Sparse Coding for Dictionary Learning in Context of Image De-noising

Dhaivat Deepak Shah
ds3267@columbia.edu

Gaurav Ahuja
ga2371@columbia.edu

Sarah Panda
sp3206@columbia.edu

Abstract

Dictionary learning involves solving the following optimization problem: in ——x —— ————m D, D 22+ 1 where is the input signal, is the dictionary and is the sparse representation of the signal. x D The problem of image restoration has been addressed with a multitude of approaches. All the approaches to solve the optimisation problem fall under the 3 broad categories of Relaxation (Basis Pursuit), Greedy approach(Matching Pursuit) or Hybrid methods. Our project primarily focuses on the Relaxation methodology. Here, both and are unknown. Mairal, Julien, et al, 2009 present an online learning D algorithm[1] which involves two optimization problem. First, assumes the to be available and D minimizes over . This is known as the sparse coding problem. Second, updates the after D obtaining . Mairal, Julien, et al, 2009 use LARS[2] to solve the sparse coding problem. We propose to compare the performance of the online dictionary learning algorithm by solving the sparse coding problem using methods[1][5] like featuresign [3], FISTA[4], Interior point, Sequential Shrinkage or Iterative Shrinkage methods and Stochastic Gradient Descent in the context of image restoration.

1 Introduction

Introduction : Problem of Image denoising

2 Intro to Dictionary learning

Intro to Dictionary learning - KSVD- general KSVD explaination - Online Dictionary Learning

3 KSVD

KSVD for learning dictionaries

4 Sparse Coding

Sparse coding problem explained in deep and ways to approximate the sparse code - Basis pursuit - Matching pursuit

5 Summary of sparse coding techniques used:

5.1 FISTA

The objective function that we have as defined above is:

$$F(x) = \frac{1}{2}\|y - Dx\|^2 + \lambda\|x\|_1$$

Here the first term, $f(x)$ is a smooth, convex function with Lipschitz continuous gradient $D^T(Dx - y)$ and $L_f = \|D^T D\|$.

FISTA, proposed by Beck [2] has a faster convergence rate as compared to ISTA. The main difference between the two is that the iterative shrinkage operator is not applied to the previous point alone but to another point which uses a specific linear combination of the previous two points. The algorithm in this case becomes:

1. L_f
2. **Step 0.** $j_1 = x_0 \,\epsilon\, \mathbb{R}^n, t_1 = 1$
3. **Step** k**.**$(k \geq 1)$
4. $\quad x_k = soft(j_k - \frac{1}{L_f}\nabla f(j_k), \frac{\lambda}{L_f})$
5. $\quad t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
6. $\quad y_{k+1} = x_k + \frac{t_k - 1}{t_k + 1}(x_k - x_{k-1})$

5.2 MP

- MP

5.3 OMP

- OMP

5.4 ALM

The Augmented Lagrangian Method was proposed individually by Powell [9] and Hestenes [5] and is explained in depth by Nocedal, Wright [12]. It is an extension to the quadratic penalty function method proposed by Courant [4]. In the penalty function method, we add a quadratic penalty term for each of the constraints in a constrained optimisation problem. That is for:

$$\min_x f(x) \text{ subj. to } c(x) = 0$$

the formulation becomes,

$$\min_x f(x) + \frac{\mu}{2}\|c(x)\|^2$$

where μ is the penalty parameter and is always positive. We can now progressively increase μ towards ∞ and try to find a minimizer for the objective function. Thus we can convert the constrained minimisation problem into an unconstrained one. However, an ill-conditioning for the Hessian in the formulation can lead to significantly poor results for the iterative methods. To reduce this possibility, we include a Lagrange multiplier to get the Augmented Lagrangian as:

$$\min_x f(x) + \frac{\mu}{2}\|c(x)\|^2 + \lambda c(x)$$

We look at both the primal and the dual formulations of this approach.

5.4.1 Primal ALM

In our case, the problem boils down to:

$$L_\mu(x, e, \lambda) = \|x\|_1 + \|e\|_1 + \frac{\mu}{2}\|y - Dx - e\|^2 + \lambda(y - Dx - e)$$

For the primal problem, Bertsekas [3] showed that there exists a λ^* and μ^* such that

$$
\begin{aligned}
e_{k+1} &= arg\min_{e} L_{\mu}(x_k, e, \lambda_k) \\
x_{k+1} &= arg\min_{x} L_{\mu}(x, e_{k+1}, \lambda_k) \\
\lambda_{k+1} &= \lambda_k + \mu(y - Dx_{k+1} - e_{k+1})
\end{aligned}
$$

Of the above equations, the one for e has a closed form solution. As for the update of x, it is a standard L_1 minimisation problem which we solve using the FISTA method explained above.

