Sparse Representation and Orthogonal Matching Pursuit Algorithm with Application to Image Denosing

Sek Cheong, Yihan Li

August 08 2019

Abstract

Sparse solutions to an underdetermined system of linear equations can be computationally tractable [1]. This leads to many interesting applications in signal/image processing and computer vision problems such as denoising[2], restoration, inpainting [5], compression and classification. The class presents relaxation techniques such LASSO for obtaining sparse solutions. This project introduces the concept of dictionary learning, the orthogonal matching pursuit (OMP) algorithm to solve for sparse solutions, and an example application in image denoising using the KSVD algorithm. Users would learn the basic idea of dictionary learning like dictionary and sparse representation. Users would also be able to understand and conduct OMP given a dictionary by following the instructions.

Background

We'll start by a simple example about representation of any vector in XY-coordinate system. In this example, the signal we're trying to represent is any vector $\boldsymbol{y} \in \mathbb{R}^2$. To represent it, we need some other vectors which in together define this coordinate system. For example, we can choose these vectors to be $(0,1)$ and $(1,0)$. This would give the representation of $\boldsymbol{y}$ to be $\boldsymbol{y} = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^T$. Let $\boldsymbol{y}$ be the measured signal, the matrix $\boldsymbol{D}$, whose columns forms the basis for the signal, the dictionary, and a sparse vector $\boldsymbol{\alpha}$. The signal $\boldsymbol{y}$ is a weighted sum of columns in $\boldsymbol{D}$ and can be written as following:

$$\boldsymbol{y} = \boldsymbol{D\alpha}$$

In this example,

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \boldsymbol{D} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \boldsymbol{\alpha} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}.$$

We can see that the representation vector $\boldsymbol{\alpha}$ is dependent on the choice of dictionary $\boldsymbol{D}$. If we change our dictionary $\boldsymbol{D}$ to

$$\boldsymbol{D} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

Then our representation vector $\boldsymbol{\alpha}$ needs to be

$$\boldsymbol{\alpha} = \begin{bmatrix} \frac{y_1+y_2}{2} \\ \frac{y_1-y_2}{2} \end{bmatrix}$$

Sparse representation has been an attractive field over the past couple of decades. It seeks to represent a signal $\boldsymbol{y}$ by using as few basic vectors, which are columns of a dictionary, as possible. In other words, it tries to find a sparse solution $\boldsymbol{\alpha}$ to the equation below given a measured signal $\boldsymbol{y}$

$$\boldsymbol{y} = \boldsymbol{D\alpha}$$

In practice, the measured signal $\boldsymbol{y}$ is usually high dimensional and probably noisy. A sparse representation vector $\boldsymbol{\alpha}$ can represent the signal concisely given a dictionary $\boldsymbol{D}$. This would help us to get information from the measured signal and process the information like compressing and encoding more easily. This would lead to many interesting applications such as denoising[2], restoration, inpainting [5], compression and classification.

The idea of sparse representation was put forward by Stephane Mallat [3] in 1993 in his research on wavelet signal processing. Mallat also proposed an iterative algorithm called Matching Pursuit for finding sparse solutions. Based on Mallat's ideal, Pati et al. proposed the Orthogonal Matching Pursuit (OMP) algorithm which has been proven to be faster and easier to implement[6]. Since OMP is critical to finding sparse solutions, we would like to explore more into the basic concept of OMP.

As we've seen before, the representation vector, or the solution is dependent on the choice of dictionary. It is true that in practice both $\boldsymbol{D}$ and $\boldsymbol{\alpha}$ are unknown and sparse representation algorithms will iteratively update both of them until convergence. For illustration proposes we will focus on finding the sparse solution $\boldsymbol{\alpha}$ for a given dictionary $\boldsymbol{D}$ using OMP. In real application both the dictionary and sparse are both unknown. The iterative method need to find both the optimal dictionary and sparse vector. A popular algorithm for both is KSVD [4] which we will see in the Image Denosing section.

Though OMP and sparse representation are powerful tools which could help us do a series of image processing, malicious use of these techniques became more hazardous and destructive. A logo or a watermark is usually a declaration of copyright for an image. This could prevent the images from being pirated while at the same time being shared with online users. However, image inpainting, which is an application of sparse representation can remove the logo or watermark easily, which impairs the owners of the copyright. Another issue is the recent topic deepfake, which could replace a person's face by any other's in a photo or even video. The misuse can result in severe invasion of privacy.

