**K-SVD FOR DUMMIES**

An Introduction to Sparse Representation and the K-SVD Algorithm

Ron Rubinstein

The CS Department
The Technion – Israel Institute of technology
Haifa 32000, Israel

University of Erlangen - Nürnberg
April 2008

---

**Noise Removal ?**

Our story begins with image denoising ...

- Practical application
- A convenient platform (being the simplest inverse problem) for testing basic ideas in image processing.

---

**Denoising By Energy Minimization**

Many of the proposed denoising algorithms are related to the minimization of an energy function of the form

\[ f(x) = \frac{1}{2} \| x - y \|_2^2 + \text{Pr}(x) \]

- \( x \): Unknown to be recovered
- \( y \): Given measurements
- Sanity (relation to measurements)
- Prior or regularization

- This is in-fact a Bayesian point of view, adopting the Maximum-Aposteriori Probability (MAP) estimation.
- Clearly, the wisdom in such an approach is within the choice of the prior – **modeling the images** of interest.

---

**The Evolution Of Pr(x)**

During the past several decades we have made all sort of guesses about the prior \( \text{Pr}(x) \) for images:

- \( \text{Pr}(x) = \lambda \| x \|_2^2 \) (Energy)
- \( \text{Pr}(x) = \lambda \| Lx \|_2^2 \) (Smoothness)
- \( \text{Pr}(x) = \lambda \| Lx \|_W^2 \) (Adapt + Smooth)
- \( \text{Pr}(x) = \lambda \| Bx \|_2 \) (Robust Statistics)
- \( \text{Pr}(x) = \lambda \| x \|_1^0 \) (Total Variation)
- \( \text{Pr}(x) = \lambda \| Wx \|_1^0 \) (Wavelet Sparsity)
- \( \text{Pr}(x) = \lambda \| Bx \|_2 \) (Bilateral Filter)
- \( \text{Pr}(x) = \lambda \| x \|_0 \) (Sparse & Redundant)

For \( x = D_\alpha \)
**Agenda**

1. **A Visit to Sparseland**
   - Introducing sparsity & overcompleteness

2. Transforms & Regularizations
   - How & why should this work?

3. What about the dictionary?
   - The quest for the origin of signals

4. Putting it all together
   - Image filling, denoising, compression, ...

---

**Generating Signals in Sparseland**

- Every column in $D$ (dictionary) is a prototype signal (atom).
- The vector $\alpha$ is generated randomly with few non-zeros in random locations and with random values.

---

**Sparseland Signals Are Special**

- **Simple**: Every signal is built as a linear combination of a few atoms from the dictionary $D$.
- **Effective**: Recent works adopt this model and successfully deploy it to applications.
- **Empirically established**: Neurological studies show similarity between this model and early vision processes. [Olshausen & Field (96)]

---

**Transforms in Sparseland?**

- Assume that $x$ is known to emerge from $M$.
- How about "Given $x$, find the $\alpha$ that generated it in $M$?"
An Introduction to Sparse Representation And the K-SVD Algorithm
Ron Rubinstein

Problem Statement
We need to solve an under-determined linear system of equations:
\[ \mathbf{D} \alpha = \mathbf{X} \]

- Among all (infinitely many) possible solutions we want the sparsest!!
- We will measure sparsity using the L_0 "norm":
  \[ \| \alpha \|_0 \]

Measure of Sparsity?
\[ \| \mathbf{x} \|_p = \sum_{j=1}^{k} |x_j|^p \]

As \( p \to 0 \) we get a count of the non-zeros in the vector
\[ \| \alpha \|_0 \]

Where We Are
A sparse & random vector
\[ \mathbf{\alpha} = \]
Multiply by \( \mathbf{D} \)
\[ \mathbf{x} = \mathbf{D} \mathbf{\alpha} \]

Min \[ \| \mathbf{\alpha} \|_0 \]
s.t. \[ \mathbf{x} = \mathbf{D} \mathbf{\alpha} \]
\[ \hat{\mathbf{\alpha}} \]

3 Major Questions
- Is \( \hat{\mathbf{\alpha}} = \mathbf{\alpha} \)?
- NP-hard: practical ways to get \( \hat{\mathbf{\alpha}} \)?
- How do we know \( \mathbf{D} \)?

