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

Image restoration is a widely studied problem in image processing and the common approaches to solve the problem include filtering, minimizing variations in an image etc. The problem we address in this paper is to remove noise from a given image using Dictionary Learning approach, assuming the noise to be Gaussian and additive. The approach bases itself on representing an image as a sparse representation over a good dictionary. So our problem of restoring images boils down to learning a good dictionary and finding a good representation of any given patch of an image using the dictionary. Designing dictionaries to better the above model can be done by either selecting one from a pre-specified set of linear transforms, or by adapting the dictionary to a set of training signals. Past research has shown that dictionaries adapted to training signals lead to a lot better signal restoration than the pre-specified dictionaries. Hence, we strive to learn a dictionary from the input signals for the purpose of restoration. We have used the K-SVD algorithm as our basis for dictionary learning and explore different sparse coding methods that can be used for the same. The next section gives a Literature review of the different components of Dictionary Learning. The third section talks about our experimental setting and further sections discuss the results, our analysis and conclusion.

2 Intro to Dictionary learning

Dictionary Learning is a focus area in the Signal Processing domain. A dictionary is used for the sparse representation or approximation of signals. A dictionary D, is an NxK matrix that contains K prototype signals called atoms. A signal y_0 can be represented by a linear combination of a small number of these atoms. For any signal y_0 and dictionary D there must exist a sparse vector $x \in R^K$

such that:

$$y_0 = D * x \tag{1}$$

There will be cases where we cannot exactly represent the signal with a linear combinations of atoms from a finite dictionary. In such a case we find an approximation of the signal, y such that:

$$y = D * x \tag{2}$$

$$\|y - y_0\| < \epsilon \tag{3}$$

Dictionary Learning is a problem of finding a dictionary such that the approximation of many vectors in the training set are as good as possible.

For a finite number of training vectors, L each of length N the aim of Dictionary Learning is to find both a Dictionary D, of size NxK and the L corresponding coefficients vectors of length K.

In this paper we will apply the problem of Dictionary Learning on noisy image. The image is broken into $8x8$ pixel patches and this $N = 64$ pixel vector is approximated by the linear combination of atoms of the Learned Dictionary.

2.1 Problem Statement

Classical dictionary learning techniques consider a finite training set of signals $Y = [y_1, y_2, \dots, y_L]$ int R^{NxL} and optimize the empirical cost function:

$$f(D) = \frac{1}{L} \sum_{i=1}^{L} l(y_i, D) \tag{4}$$

where $D \in R^{NxK}$ is the dictionary and l is the loss function such that $l(y, D)$ is small if D is a good at representing the signal x.

$l(y, D)$ can be define as the optimal value of the l_1 or l_0 sparse coding problem. See section on Sparse coding for more details, TODO: rephrase this. We consider the l_1 norm for illustration in this section.

$$l(y, D) = \min_{x \in R^K} \frac{1}{2} \|y - Dx\| + \lambda \|x\|_1 \tag{5}$$

λ is the regularization parameter. The problem of minimizing the empirical cost function $f(D)$ is not convex with respect to D. It can be rewritten as a joint optimization problem with respect to dictionary D and the coefficients $x = [x_1, x_2, \dots, x_L]$, which is not jointly convex, but convex with respect to each of the two variables D and x when the other one is fixed.

$$\min_{D \in R^{NxK}, x \in R^{KxL}} \frac{1}{L} \sum_{i=1}^{L} \left(\frac{1}{2} \|y_i - Dx_i\|_2^2 + \lambda \|x\|_1 \right) \tag{6}$$

A natural approach to solving this problem is to alternate between the two variables, minimizing one over the other while keeping the other fixed.

3 K-SVD for dictionary learning

The goal of K-SVD is to adapt a dictionary to the training signals such that it ensures the best sparse representation for each of the signals, under constraints of reconstruction error and sparsity. It is a two-step iterative process, which begins with a dictionary initialization. The dictionary could be initialized with any pre-calculated well known dictionary, like DCT, or created from the noisy image by randomly selecting atoms from the patches of the given image. The first step of the process determines the sparse code for the given training signals with fixed dictionary. The second step assumed fixed sparse code and updates the dictionary to better fit the data. The second step further updates the sparse representation of the training signals to accelerate convergence.

