IMAGE DENOISING AND SUPER-RESOLUTION USING K-SVD

Sumanth Varambally

Technical University of Denmark

ABSTRACT

This project aims to examine the K-SVD algorithm, an algorithm for designing overcomplete dictionaries for sparse representation and apply it for image denoising and image super-resolution. In the course of the project, both the classic K-SVD as described in the original paper by Aharon et. al and the Approximate K-SVD algorithm by Rubinstein et. al are implemented. Furthermore, approximate K-SVD is used to solve the problem of image-denoising and image super-resolution.

1. INTRODUCTION

There is a growing interest in the problem of finding a sparse representation for a signal. The general problem can be stated as follows: Given a signal $\mathbf{y} \in \mathbb{R}^n$, design an overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{n \times K}$ and a sparse representation $\mathbf{x} \in \mathbb{R}^K$ such that $\mathbf{y} = \mathbf{Dx}$ (exact) or $\mathbf{y} \approx \mathbf{Dx}$ (approximate). One approach for designing the dictionary is based on choosing a pre-specified transform matrix like a wavelet transform. However, the approach taken by K-SVD is to learn the dictionary based on sparse-representation of the training signals [1].

Given a set of training signals $\{y_i\}_{i=1}^N$, K-SVD aims to find the dictionary $\mathbf{D}$ that solve the problem:

$$\min \|\mathbf{Y} - \mathbf{DX}\|_2^2, \text{subject to} \|x_i\|_0 < T_0, i = 1, 2..M$$

The K-SVD algorithm, as described in the paper by Aharon et.al [1], is an iterative algorithm that alternates between sparse-coding and dictionary update. An improved version of the algorithm was proposed by Rubinstein et. al [2] which replaced the SVD step with a simpler, more efficient update step. In the course of this project, the latter was implemented during the tests.

This project also examines the application of K-SVD to the problems of image denoising and image super-resolution.

2. A DESCRIPTION OF THE K-SVD ALGORITHM

This section is a summarization of the K-SVD algorithm as described in [1]. As described above, the K-SVD algorithm seeks to solve the problem:

$$\min \|\mathbf{Y} - \mathbf{DX}\|_2^2, \text{subject to} \|x_i\|_0 < T_0, i = 1, 2..M$$

It does so iteratively in two steps, the sparse-coding step and the dictionary update step.

In the first step, the algorithm solves the problem of sparse-coding, i.e. given a dictionary $\mathbf{D}$, find the sparse representation $\mathbf{X}$ satisfying:

$$\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{DX}\|_2^2, \text{subject to} \|x_i\|_0 < T_0, i = 1, 2...M$$

The penalty term can be written as:

$$\|\mathbf{Y} - \mathbf{DX}\|_2^2 = \sum_{i=1}^{M} \|y_i - \mathbf{D}x_i\|_2^2$$

Hence, the above can be decomposed into M smaller problems:

$$\min_{x_i} \|y_i - \mathbf{D}x_i\|_2^2, \text{subject to} \|x_i\|_0 < T_0, i = 1, 2...M$$

Note that this problem is solved using the method of Orthogonal Matching Pursuit (OMP) [2], the details of which are beyond the scope of this document. Hence, the sparse-coding step can be solved easily.

The problem of updating the dictionary is a bit more involved. Assume we are updating the k^{th} colum of $\mathbf{D}$. Note that the notation x_T^j is used to indicate the j^{th} row in $\mathbf{X}$. The penalty term can be re-written as:

$$\|\mathbf{Y}-\mathbf{DX}\|_2^2 = \|\mathbf{Y}-\sum_{j=1}^{M} d_j x_T^j\|_2^2 = \|\mathbf{Y}-(\sum_{j \neq k} d_j x_T^j)-d_k x_T^k\|_2^2$$

$$= \|\mathbf{E}_k - d_k x_T^k\|_2^2$$

In order to maintain the sparsity constraint, only the sparse codes $x_R^k = x_T^k(i)$ which use atom d_k are considered. Similarly, Y_k^R is considered, which is the subset of examples that currently use the d_k atom. A similar shrinkage is considered with E_k^R, which corresponds to a selection of error columns that correspond to the examples that use atom d_k. Now the problem is equivalent to the minimization problem:

$$min_{d_k, x_R^k} \|\mathbf{E}_k^R - d_k x_R^k\|_2^2$$

which can be solved using SVD.

3. EFFICIENT IMPLEMENTATION USING APPROXIMATE K-SVD

An improvement to the above algorithm was proposed by Rubinstein, et.al [2], which has been listed below. This was the method implemented in the project. In this method, an explicit computation of the matrix E and the subsequent SVD operation are avoided. However, the details of the algorithm are beyond the scope of this document.

