FAST LOW-RANK SHARED DICTIONARY LEARNING FOR IMAGE CLASSIFICATION

Institute of Statistics &
Department of Applied Mathematics

Huynh Huu Dinh
ID: 8107053003

Purpose of paper

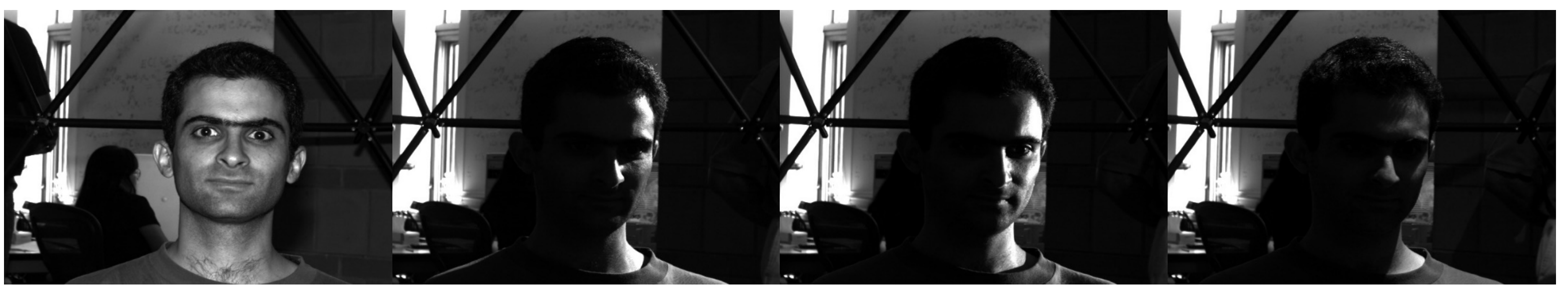
1.Building dictionary learning framework is characterized by particular dictionaries and a shared dictionary.

2. Using the low-rank shared dictionary learning to classify images and computing the accuracy.

Dataset: The Extended Yale B

- This dataset contains about 2414 frontal face images of 38 individuals. We used the cropped and normalized face images of 192x168 pixels (see [2]).

- We randomly split the dataset into two halves. One half, which contains 32 images for each person, was used for training the dictionary. The other half was used for testing.



The extended Yale B

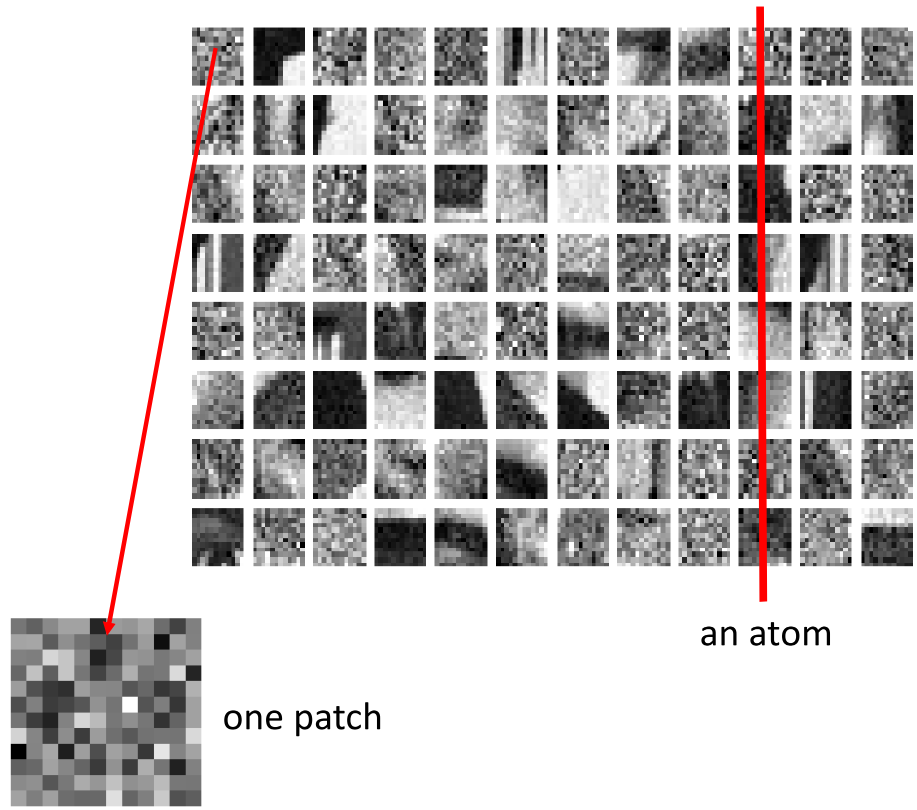
Vectorization the dataset (see [2])

	1	2	3	4	…	2413	2414
1	31.75248	28.83727	-29.2841	39.18376		-47.3624	-19.6828
2	-111.236	-82.9543	-39.9927	13.33082		5.692033	-3.7811
3	51.63764	55.73638	21.89475	-51.5378		23.22424	61.22354
4	14.09386	3.271673	1.735423	18.93872		268.1512	162.8671
5	-195.57	-211.417	-236.041	-35.5466		-103.549	-63.2441
…							
504	140.3989	127.1401	105.0745	29.25712		131.8584	111.5676
Label	**1**	**1**	**1**	**1**		**38**	**38**

- The learned dictionary contains 304 atoms, which corresponds to, on average, roughly 8 atoms for each person.