Orthogonal Matching Pursuit (OMP) Algorithm

Here we are trying to solve the minimization of the l_0 problem:

$$\min ||\boldsymbol{\alpha}||_0 \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{D\alpha}$$

However, the $\min ||\boldsymbol{\alpha}||_0$ is non-convex, therefore an iterative algorithm (OMP) is used. In the formulation above we want to enforce the sparsity by solving $\min ||\boldsymbol{\alpha}||_0$ subject to the constraint $\boldsymbol{y} = \boldsymbol{D\alpha}$. The matrix $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is the dictionary and solution $\boldsymbol{\alpha} \in \mathbb{R}^{N\times 1}$ is the sparse vector. We can represent $\boldsymbol{D}$ as a matrix of columns vector $\boldsymbol{d_1}, \boldsymbol{d_2}, \cdots \boldsymbol{d_N}$:

$$\boldsymbol{D\alpha} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \boldsymbol{d_1} & \boldsymbol{d_2} & \cdots & \boldsymbol{d_N} \\ \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}$$

The term "matching" means find the column in $\boldsymbol{D}$ that has the largest correlation or projection with $\boldsymbol{y}$. We begin by choose column j which maximizes projection of $\boldsymbol{y}$ onto this column. The selected column can be viewed as the most similar vector to $\boldsymbol{y}$ that one can find from the columns in the dictionary.

$$i_{(1)} = \arg\max_j \boldsymbol{d_j}^T \boldsymbol{y}$$

We then build our basis matrix $\boldsymbol{A}$ begin with the first iteration.

$$\boldsymbol{A}_{(1)} = \left[\boldsymbol{d}_{i(1)}\right]$$

Next we find the $\boldsymbol{\alpha}^{(1)}$ that minimizes the error given the basis $\boldsymbol{A}$ by solving the least squares problem:

$$||\boldsymbol{y} - \boldsymbol{A}_{(1)}\boldsymbol{\alpha}^{(1)}||_2^2$$

$$\hat{\boldsymbol{\alpha}}^{\mathbf{(1)}} = (\boldsymbol{A}_{(1)}^T\boldsymbol{A}_{(1)})^{-1}\boldsymbol{A}_{(1)}^T\boldsymbol{y}$$

Here $\hat{\boldsymbol{\alpha}}^{(1)}$ is the estimation of $\boldsymbol{\alpha}$ in the first iteration in terms of basis $A_{(1)}$. The residue after the first iteration is:

$$\boldsymbol{r}_{(1)} = \boldsymbol{y} - \boldsymbol{A}_{(1)}\hat{\boldsymbol{\alpha}}^{(1)}$$

After the first iteration we choose the column that has the maximum projection on the residue:

$$\boldsymbol{i}_{(2)} = \arg\max_{j} \boldsymbol{d}_{\boldsymbol{j}}^T\boldsymbol{r}_{(1)}$$

We than augment the basis matrix $\boldsymbol{A}$ with $\boldsymbol{i}_{(2)}$ that is:

$$\boldsymbol{A}_{(2)} = \begin{bmatrix}\boldsymbol{d}_{i_{(1)}} & \boldsymbol{d}_{i_{(2)}}\end{bmatrix}$$

Once again we use the basis $\boldsymbol{A}$ to solve the least squares problem:

$$||\boldsymbol{y} - \boldsymbol{A}_{(2)}\boldsymbol{\alpha}^{(2)}||_2^2$$

$$\hat{\boldsymbol{\alpha}}^{(2)} = (\boldsymbol{A}_{(2)}^T\boldsymbol{A}_{(2)})^{-1}\boldsymbol{A}_{(2)}^T\boldsymbol{y}$$

The residue after the second iteration is:

$$\boldsymbol{r}_{(2)} = \boldsymbol{y} - \boldsymbol{A}_{(2)}\hat{\boldsymbol{\alpha}}^{(2)}$$

We repeat this process k times by carrying over the residue to the next iteration until the difference in residue in succession is smaller than or equal to a threshold ϵ:

$$||\boldsymbol{r}_{(k)} - \boldsymbol{r}_{(k-1)}||_2^2 \leq \epsilon$$

After k iteration we have:

$$\hat{\alpha}^{(k)} = \begin{bmatrix}\hat{\alpha}_1 \\ \hat{\alpha}_2 \\ \vdots \\ \hat{\alpha}_k\end{bmatrix} \begin{matrix}\dashrightarrow \\ \dashrightarrow \\ \vdots \\ \dashrightarrow\end{matrix} \begin{bmatrix}\boldsymbol{d}_{i_{(1)}}^T \\ \boldsymbol{d}_{i_{(2)}}^T \\ \vdots \\ \boldsymbol{d}_{i_{(k)}}^T\end{bmatrix}$$

Set $\hat{\alpha}_1, \hat{\alpha}_1, \cdots, \hat{\alpha}_k$ at $i_{(1)}, i_{(2)}, \cdots i_{(k)}$ respectively and rest of the entries of $\hat{\boldsymbol{\alpha}}$ to 0.