Algorithm 1: Approximate K-SVD

```
Result: Output dictionary D and sparse
        representation X
Initialize D as a matrix of random numbers. K is the
  number of iterations, L is the dictionary size ;
for n ← 1 to K do
    ∀i : x_i = arg min_γ ||y_i − Dx_i||_2^2
    subject to ||γ||_0 < T_0
    for j ← 1 to L do
        D_j := 0
        I := {indices of the signals in Y whose
          representations use d_j}
        g := X_{j,I}^T
        d := X_I g − DX_I g
        d := d / ||d||_2
        g := X_I^T d − (DX_I)^T d
        D_j := d
        X_{j,I} := g^T
    end
end
```

---

4. APPLICATION TO IMAGE DENOISING AND IMAGE SUPER-RESOLUTION

4.1. Image Denoising

Image Denoising refers to the process of removing noise from an image. This is achieved using K-SVD by building a dictionary of image patches extracted from the noisy image and building a sparse representation for it. The reconstructed image is then simply obtained by multiplying the sparse representation of the noisy image and the obtained dictionary. The idea here is that by building a sparse enough representation of the image, the high-frequency noise is filtered away, while the 'low-frequency' details of the image are preserved.

An image of suitably high resolution is used. The RGB colour values are first normalized to be between 0 and 1, and Gaussian noise with $\sigma = 0.2$ is added to the pixel values. The image is then partitioned into patches of size 4×4 in the first example and 8×8 in the second. and fashioned into a matrix of size $3P^2 \times Q$ where $P = 8$ is the patch size and $Q = \frac{M \times N}{P^2}$, where $M \times N$ is the image resolution. This is the signal set Y. A dictionary of size 2000 is learnt with a sparsity of 2. While in the second example, the dictionary was trained on the original noisy image, in the first example, three images of faces are considered as part of the 'training set'. These images are added with Gaussian noise and used to build the dictionary. Then, an 'untrained' noisy image is denoised, by running the OMP algorithm on the noisy image with the learnt dictionary. The reconstructed image is obtained simply by computing the product $Y_{\text{reconstructed}} = DX$. The results are shown in the Results section.

4.2. Image Super-Resolution

Image super-resolution refers to the problem of up-scaling an image with the loss of minimal detail. Regular interpolation techniques often cause the appearance of artifacts and are not effective at retaining detail and sharpness of the image. In this project, image super-resolution was explored as an extension of the K-SVD algorithm.

Consider a matrix X which represents the patches matrix extracted from a low resolution image (as described above), with a patch size of P_1. If we perform the same procedure of extracting patches from the corresponding high resolution image such that the same number of patches are extracted, the resulting patch size would be $P_2 = \frac{M \times N}{P_1} \times (F \times G)$, where the high resolution image has dimensions $M \times N$ and the low-resolution image has dimensions $F \times G$.

The proposed idea is as follows:
During the training process:

1. Learn a sparse representation X_s of the low resolution image X, with resulting dictionary D such that $X = DX_s$ and sparsity s
2. Obtain patch matrix Y from the high resolution image with patch size $P_2 = \frac{M \times N}{P_1} \times (F \times G)$
3. Learn a matrix Q such that $Y = QX_s$

To obtain the high-resolution image:

1. Obtain sparse representation X_s from the dictionary D using OMP.
2. Obtain the high-resolution patch matrix $Y_{test} = QX_s$
3. Obtain the image by processing the patch matrix into an image of resolution $M \times N$

In order to learn the matrix Q, we need to solve the optimization problem:

$$Q = \arg\min_{Q} \|Y - Qx_s\|_2^2$$

The solution can be obtained by noting that

$$\|Y - Qx_s\|_2^2 = \text{trace}((Y - Qx_s)(Y - Qx_s)^T)$$

Fig. 1. (top) Bilinear filtering. (bottom) Super-resolution using K-SVD.

and setting

$$\frac{\partial \text{trace}((Y - Qx_s)(Y - Qx_s)^T)}{\partial Q} = 0$$

We obtain that:

$$Q = YX_s^T(X_sX_s^T)^{-1}$$

The shown example compares a zoomed in view of the 512×512 image and the upscaled 1024×1024 image.

5. RESULTS

5.1. Image Denoising

The results from the image denoising examples are shown in Fig 2. and Fig 3.

5.2. Image Super-resolution

The results from the image resolution examples are shown in Fig 1

6. REFERENCES

1. M. Aharon, M. Elad, and A. Bruckstein - K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation
2. R. Rubinstein, M. Zibulevsky and M. Elad - Efficient Implementation of the K-SVD Algorithm and the Batch-OMP Method

Fig. 2. (top left) The original image. (top right) The noisy image. (bottom) The denoised image from K-SVD

Fig. 3. (top) The noisy image. (bottom) The denoised image.