Inverse Problems in Sparseland?
- Assume that \( \mathbf{x} \) is known to emerge from \( \mathcal{M} \).
- Suppose we observe \( \mathbf{y} = \mathbf{Hx} + \mathbf{v} \), a degraded and noisy version of \( \mathbf{x} \) with \( \| \mathbf{v} \|_2 \leq \varepsilon \). How do we recover \( \mathbf{x} \)?
- How about "find the \( \mathbf{\alpha} \) that generated \( \mathbf{y} " ?

\[ \mathcal{M} \]
\[ \mathbf{Hx} \]
\[ \mathbf{y} \]
\[ \mathbf{Q} \]
\[ \hat{\mathbf{\alpha}} \]

\[ \mathbf{D} \mathbf{\alpha} \]
\[ \mathbf{\alpha} \]
Noise
### Inverse Problem Statement

- **A sparse & random vector**
  \[ \alpha = \ldots \]

- **Multiply by \( D \)**
  \[ \text{"blur" by } H \]
  \[ y = Hx + v \]

- **Minimize**
  \[ \alpha \text{ s.t. } \|x\|_0 \leq \varepsilon \]

3 Major Questions (again!)

- Is \( \hat{\alpha} = \alpha \)?
- How can we compute \( \hat{\alpha} \)?
- What \( D \) should we use?

### Agenda

1. **A Visit to Sparseland**
   - Introducing sparsity & overcompleteness

2. **Transforms & Regularizations**
   - How & why should this work?

3. **What about the dictionary?**
   - The quest for the origin of signals

4. **Putting it all together**
   - Image filling, denoising, compression, ...

### The Sparse Coding Problem

Our dream for now: Find the sparsest solution to

\[ D\alpha = x \]

Put formally,

\[ \min_{\alpha} \|\alpha\|_0 \quad \text{s.t.} \quad x = D\alpha \]

Why should we necessarily get \( \hat{\alpha} = \alpha \)?

It might happen that eventually \( \|\hat{\alpha}\|_0 < \|\alpha\|_0 \).
**Matrix "Spark"**

**Definition:** Given a matrix $D$, $\sigma = \text{Spark}\{D\}$ is the smallest number of columns that are linearly dependent.  

Donoho & Elad (’02)

- By definition, if $Dv=0$ then $\|v\|_0 \geq \sigma$
- Say I have $\alpha_1$ and you have $\alpha_2$, and the two are different representations of the same $x$:
  
  $x = D\alpha_1 = D\alpha_2 \quad \Rightarrow \quad D(\alpha_1 - \alpha_2) = 0$

  $\Rightarrow \quad \|\alpha_1 - \alpha_2\|_0 \geq \sigma$

**Uniqueness Rule**

- Now, what if my $\alpha_1$ satisfies $\|\alpha_1\|_0 < \frac{\sigma}{2}$?
- The rule $\|\alpha_1 - \alpha_2\|_0 \geq \sigma$ implies that $\|\alpha_2\|_0 > \frac{\sigma}{2}$!

Donoho & Elad (’02)

Uniqueness: If we have a representation that satisfies $\frac{\sigma}{2} > \|\alpha\|_0$, then necessarily it is the sparsest.

So, if $M$ generates signals using "sparse enough" $\alpha$, the solution of will find them exactly.

$$P_0: \min_{\alpha} \|\alpha\|_0 \text{ s.t. } x = D\alpha$$

**Question 2 – Practical $P_0$ Solver?**

Are there reasonable ways to find $\hat{\alpha}$?

**Matching Pursuit (MP)** Mallat & Zhang (1993)

- The MP is a greedy algorithm that finds one atom at a time.
- Step 1: find the one atom that best matches the signal.
- Next steps: given the previously found atoms, find the next one to best fit...
- The Orthogonal MP (OMP) is an improved version that re-evaluates the coefficients after each round.
An Introduction to Sparse Representation
And the K-SVD Algorithm
Ron Rubinstein

Basis Pursuit (BP) Chen, Donoho, & Saunders (95)

Instead of solving
\[ \min_{\alpha} \| x \|_0 \text{ s.t. } x = D\alpha \]

→ Solve this:
\[ \min_{\alpha} \| \alpha \|_1 \text{ s.t. } x = D\alpha \]

- The newly defined problem is convex (linear programming).
- Very efficient solvers can be deployed:
  - Interior point methods [Chen, Donoho, & Saunders ('95)],
  - Iterated shrinkage [Figueiredo & Nowak ('03), Daubechies, Defrise, & Demole ('04), Elad ('05), Elad, Matalon, & Zibulevsky ('06)].

Question 3 – Approx. Quality?

\[ \alpha = \text{Multiply by } D \]
\[ x = D\alpha \]
\[ \hat{\alpha} \]

How effective are MP/BP?

BP and MP Performance

Donoho & Elad ('02)
Gribonval & Nielsen ('03)
Tropp ('03)
Temlyakov ('03)

Given a signal \( x \) with a representation \( x = D\alpha \), if \( \| \alpha \|_1 < \text{(some threshold)} \) then BP and MP are guaranteed to find it.