Warm-Up Activity

1. What is a dictionary in context of machine learning?

2. Suppose we have a signal vector $\boldsymbol{y} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$ and a dictionary $\boldsymbol{D} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Find the representation vector $\boldsymbol{\alpha}$ such that
$$\boldsymbol{y} = \boldsymbol{D\alpha}$$

3. Find the representation vector $\boldsymbol{\alpha}$ if the dictionary is $\boldsymbol{D} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. What if the dictionary is $\boldsymbol{D} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$?

4. Suppose that we run OMP for a dictionary and got a solution $\boldsymbol{\alpha}$ with 1 nonzero. What is the sparsity of this vector? What property does the column of the dictionary corresponding to this nonzero element?

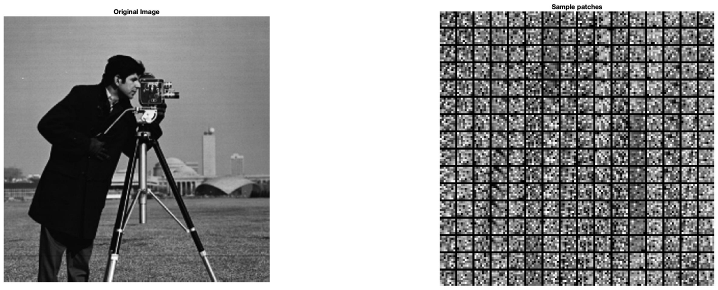
Activity

1. Prove that at each each step, the residual vector is orthogonal to the columns of the dictionary that have been selected with nonzero entries. This is why we call the algorithm ***orthogonal*** matching pursuit. (Hint: Recall that the form of the projection vector which project a vector to the space orthogonal to columns of $\boldsymbol{A}$ is $\boldsymbol{p_{A\perp} = I - A(A^T A)^{-1} A^T}$).

2. The code `OMPtest.m` is a simple implementation of the OMP algorithm. There is a 10-by-10 full rank matrix $\boldsymbol{D}$ stored in `D.mat` and a 10-by-1 vector $\boldsymbol{y}$ stored in `y.mat`. The function calculates the corresponding solution to the input sparsity level k and an error vector stores the two norms of residuals from each step. Sparsity level k predefined the sparsity level of the solution, or the largest number of iterations the algorithm will run. Set $k = 10$ first and run the code. What did you notice about the trend in residuals? Set $k = 5$ and run the code again. What's the two norm of the residual after 5 iterations? What's the sparsity of the solution after 5 iterations? When does the error decrease to less than 10% of the original error?

Image Denosing

As mentioned in the background section sparse representation is a emerging filed that has many applications in signal/image processing, computer vision and information theory. For simplicity we will work on gray level images only. Basically, we can model gray level image as an $N \times M$ matrix, where N is number of rows and M is number of columns in the image. You can extend the concept to RGB image by using three $N \times M$ matrices with each matrix representing the red, blue, and green component of the image.



(a) The Camera Man

(b) The first 64 patches

Figure 1: The Camera Man image and its patches

In denoising and computer vision, it is common to represent an image as a linear combination of vectors in $\mathbb{R}^n$, to be more concrete an image can be represented as a linear combination of the small image patches of $\sqrt{n} \times \sqrt{n}$, n is typically 64 and therefore the patch size is typically 8×8 pixels. Figure 1 (a), shows the original picture and Figure 1 (b), shows the first 64 image patches extracted from the picture. The actual number of patches is proportion to $O(M \times N)$.

