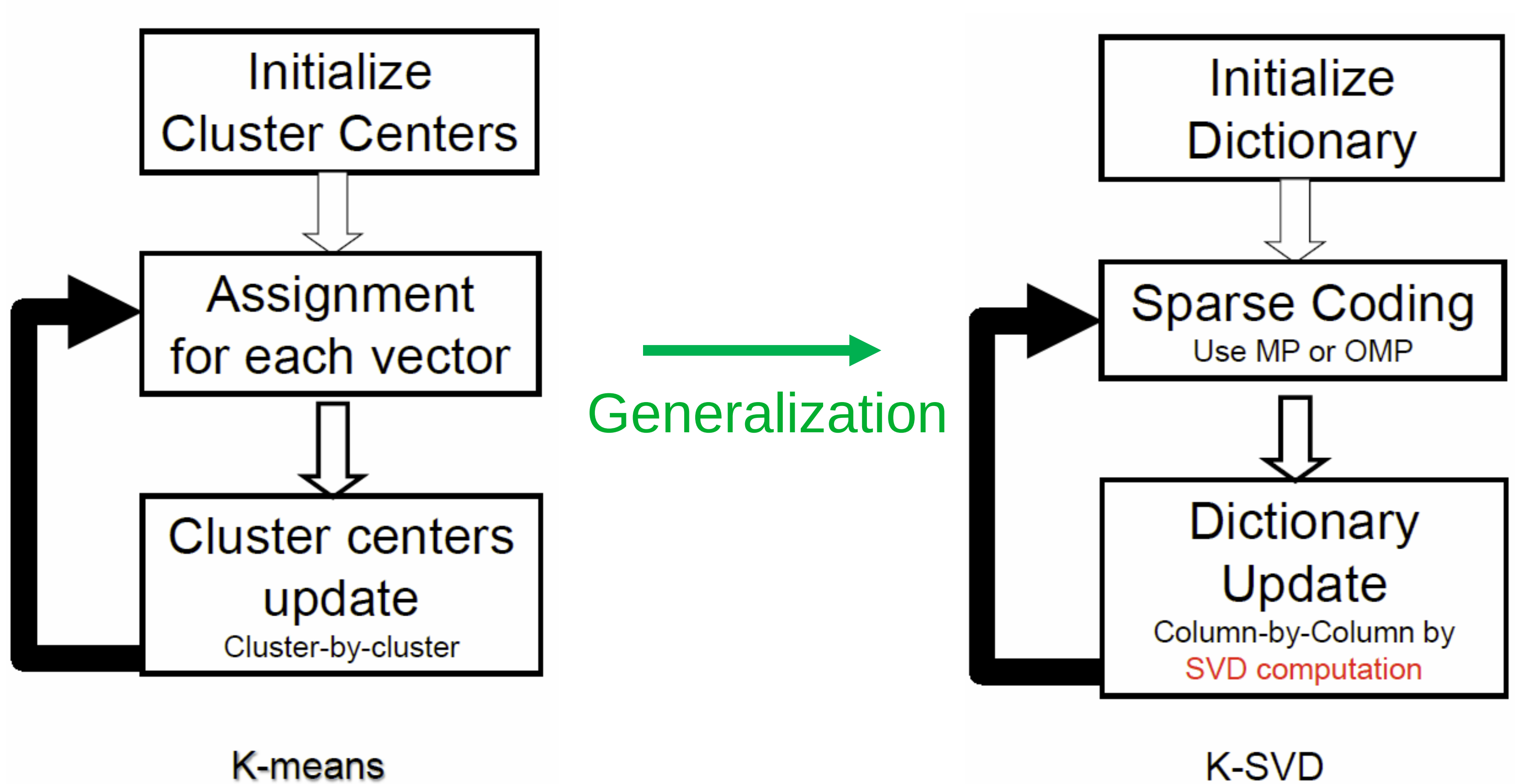
$$\boldsymbol{E}_k^R = U \Delta V^T$$

$$\boxed{\begin{aligned} \boldsymbol{d}_k &\rightarrow \tilde{\boldsymbol{d}}_k = U(:,1) \\ \boldsymbol{w}_R^k &\rightarrow V(:,1)\Delta(1,1) \end{aligned}}$$



Aharon, M, Elad, M, Bruckstein, A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, 2006.