- MP and BP are different in general (hard to say which is better).
- The above results correspond to the worst-case.
- Average performance results available too, showing much better bounds [Donoho ('04), Candes et.al. ('04), Tanner et.al. ('05), Tropp et.al. ('06)].
- Similar results for general inverse problems [Donoho, Elad & Temlyakov ('04), Tropp ('04), Fuchs ('04), Gribonval et.al. ('05)].

Agenda

1. A Visit to Sparseland
   Introducing sparsity & overcompleteness
2. Transforms & Regularizations
   How & why should this work?
3. What about the dictionary?
   The quest for the origin of signals
4. Putting it all together
   Image filling, denoising, compression, ...
**Problem Setting**

Multiply by $D$

$X = D\alpha$

$\|\alpha\|_0 \leq L$

Given these P examples and a fixed size (N x K) dictionary, how would we find $D$?

**The Objective Function**

$$\min_{D,A} \|DA - X\|^2_F \quad \text{s.t.} \quad \forall j, \|a_j\|_0 \leq L$$

The examples are linear combinations of atoms from $D$

Each example has a sparse representation with no more than L atoms

(N, K, L are assumed known, $D$ has normalized columns)

**K–SVD – An Overview**

- Initialize $D$
- Sparse Coding: Use MP or BP
- Dictionary Update: Column-by-Column by SVD computation

Aharon, Elad & Bruckstein ('04)

**K–SVD: Sparse Coding Stage**

For the jth example we solve

$$\min_{\alpha} \|D\alpha - X_j\|^2_2 \quad \text{s.t.} \quad \|\alpha\|_0 \leq L$$

Ordinary Sparse Coding!
K–SVD: Dictionary Update Stage

\[ \text{Min}_D \| \mathbf{DA} - \mathbf{X} \|_F^2 \quad \text{s.t.} \quad \forall j, \| \mathbf{a}_j \|_0 \leq L \]

For the \( k \)th atom we solve

\[ \text{Min}_{\mathbf{a}_k} \| \mathbf{d}_k \mathbf{a}_k^T - \mathbf{E}_k \|_F^2 \]

\[ \mathbf{E}_k = \sum_{p,k} \mathbf{d}_p \mathbf{a}_k^T \mathbf{x} \quad \text{(the residual)} \]

K–SVD Dictionary Update Stage

We can do better than this

\[ \text{Min}_{\mathbf{a}_k} \| \mathbf{d}_k \mathbf{a}_k^T - \mathbf{E}_k \|_F^2 \]

But wait! What about sparsity?

K–SVD Dictionary Update Stage

We want to solve:

\[ \text{Min}_{\mathbf{a}_k} \| \mathbf{d}_k - \mathbf{\hat{a}}_k \|_F^2 \]

Only some of the examples use column \( \mathbf{d}_k \)!

When updating \( \mathbf{\hat{a}}_k \), only recompute the coefficients corresponding to those examples

Solve with SVD!

The K–SVD Algorithm – Summary

Initialize \( \mathbf{D} \)

Sparse Coding
Use MP or BP

Dictionary Update
Column-by-Column by SVD computation
Agenda

1. A Visit to Sparseland
   Introducing sparsity & overcompleteness
2. Transforms & Regularizations
   How & why should this work?
3. What about the dictionary?
   The quest for the origin of signals
4. Putting it all together
   Image filling, denoising, compression, ...

Image Inpainting: Theory

- Assumption: the signal $x$ was created by $x = D\alpha_0$, with a very sparse $\alpha_0$.
- Missing values in $x$ imply missing rows in this linear system.
- By removing these rows, we get $D\alpha_0 = \tilde{x}$.
- Now solve $\min \|\alpha\|_0 \text{ s.t. } \tilde{x} = D\alpha$.
- If $\alpha_0$ was sparse enough, it will be the solution of the above problem! Thus, computing $D\alpha_0$ recovers $x$ perfectly.

Inpainting: The Practice

- We define a diagonal mask operator $W$ representing the lost samples, so that $y = W\hat{x} + \nu$ where $w_{i,i} \in \{0,1\}$.
- Given $y$, we try to recover the representation of $\hat{x}$ by solving $\hat{\alpha} = \text{ArgMin}_{\alpha} \|\alpha\|_0 \text{ s.t. } y - WD\alpha \leq \epsilon$.
- We use a dictionary that is the sum of two dictionaries, to get an effective representation of both texture and cartoon contents. This also leads to image separation [Elad, Starck, & Donoho (05)].