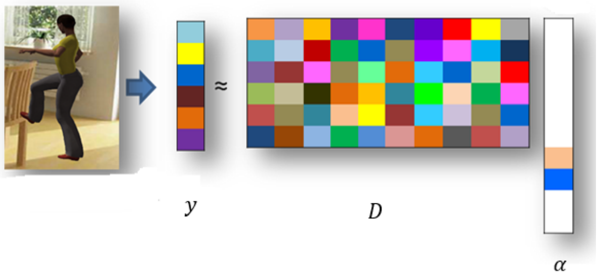


Figure 2: Sparse coding of an image

Let's look at the model $\boldsymbol{y} = \boldsymbol{D}\alpha$ and interpreter it as a way of constructing the image signal $\boldsymbol{y}$. A simple visualization is shown in Figure 2. The multiplication of $\boldsymbol{D}$ by a sparse vector $\boldsymbol{\alpha}$

with $||\boldsymbol{\alpha}||_0^0 = k_0 \ll n$ produces a linear combination of k_0 atoms which generating the image $\boldsymbol{y}$.

A measured and probably degraded image $\boldsymbol{y}$ can be represented as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{v}$$

Where $\boldsymbol{x}$ is the original image we're trying to recover, $\boldsymbol{H}$ is a degradation operator, and $\boldsymbol{v}$ is an additive linear noise term. If the degradation $\boldsymbol{H}$ matrix is set to the identity matrix $\boldsymbol{I}$ we have $\boldsymbol{Y} = \boldsymbol{x} + \boldsymbol{v}$ and the problem reduces to the denoise problem. To perform denosing is to recover the original signal $\boldsymbol{x}$ from the measured signal $\boldsymbol{y}$. The idea of the sparse representation is to represent the reconstructed, or the recovered image, $\hat{\boldsymbol{x}}$, by the product of a fixed and overcomplete matrix (also known as dictionary) $\boldsymbol{D}$ and a sparse and random vector $\hat{\boldsymbol{\alpha}}$

$$\hat{\boldsymbol{x}} = \boldsymbol{D}\hat{\boldsymbol{\alpha}}$$

To find the optimal $\hat{\boldsymbol{\alpha}}$ given $\boldsymbol{D}$ we need to solve the following optimization problem:

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \, ||\boldsymbol{\alpha}||_0 \quad \text{s.t.} \quad ||\boldsymbol{y} - \boldsymbol{D}\boldsymbol{\alpha}|| \leq \epsilon$$

$$||\hat{\boldsymbol{\alpha}}||_0 \leq ||\boldsymbol{\alpha}||_0 \implies \hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}$$

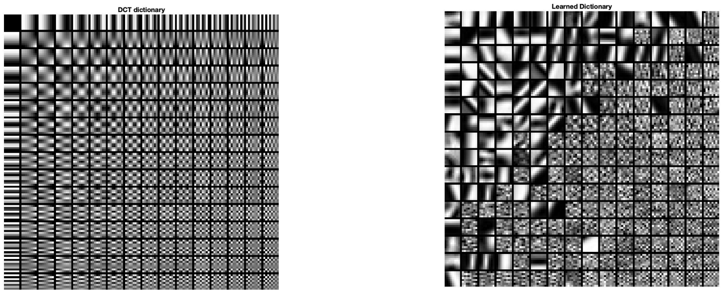
where ϵ is a is a predefined error bound that controls the deviation of the observed image $\boldsymbol{y}$ from the estimated original image $\hat{\boldsymbol{x}}$.

In previous sections, we discussed the OMP algorithm for solving the sparse vector $\hat{\boldsymbol{\alpha}}$ of a known dictionary and commonly used dictionaries includes wavelets, Fourier transformation basis, and Discrete Cosine Transformation (DCT) basis (as shown in Figure 3 (a)). However, such basis may be optimal for certain signals but they may not be the best choice for all kinds of signals, therefore, depending on the application, a data dependent dictionary may be the best choice. To find a data depend dictionary we need to reformulate our sparse problem to simultaneously solve the sparse vector $\hat{\boldsymbol{\alpha}}$ and the dictionary $\boldsymbol{D}$:

$$\boldsymbol{D}, \hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{D}, \hat{\boldsymbol{\alpha}}}{\operatorname{argmin}} \, ||\boldsymbol{y} - \boldsymbol{D}\boldsymbol{\alpha}||_F^2 \quad \text{s.t.} \quad ||\hat{\boldsymbol{\alpha}}||_0 \leq ||\boldsymbol{\alpha}||_0$$

One of the method for solving the problem above is KSVD. The mathematical details of KSVD is beyond the scope of this project, one can refer to the literature in the references section. Basically, the KSVD algorithm attempts to minimize the cost function iteratively, by first finding sparse vector using the OMP algorithm with an initial estimate of the dictionary. This sparse vector minimizes the error in estimation, and at the same time maintain a sparsity constraint as defined in the equation above. Once this sparse coding stage is done, the algorithm proceeds to update the atoms of the dictionary, one atom at a time, such that the error term is further reduced. Proceeding in such an iterative method, the algorithm reduces the error of estimation at each iteration.