So the overall algorithm can be summarized as in [13]:

1. **Input:**$y \,\epsilon\, \mathbb{R}^m$, $D \,\epsilon\, \mathbb{R}^{mxn}$, $x_1 = 0$, $e_1 = y$, $\lambda_1 = 0$
2. **while** not converged(k = 1,2,...) **do**
3. $\quad e_{k+1} \leftarrow shrink(y - Dx_k + \frac{1}{\mu}\lambda_k, \frac{1}{\mu})$
4. $\quad t_1 \leftarrow 1, z_1 \leftarrow x_k, w_1 \leftarrow x_k$
5. $\quad$ **while** not converged (l = 1,2,...) **do**
6. $\quad\quad w_{l+1} \leftarrow shrink(z_l + \frac{1}{l}D^T(y - Dz_1 - e_{k+1} + \frac{1}{\mu}\lambda_k), \frac{1}{\mu L})$
7. $\quad\quad t_{l+1} \leftarrow \frac{1}{2}(1 + \sqrt{1 + 4t_l^2})$
8. $\quad\quad z_{l+1} \leftarrow w_{l+1} + \frac{t_l - 1}{t_l + 1}(w_{l+1} - w_l)$
9. $\quad$ **end while**
10. $\quad x_{k+1} \leftarrow w_l, \lambda_{k+1} \leftarrow \lambda_k + \mu(y - Dx_{k+1} - e_{k+1})$
11. **end while**
12. **Output:** $x^* \leftarrow x_k, e^* \leftarrow e_k$

While in the general case of Augmented Lagrangian methods, the value of μ is incremented after every iteration, we are holding it fixed to the initialisation value.

5.4.2 Dual ALM

The dual Augmented Lagrangian method for efficient sparse reconstruction was proposed by Tomioka [11]. It tries to solve the dual of the problem we have been tackling so far as:

$$\max_{j} y^T j \text{ subj. to } D^T j \,\epsilon\, \mathbb{B}_1^{\infty}$$

where $\mathbb{B}_1^{\infty} = \{x \,\epsilon\, \mathbb{R}^n : \|x\|_{\infty} \leq 1\}$

The associated Lagrangian function becomes:

$$\min_{j,z} -y^T j - \lambda^T(z - D^T j) + \frac{\mu}{2}\|z - D^T j\|^2 \text{ subj. to } z \,\epsilon\, \mathbb{B}_1^{\infty}$$

Here, again there is a simultaneous minimization w.r.t j,λ and z. So we adopt an alternation strategy to get the following algorithm as in [13]:

1. **Input:** $y \,\epsilon\, \mathbb{R}^{>}, B = [A, I] \,\epsilon\, \mathbb{R}^{mx(n+m)}, w_1 = 0, j_1 = 0$
2. **while** not converged (k = 1,2,...) **do**
3. $\quad z_{k+1} = \mathbb{P}_{\mathbb{B}_1^{\infty}}(B^T j_k + \frac{w_k}{\mu})$
4. $\quad j_{k+1} = (BB^T)^{-1}(Dz_{k+1} - (Bw_k - y)/\mu)$
5. $\quad w_{k+1} = w_k - \mu(z_{k+1} - D^T j_{k+1})$
6. **end while**
7. **Output:** $\lambda^* \leftarrow w_k[1 : n], e^* \leftarrow w_k[n + 1 : n + m], j^* \leftarrow j_k$

5.5 Feature Sign

- Feature Sign

5.6 L1LS

- L1LS

6 Experimental Setup

We have used the K-SVD approach suggested in the previous sections for Dictionary Learning, and have used above mentioned 7 solvers for benchmarking.

8 iterations of K-SVD were performed. For each of the solvers, the following parameters have been varied:

1. Starting Image Noise Levels: 10, 20, 50 dB
2. Dictionary Size: 64, 128, 256

And the comparison is based on:

1. Execution Time
2. SNR of the de-noised image

All other supplied parameters were fixed across approaches, although tuning them specifically to a particular method might have resulted in better results, in order to maintain consistency across the experiment.

K-SVD toolbox [10] and L1 Solvers [13], [6] were used for computation, and all the processes were run using matlab in nodesktop mode.

Pertinent values, images and dictionaries at each of the intermediate steps were recorded and are hosted at AML Project Results under the respective results folders.

7 Findings

8 Analysis

9 Conclusion

References

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. -svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[3] Dimitri P Bertsekas. Nonlinear programming. 1999. *Athena Scientific*.

[4] Richard Courant et al. Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc*, 49(1):1–23, 1943.

[5] Magnus R Hestenes. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5):303–320, 1969.

[6] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19:801, 2007.

[7] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, 1993.

[8] Yagyensh Chandra Pati, Ramin Rezaiifar, and PS Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993.

[9] Michael JD Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155–162, 1964.

[10] Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. *CS Technion*, page 40, 2008.

[11] Ryota Tomioka and Masashi Sugiyama. Dual-augmented lagrangian method for efficient sparse reconstruction. *Signal Processing Letters, IEEE*, 16(12):1067–1070, 2009.

[12] SJ Wright and J Nocedal. *Numerical optimization*, volume 2. Springer New York, 1999.

[13] Allen Y Yang, Zihan Zhou, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Fast l1-minimization algorithms for robust face recognition. *arXiv preprint arXiv:1007.3753*, 2010.

10 Appendix

10.1 Appendix-1

10.2 Appendix2