Compare K-SVD with K-means



$$\min_{\boldsymbol{C},\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{C}\boldsymbol{X}\|_F^2 \ \text{ subject to } \forall i,\ \boldsymbol{x}_i = \boldsymbol{e}_k \text{ for some } k \longrightarrow \min_{\boldsymbol{D},\boldsymbol{W}} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{W}\|_F^2 \ \text{ subject to } \forall i,\ \|\boldsymbol{w}_i\|_0 \leq T_0$$

Online Dictionary Learning (ODL)

Algorithm 1 Online dictionary learning.

Require: $\mathbf{x} \in \mathbb{R}^m \sim p(\mathbf{x})$ (random variable and an algorithm to draw i.i.d samples of p), $\lambda \in \mathbb{R}$ (regularization parameter), $\mathbf{D}_0 \in \mathbb{R}^{m \times k}$ (initial dictionary), T (number of iterations).

$\mathbf{A}_0 \leftarrow 0$, $\mathbf{B}_0 \leftarrow 0$ (reset the "past" information).
for $t = 1$ to T **do**
  Draw $\mathbf{x}_t$ from $p(\mathbf{x})$.
  Sparse coding: compute using LARS

$$\boxed{\boldsymbol{\alpha}_t \triangleq \operatorname*{arg\,min}_{\boldsymbol{\alpha} \in \mathbb{R}^k} \frac{1}{2} ||\mathbf{x}_t - \mathbf{D}_{t-1}\boldsymbol{\alpha}||_2^2 + \lambda ||\boldsymbol{\alpha}||_1.} \quad (8)$$

  $\mathbf{A}_t \leftarrow \mathbf{A}_{t-1} + \boldsymbol{\alpha}_t \boldsymbol{\alpha}_t^T$.
  $\mathbf{B}_t \leftarrow \mathbf{B}_{t-1} + \mathbf{x}_t \boldsymbol{\alpha}_t^T$.
  Compute $\mathbf{D}_t$ using Algorithm 2, with $\mathbf{D}_{t-1}$ as warm restart, so that

$$\mathbf{D}_t \triangleq \operatorname*{arg\,min}_{\mathbf{D} \in \mathcal{C}} \frac{1}{t} \sum_{i=1}^{t} \frac{1}{2} ||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda ||\boldsymbol{\alpha}_i||_1,$$
$$= \operatorname*{arg\,min}_{\mathbf{D} \in \mathcal{C}} \frac{1}{t} \Big(\frac{1}{2} \operatorname{Tr}(\mathbf{D}^T \mathbf{D} \mathbf{A}_t) - \operatorname{Tr}(\mathbf{D}^T \mathbf{B}_t) \Big). \quad (9)$$

end for
Return $\mathbf{D}_T$ (learned dictionary).

Algorithm 2 Dictionary Update.

Require: $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_k] \in \mathbb{R}^{m \times k}$ (input dictionary),
$\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_k] \in \mathbb{R}^{k \times k} = \sum_{i=1}^{t} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_i^T$,
$\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_k] \in \mathbb{R}^{m \times k} = \sum_{i=1}^{t} \mathbf{x}_i \boldsymbol{\alpha}_i^T$.

repeat
  for $j = 1$ to k **do**
    Update the j-th column to optimize for (9):

$$\mathbf{u}_j \leftarrow \frac{1}{\mathbf{A}_{jj}} (\mathbf{b}_j - \mathbf{D}\mathbf{a}_j) + \mathbf{d}_j.$$
$$\mathbf{d}_j \leftarrow \frac{1}{\max(||\mathbf{u}_j||_2, 1)} \mathbf{u}_j. \quad (10)$$

  end for
until convergence
Return D (updated dictionary).