For image processing tasks it is common to use the DCT basis, shown in Figure 3 (a), as the starting dictionary, as the algorithm converges, the DCT basis will look more like patches of edges and textures in the target image as shown in Figure 3 (b).



(a) The 2D discrete cosine basis

(b) The learned dictionary from “The Camera Man”

Figure 3: The dictionary

Let’s now explorer image denosing with KSVD alogrithm. In your Matlab environment please change the current directory to `"sparse_rep/ksvd_denoise"` directory and open the `"demo.m"` file. Note, each code section is delineated by a double ampersand `"%%"` sign. You can click on the section and press the `"Run Section"` button on the Matlab’s toolbar to run only the selected section.

1. Select and run the `"Read the image data"` section to view the image. You can change the `"imgfile"` variable to use a different image in the directory.
2. Now, let’s add some Gaussian noise to the image. Select and run the `"Add Gaussian noise with sigma 20"` section. You can experiment different values of σ for the Gaussian noise by changing the `"sigma"` variable. What is the signal to noise ratio (SNR) when `"sigma=20"` and `"sigma=35"`?
3. Let’s run the entire program by clicking the `"Run"` button on your Matlab’s toolbar.
    i. Inspect the noisy image and the denoised image and comment.
    ii. Does the error (RMSE) decrease with each iteration?
    iii. What is the SNR on the denoised image?
    iv. Inspect the initial dictionary (DCT) and the learned dictionary. Give brief comment.

Appendix

Solutions to Warm-Up Activity

1. A dictionary is a matrix whose columns are basic vectors, which we call atoms, containing basic information to construct a vector in the same dimension of its column vector.
2. $\boldsymbol{\alpha} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$
3. $\boldsymbol{\alpha} = \begin{bmatrix} 2.5 \\ 0.5 \end{bmatrix}$ in the first case, and $\boldsymbol{\alpha} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ in the second case.
4. The sparsity is 1. Since the column is the first column OMP selected, it is the most similar to our target vector among all columns.

Solutions to Activity

1. We denote by $\boldsymbol{A}$ the columns that have been selected by OMP, and residual vector at this step to be $\boldsymbol{r}$. Note that the solution vector $\boldsymbol{\alpha}$ at this step is updated by solving the least squares problem
   $$||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\alpha}||_2^2$$
   Then the solution vector should be
   $$\boldsymbol{\alpha} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y}$$
   And residual vector is
   $$\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y}$$
   which is the projection of $\boldsymbol{y}$ to the space orthogonal to the span of columns of $\boldsymbol{A}$.
2. The residual error decreases with more iterations. The decreasing speed is fastest at the beginning, and then slows down with more iterations. This suggest that OMP finds the most related or similar vectors first. The two norm of the residual after 5 iterations is 4.6. The sparsity of the solution is 5, which is controlled by the value of k. After 7 iterations, two norm of the residual is 1.37, which is less than 10% of the original error 19.62.

Solutions to Image Denoising

2. 16.3dB and 11.68dB

3. i. The the denoised image has a higher SNR compare to the noisy image. However, the denoised image did lose some sharpness compare to the original image.

   ii. Yes the error decreases with each iteration.

   iii. The denoised image has a measurable improvement in SNR. The SNR in the denoised image is 22.35dB.

   iv. The learned dictionary adopted a lot of features from the image. Specifically, it has learned some patches of edges, textures and patterns from the image.

References

[1] Michael Elad. *Sparse and Redundant Representations From Theory to Applications in Signal and Image Processing.* Springer New York Dordrecht Heidelberg London, 2010.

[2] Michael Elad and Michal Aharon. Image Denoising via Sparse and Redundant Representations Over Learned Dictionaries. *IEEE Transactions on image processing*, 15(12), 2006.

[3] Stphane Mallat and Zhifen Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12), 1993.

[4] Michael Elad Michal Aharon and Alfred Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54(11), 2006.

[5] Bin Shen and Wei Hu et la. Image Inpainting via Sparse Representation. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009.

[6] Joel A. Tropp and Anna C. Gilbert. Signal Recovery from Random Measurements via Orthogonal Matching Pursuit. *IEEE Transactions on information theory*, 53(12), 2007.