Inpainting Results

- Source: Curvelet (cartoon) + Global DCT (texture)
- Dictionary: Curvelet (cartoon) + Global DCT (texture)
An Introduction to Sparse Representation And the K-SVD Algorithm

Ron Rubinstein

**Inpainting Results**

<table>
<thead>
<tr>
<th>Source</th>
<th>Dictionary: Curvelet (cartoon) + Overlapped DCT (texture)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outcome</td>
<td></td>
</tr>
</tbody>
</table>

**Denoising: Theory and Practice**

- Given a noisy image \( y \), we can clean it by solving:
  \[
  \hat{x} = \arg\min_\alpha \| \alpha \| \text{ s.t. } \| y - D\alpha \|_2 \leq \epsilon \quad \Rightarrow \quad \hat{x} = D\hat{\alpha}
  \]

- Can we use the K-SVD dictionary?

- With K-SVD, we cannot train a dictionary for an entire image. How do we go from local treatment of patches to a global prior?

- Solution: force shift-invariant sparsity – for each NxN patch of the image, including overlaps.

**From Local to Global Treatment**

\[
\hat{x} = \arg\min_\alpha \frac{1}{2} \| x - y \|_2^2 + \mu \sum_{i,j} \| R_{ij} (x - D\alpha_{ij}) \|_2 \text{ s.t. } \| \alpha_{ij} \|_0 \leq L
\]

Extracts the \((i,j)\)th patch

For patches, our MAP penalty becomes

Our prior
What Data to Train On?

Option 1:
- Use a database of images: works quite well (~0.5-1dB below the state-of-the-art)

Option 2:
- Use the corrupted image itself!
- Simply sweep through all $N \times N$ patches (with overlaps) and use them to train
- Image of size 1000x1000 pixels $\Rightarrow 10^6$ examples to use – more than enough.
- This works much better!

Image Denoising: The Algorithm

\[
\hat{x} = \text{ArgMin}_x \left\{ \frac{1}{2} \| x - y \|_2^2 + \mu \sum_i \| R_i \cdot x \cdot D \alpha_i \|_2^2 \right\} \quad \text{s.t.} \quad \| \alpha_i \|_0 \leq L
\]

Compute $\alpha_{ij}$ per patch

\[
\alpha_{ij} = \text{Min}_{\alpha_{ij}} \left\{ \| R_i \cdot x \cdot D \alpha_{ij} \|_2^2 \right\}
\quad \text{s.t.} \quad \| \alpha_{ij} \|_0 \leq L
\]

using matching pursuit

Compute $D$ to minimize

\[
\text{Min}_D \sum_i \| R_i \cdot x \cdot D \alpha_{ij} \|_2^2
\]

using SVD, updating one column at a time

K-SVD

Denoising Results

Source

Result 30.829dB

Noisy image

PSNR = 22.1dB

Obtained dictionary after 10 iterations

Denoising Results: 3D

Source: Vis. Male Head (Slice #137)

2d-KSVD: PSNR=27.3dB

PSNR=12dB

3d-KSVD: PSNR=32.4dB
Image Compression

- Problem: compressing photo-ID images.
- General purpose methods (JPEG, JPEG2000) do not take into account the specific family.
- By adapting to the image-content, better results can be obtained.

Compression: The Algorithm

1. Detect main features and align the images to a common reference (20 parameters)
2. Divide each image to disjoint 15x15 patches, and for each compute a unique dictionary
3. Detect features and align
4. Divide to disjoint patches, and sparse-code each patch
5. Quantize and entropy-code

Training set (2500 images)

Compression Results

<table>
<thead>
<tr>
<th>Original</th>
<th>JPEG</th>
<th>JPEG 2000</th>
<th>PCA</th>
<th>K-SVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results for 820 bytes per image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottom: RMSE values</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original</th>
<th>JPEG</th>
<th>JPEG 2000</th>
<th>PCA</th>
<th>K-SVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results for 550 bytes per image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottom: RMSE values</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Today We Have Discussed

1. A Visit to Sparseland
   Introducing sparsity & overcompleteness
2. Transforms & Regularizations
   How & why should this work?
3. What about the dictionary?
   The quest for the origin of signals
4. Putting it all together
   Image filling, denoising, compression, ...

Summary

Sparsity and over-completeness are important ideas for designing better tools in signal and image processing.

Approximation algorithms can be used, are theoretically established and work well in practice.

Coping with an NP-hard problem

We have seen inpainting, denoising and compression algorithms.

What dictionary to use?

How is all this used?

Several dictionaries already exist. We have shown how to practically train $D$ using the K-SVD.

(a) Generalizations: multiscale, non-negative,...
(b) Speed-ups and improved algorithms
(c) Deploy to other applications

Why Over-Completeness?

Desired Decomposition
Inpainting Results

70% Missing Samples

DCT (RMSE=0.04)

Haar (RMSE=0.045)

K-SVD (RMSE=0.03)

90% Missing Samples

DCT (RMSE=0.085)

Haar (RMSE=0.07)

K-SVD (RMSE=0.06)