Advantages of online learning:

1. Handle large and dynamic datasets,
2. Could be much faster than batch algorithms

Dictionary properties

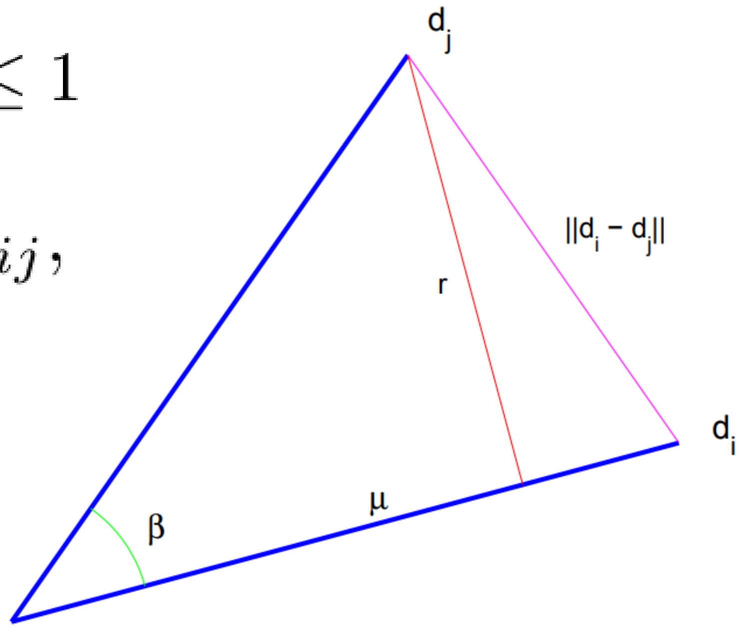
$$A\left\|x\right\|^2 \leq \sum_k |<\boldsymbol{x}, \boldsymbol{d}_k>|^2 \leq B\left\|x\right\|^2, \quad \text{for all } \boldsymbol{x} \in H^*$$

where $\quad A = \|\boldsymbol{D}\|_2^2 = \sigma_N^2, \; B = \|\boldsymbol{D}\|_2^2 = \sigma_1^2$

$$\mu_{ij} = |\boldsymbol{d}_i^T \boldsymbol{d}_j| = cos\beta_{ij}, \;\; 0 \leq \mu_{ij} \leq 1$$

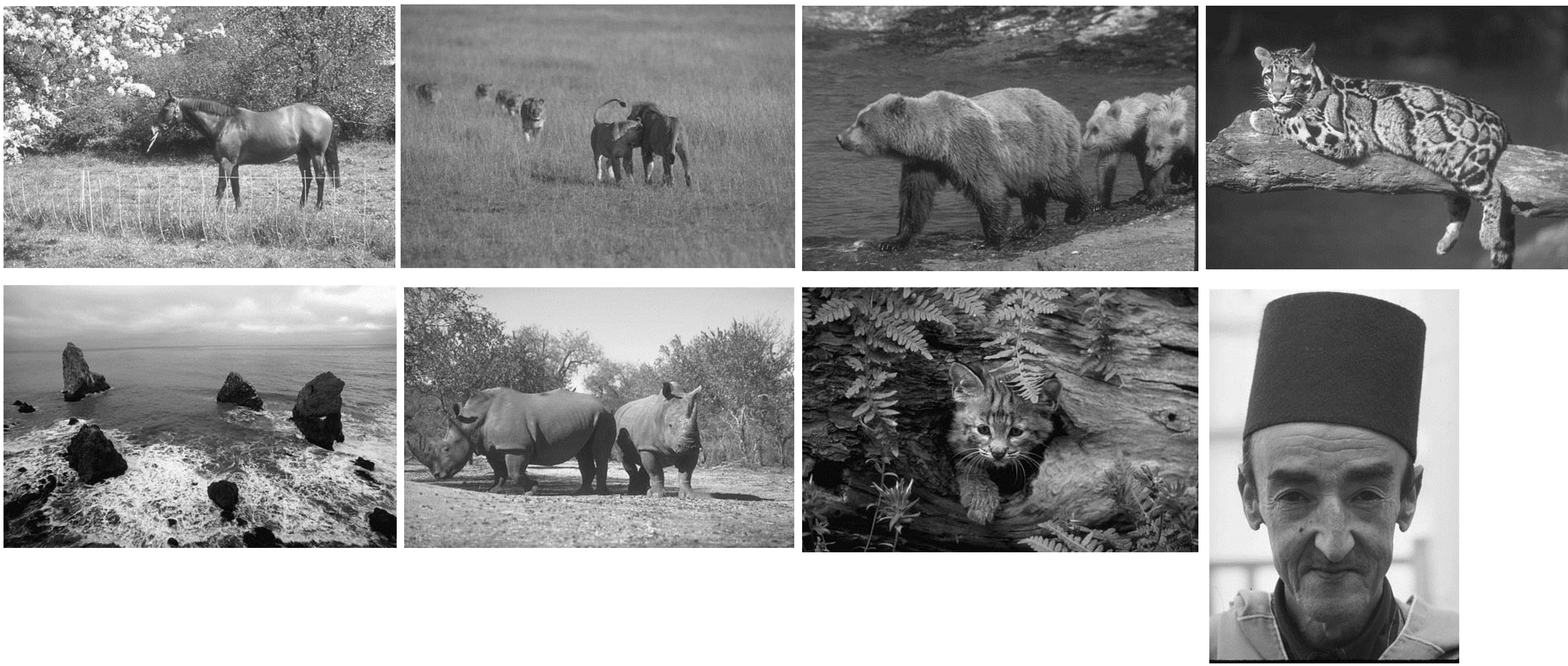
$$\beta_{ij} = arccos|\boldsymbol{d}_i^T \boldsymbol{d}_j| = arccos\mu_{ij},$$