K-SVD tries to optimize the following overall MSE under sparsity constraints in each iteration:

$$\min_{d,x} \|Y - DX\|_F{}^2 \text{ subj. to, } \forall i, \|x_i\|_0 <= L$$

The algorithm is outlined below:

```
1. Initialization: Set dictionary matrix $D \in R^{nXK}$ with $l_2$ normalized columns
2. while not converged($j$ = 1,2,...) do
3.     Sparse Coding:
4.         Compute $x_i$ for every training signal $y_i$, by solving for
           $\min_{x_i} \|y_i - Dx_i\|_F{}^2$ subject to $\|x_k\|_0 \leq L$
5.     Dictionary Update:
6.         Update each atom k in 1,2,..K in $D_{j-1}$ as below:
   (a)         Let $w_k$ be the set of input signals that use this atom D(k).
   (b)         Find the overall error in representation, in matrix $E_k$,
               $E_k = Y - \sum(j \neq k) d_j x_j{}^2$
   (c)         Restrict $E_k$ by choosing columns corresponding to $w_k$ as $E_k^R$
   (d)         Apply SVD decomposition on the restricted error matrix :
               $E_k^R = U\delta V^T$
               Set $d_k = U[: 1]$
               and $x_r^k = V[: 1] * \delta(1, 1)$
7.         j= j+1
8. end while
9. Output: $X^* \leftarrow X_j, D^* \leftarrow D_j$
```

In this paper, we focus on the first step of K-SVD and that is the sparse coding step. The following sections discuss the problem of sparse coding in detail along with the algorithms we have used.

4 Sparse Coding

As mentioned before, any signal y can be represented as a linear combination of atoms in a dictionary D. Sparse coding problem aims at finding the sparsest such representation. On a high level, those atoms summarize the most important features in the input.

$$\min \|x\|_0 \text{ subj. to } y = Dx$$

The above equation tries to find the sparsest coefficient vector that exactly reconstructs the input signal. However, this form of exact determination is NP-hard. So, to make it easier to solve, the problem needs to be relaxed by allowing for a small amount of reconstruction error, ϵ. The problem can be thus formulated as,

$$\min \|x\|_0 \text{ subj. to } \|y - Dx\| \leq \epsilon$$

Still the problem is computationally expensive and following are two of the well known approaches to approximate the solution to the above.

4.1 Greedy (Matching pursuit)

Greedy methodology, on the other hand, solves the problem in (2) by sequentially selecting the atom that best explains the data. Upon convergence,

4.2 Relaxation (Basis pursuit)

In the relaxation methodology, the l_0 norm in (2) is replaced with the l_1 norm, resulting in the following equivalent objective,

$$\min \|y - Dx\|^2 + \lambda \|x\|_1$$

L1 regularization penalizes the non-sparse terms based on their actual values, which may seem unfair, but past research has shown that it leads to a very good approximation and often leads to the sparsest representation. The resulting optimization problem is convex and unconstrained, and hence can be solved using several methods like gradient descent.

5 Summary of sparse coding techniques used:

5.1 MP

Matching pursuit, proposed by Mallat and Zhang et al [4] is a well known greedy algorithm for learning sparse code of a signal over a given dictionary. The basic idea of the algorithm is outlined below.
At every step of refining the sparse code, the algorithm picks an atom of the dictionary that best represents a portion of the input signal.