Original images

Dictionary



Algorithm

Algorithm 1 LRSDL Sparse Coefficients Update by FISTA [34]

function $(\hat{\mathbf{X}}, \hat{\mathbf{X}}^0) = \text{LRSDL\_X}(\mathbf{Y}, \mathbf{D}, \mathbf{D}_0, \mathbf{X}, \mathbf{X}^0, \lambda_1, \lambda_2)$.

1. Calculate:

$$\mathbf{A} = \mathcal{M}(\mathbf{D}^T\mathbf{D}) + 2\lambda_2\mathbf{I};$$
$$\mathbf{B} = 2\mathbf{D}_0^T\mathbf{D}_0 + \lambda_2\mathbf{I}$$
$$L = \lambda_{\max}(\mathbf{A}) + \lambda_{\max}(\mathbf{B}) + 4\lambda_2 + 1^4$$

2. Initialize $\mathbf{W}_1 = \mathbf{Z}_0 = \begin{bmatrix}\mathbf{X}\\ \mathbf{X}^0\end{bmatrix}, t_1 = 1, k = 1$

while not convergence and $k < k_{\max}$ **do**

3. Extract $\mathbf{X}, \mathbf{X}^0$ from $\mathbf{W}_k$.
4. Calculate gradient of two parts:

$$\begin{aligned}
\mathbf{M} &= \mu(\mathbf{X}), \mathbf{M}_c = \mu(\mathbf{X}_c), \widehat{\mathbf{M}} = [\mathbf{M}_1, \dots, \mathbf{M}_C].\\
\mathbf{V} &= \mathbf{Y} - \frac{1}{2}\mathbf{D}\mathcal{M}(\mathbf{X})\\
\mathbf{G} &= \begin{bmatrix}\mathbf{A}\mathbf{X} - \mathcal{M}(\mathbf{D}^T(\mathbf{Y} - \mathbf{D}_0\mathbf{X}^0)) + \lambda_2(\mathbf{M} - \widehat{\mathbf{M}})\\ \mathbf{B}\mathbf{X}^0 - \mathbf{D}_0^T\mathbf{V} - \lambda_2\mu(\mathbf{X}^0)\end{bmatrix}
\end{aligned}$$

5. $\mathbf{Z}_k = \mathcal{S}_{\lambda_1/L}(\mathbf{W}_k - \mathbf{G}/L)$ ($\mathcal{S}_\alpha()$ is the element-wise soft thresholding function. $\mathcal{S}_\alpha(x) = \text{sgn}(x)(|x| - \alpha)_+$).
6. $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$
7. $\mathbf{W}_{k+1} = \mathbf{Z}_k + \frac{t_k - 1}{t_{k+1}}(\mathbf{Z}_k - \mathbf{Z}_{k-1})$
8. $k = k + 1$

end while

9. OUTPUT: Extract $\mathbf{X}, \mathbf{X}^0$ from $\mathbf{Z}_k$.

end function

Algorithm 2 LRSDL Algorithm

function $(\hat{\mathbf{X}}, \hat{\mathbf{X}}^0) = \text{LRSDL}(\mathbf{Y}, \lambda_1, \lambda_2, \eta)$.

1. Initialization $\mathbf{X} = \mathbf{0}$, and:

$$\begin{aligned}
(\mathbf{D}_c, \mathbf{X}_c^c) &= \arg\min_{\mathbf{D},\mathbf{X}} \frac{1}{2}\|\mathbf{Y}_c - \mathbf{D}\mathbf{X}\|_F^2 + \lambda_1\|\mathbf{X}\|_1\\
(\mathbf{D}_0, \mathbf{X}^0) &= \arg\min_{\mathbf{D},\mathbf{X}} \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda_1\|\mathbf{X}\|_1
\end{aligned}$$

while not converge **do**

2. Update $\mathbf{X}$ and $\mathbf{X}^0$ by Algorithm 1.
3. Update $\mathbf{D}$ by ODL [35]:

$$\begin{aligned}
\mathbf{E} &= (\mathbf{Y} - \mathbf{D}_0\mathbf{X}^0)\mathcal{M}(\mathbf{X}^T)\\
\mathbf{F} &= \mathcal{M}(\mathbf{X}\mathbf{X}^T)\\
\mathbf{D} &= \arg\min_{\mathbf{D}}\{-2\text{trace}(\mathbf{E}\mathbf{D}^T) + \text{trace}(\mathbf{F}\mathbf{D}^T\mathbf{D})\}
\end{aligned}$$

4. Update $\mathbf{D}_0$ by ODL [35] and ADMM [36] (see equations (17) - (20)).

end while

end function

Difficulties

Implement the FISTA algorithm (see [3]).

Implement the ADMM algorithm (see [4]).

References

[1] T. H. Vu, V. Monga, *Fast Low-Rank Shared Dictionary Learning For Image Classification,* IEEE Trans. Image Process., vol. 26, no. 11, pp. 5160-5175, Nov. 2017.

[2] M. Yang, L. Zhang, X. Feng, and D. Zhang, *Fisher discrimination dictionary learning for sparse representation*, in Proc. IEEE International Conference on Computer Vision, pp. 543–550, Nov. 2011.

[3] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear invers problems*, SIAM J. Image. Sci, Vol. 2, No. 1, pp. 183–202, Nov. 2009.

[4] S. Boyd, N. Parith, E. Chu, B. Peleato, J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn, Vol. 3, No. 1, pp. 1–122, Nov. 2009.