$$0 \leq \beta_{ij} \leq \pi/2$$



*A Hilbert space H is separable if and only if it admits a countable orthonormal basis

Experiment: Image Compression



https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

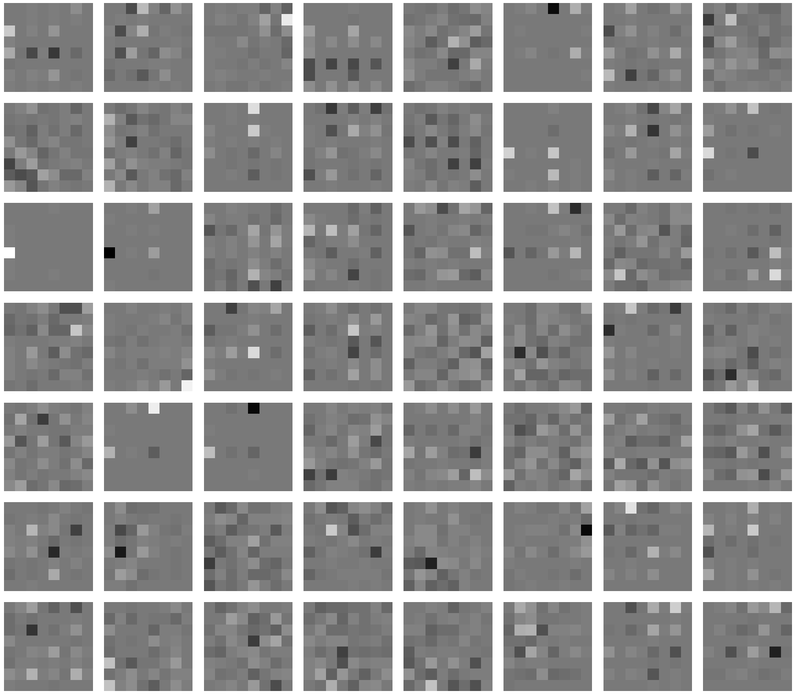
Experiment: Results

1. MOD (ILS-DLA)
2. RLS-DLA
3. K-SVD

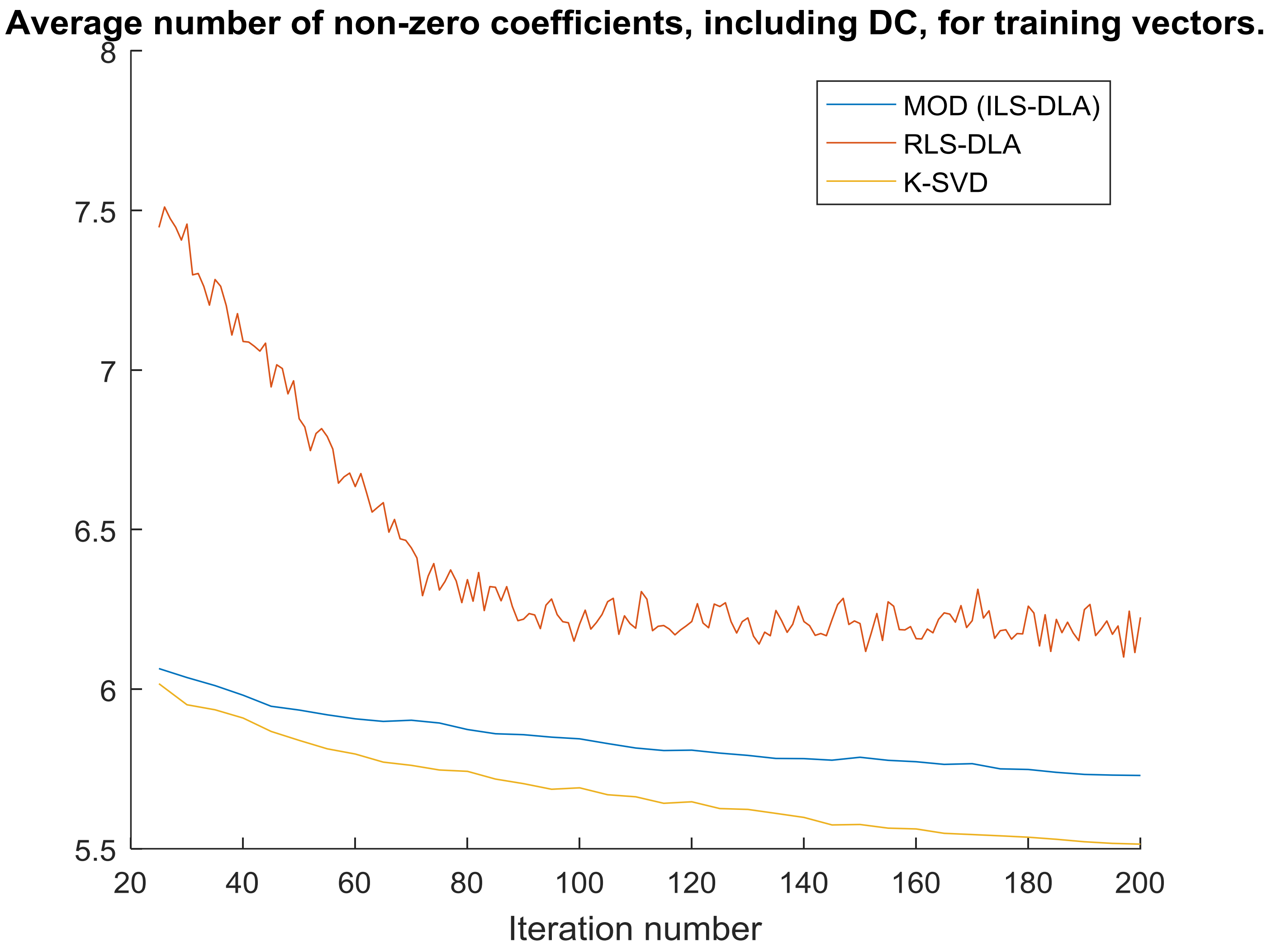
Source Code:
http://www.ux.uis.no/~karlsk/dle/

	MOD (ILS-DLA)	RLS-DLA	K-SVD
Iteration	200	200	200
tPSNR	36	36	36
Size	64×440	64×440	64×440
A	0.20	0.92	0.25
B	63.64	23.26	66.59
β_{min}	13.5	22.77	3.34
β_{avg}	47.09	57.58	46.29

The given matrix D has size 64x440



Experiment: Results



Experiment: Results

K-SVD

original Lena

512 x 512 (262144 pixels)

learned Lena

15051 nonzeros in w

$$spareness = \frac{15051}{262144} = 0.0574$$

Limitations and Challenges

- Restriction on low-dimension ($n \leq 1000$)
- Suitability of sparse dictionary learning model on general signal
  - "try and see" approach
- Ignores the existing dependencies between atoms
- Cannot synthesize reasonable signals from this model (e.g. natural image cannot obtained by direct $\boldsymbol{Dw}$)

Thanks
Any Questions?

Key Reference:

K. Engan, S. O. Aase and J. Hakon Husoy, "Method of optimal directions for frame design," *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*

M. Aharon, M. Elad, and A. M. Bruckstein. The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations*. IEEE Transactions on Signal Processing*, 54(11):4311-4322, November 2006.

Michael Elad. 2010. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing (1st ed.). Springer Publishing Company, Incorporated.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research 2010.

Useful web:

http://www.ux.uis.no/~karlsk/dle/

Backup slides-Image compression