The following pseudo-code briefly describes the algorithm:

```
1. Input: Training signal y, Dictionary - D
2. Initialization: $R_1 = y, x = 0, n = 1$
3. while $R_n >$ Threshold do
4.     Find $d_i \in$ D with max correlation with residual i.e. $< R_n, d_i >$
5.     $x_i = x_i + < R, d_i >$
6.     $R_{n+1} = R_n - x_i d_i$
7.     $n = n + 1$
8. end while
9. Output: Sparse code of y over D, $x^* \leftarrow x_n$
```

The algorithm evaluates the best matching atom at every step by evaluating the inner product between the Residual, R_n and the individual atoms from the dictionary. The coefficient vector x is updated with the atom that has maximum inner product and its contribution to the signal is deducted from the signal. The process repeats until the residual falls below a threshold. The sparse coefficients thus learned is believed to be the sparsest approximation of the decomposition of the signal over the dictionary.

5.2 OMP

A variation of the Matching Pursuit is the Orthogonal Matching Pursuit [5], also a greedy method that iteratively selects the atom having highest correlation with the residual. The difference is that after adding the best matched atom to the coefficient vector, the residual is updated with the orthogonal projection of the input signal y onto the subspace of previously selected atoms. This set of selected atoms is called the active set and since the resulting residual is orthogonal to each of the atoms in the set, none of those atoms are selected in the future iterations.

The OMP algorithm is as below:

```
1. Input: Training signal y, Dictionary - D
```

```
Initialization: $R_0 = y, X(c_0) = 0, n = 0$
while not converged do
    Solve for $X_{t_i}$ by minimizing,
    $\max_t {X_t}^T R_{n-1}$
    Add $X_{t_i}$ to the set of selected atoms; $c_n = c_{n-1} \cup t_i$
    Let $P_n = X(c_n)(X(c_n)^T X(c_n))^{-1} X(c_n)^T$ denote the projection onto the linear space
    spanned by the elements of $X(c_n)$.
    Update $R_n = (I - P_n)y$.
    $n = n + 1$
end while
Output: Sparse code of y over D, $x^* \leftarrow x_n$
```

5.3 FISTA

The objective function that we have as defined above is:

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

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 [**?**] 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:

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

5.4 ALM

The Augmented Lagrangian Method was proposed individually by Powell [**?**] and Hestenes [**?**] and is explained in depth by Nocedal, Wright [**?**]. It is an extension to the quadratic penalty function method proposed by Courant [**?**]. 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,

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

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:

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

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 [**?**] 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 [6]:

```
Input:$y \, \epsilon \, \mathbb{R}^m$, $D \, \epsilon \, \mathbb{R}^{mxn}$, $x_1 = 0$, $e_1 = y$, $\lambda_1 = 0$
while not converged($k$ = 1,2,...) do
    $e_{k+1} \leftarrow shrink(y - Dx_k + \frac{1}{\mu}\lambda_k, \frac{1}{\mu})$
    $t_1 \leftarrow 1, z_1 \leftarrow x_k, w_1 \leftarrow x_k$
    while not converged ($l$ = 1,2,...) do
        $w_{l+1} \leftarrow shrink(z_l + \frac{1}{l}D^T(y - Dz_1 - e_{k+1} + \frac{1}{\mu}\lambda_k), \frac{1}{\mu L})$
        $t_{l+1} \leftarrow \frac{1}{2}(1 + \sqrt{1 + 4t_l^2})$
        $z_{l+1} \leftarrow w_{l+1} + \frac{t_l - 1}{t_l + 1}(w_{l+1} - w_l)$
    end while
    $x_{k+1} \leftarrow w_l, \lambda_{k+1} \leftarrow \lambda_k + \mu(y - Dx_{k+1} - e_{k+1})$
end while
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 [**?**]. 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 [6]:

```
Input: $y \, \epsilon \, \mathbb{R}^{>}, B = [A, I] \, \epsilon \, \mathbb{R}^{mx(n+m)}, w_1 = 0, j_1 = 0$
while not converged ($k$ = 1,2,...) do
    $z_{k+1} = \mathbb{P}_{\mathbb{B}_1^\infty}(B^T j_k + \frac{w_k}{\mu})$
    $j_{k+1} = (BB^T)^{-1}(Dz_{k+1} - (Bw_k - y)/\mu)$
    $w_{k+1} = w_k - \mu(z_{k+1} - D^T j_{k+1})$
end while
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 aims to solve the following L-1 regularized sparse coding problem

$$\min_x f(x) = \|y - Dx\|^2 + \lambda\|x\|_1 \tag{7}$$

The intuition behind this algorithm is that if we know the signs(positive, negative or zero) of x_i at the optimal value we can replace $\|x_i\|$ by either x_i if $x_i > 0$, $-x_i$ if $x_i < 0$ or 0 if $x_i = 0$. Considering only non zero coefficients this problem reduces to a standard quadratic optimization problem. This problem can them be solved analytically and efficiently. Feature sign algorithm tries to search for the signs of the coefficients. Given any such search results, it efficiently solves the resulting unconstrained quadratic problem. Further the algorithm systematically refines the guess if it turns out to be initially incorrect.

1. **Initialize** $x := 0$, $\theta : 0$ and active set $:= \{\}$, $\theta_i \in \{-1, 0, 1\}$ denotes sign(x_i)
2. From zero coefficients of x, select $i = \text{argmax}_i \, |\frac{\partial\|y-Dx\|^2}{\partial x_i}|$
   Activate x_i (add i to active set) only if it locally improves the objective
   (a) If $|\frac{\partial\|y-Dx\|^2}{\partial x_i}| > \lambda$; set $\theta_i := -1$, active set $:= \{i\} \cup$active set
   (b) If $|\frac{\partial\|y-Dx\|^2}{\partial x_i}| < -\lambda$; set $\theta_i := 1$, active set $:= \{i\} \cup$active set
3. **Feature-Sign step:**
   Let $\hat{D}$ be a submatrix of D that contains only the columns corresponding to the active set.
   Let $\hat{x}$ and $\hat{\theta}$ be sub vectors of x and θ corresponding to the active set.
   Compute the analytical solution to the resulting quadratic problem
   $\hat{x}_{new} := (\hat{D}^T\hat{D})^{-1}(\hat{D}^T y - \lambda\hat{\theta}/2)$
   Perform a discrete line search on the closed line segment from $\hat{x}$ to $\hat{x}_{new}$:
   - Check the objective value at $\hat{x}_{new}$ and all points where any coefficient changes sign.
   - Update $\hat{x}$ (and all corresponding entries in x) to the point with the lowest objective value
   Remove zero coefficients of $\hat{x}$ from the active set and update $\theta :=$sign(x)
4. **Check the optimality conditions**:
   (a) Optimality condition for nonzero coefficients: $|\frac{\partial\|y-Dx\|^2}{\partial x_i}| + \lambda\text{sign}(x_i) = 0 \; \forall x_i \neq 0$
   If this condition is not satisfied, go to Step 3(without any new activation); else check condition (b).
   (b) Optimality condition for zero coefficients: : $|\frac{\partial\|y-Dx\|^2}{\partial x_i}| + \lambda\text{sign}(x_i) \leq 0 \; \forall x_i = 0$
   If this condition is not satisfied go to Step 3; otherwise return x as the solution.

5.6 TNIPM

In [2], the authors formulate a Truncated Newton Interior Point method. The algorithm solves the L-1 regularised problem:

$$\min_x f(x) = \|y - Dx\|^2 + \lambda\|x\|_1 \tag{8}$$

This problem can be transformed to a convex quadratic program with linear inequality constraints,

$$\min \|y - Dx\|^2 + \sum_{i=1}^{K} \lambda u_i \tag{9}$$

$$\text{subject to:} - u_i \leq x_i \leq u_i, \forall i \in \{i, K\} \tag{10}$$

The aim of this algorithm is to solve this quadratic program using the Newton's method. Since solving the Newton system exactly is computationally expensive for large L-1 problems, TNIPM uses an iterative method, Preconditioned Conjugate Gradient (PCG) to approximately solve the Newton system and develops an interior point method to solve the L-1 problem.

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. The dictionary has been initialised with a DCT dictionary in each of the cases.

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 [**?**] and L1 Solvers [6], [3] 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