Compressive Sensing

CS 754
Ajit Rajwade

Outline of the Lectures

- Theorem 3: Sketch of Proof and comments on the theorem
- Design of compressed sensing matrices
- Compressive classification

Theorem 3

- Suppose the matrix $\mathbf{A}=\mathbf{\Phi\Psi}$ of size *m* by *n* (where sensing matrix $\mathbf{\Phi}$ has size *m* by *n*, and basis matrix $\mathbf{\Psi}$ has size *n* by *n*) has RIP property of order 2*S* where $\delta_{2S} < 0.41$. Let the solution of the following be denoted as $\boldsymbol{\theta}^*$, (for signal $\mathbf{f} = \mathbf{\Psi}\boldsymbol{\theta}$, measurement vector $\mathbf{y} = \mathbf{\Phi\Psi}\boldsymbol{\theta}$):

$$\min \|\boldsymbol{\theta}\|_1 \text{ such that } \|\mathbf{y} - \mathbf{\Phi\Psi}\boldsymbol{\theta}\|_2 \leq \varepsilon$$

Then we have:

$$\|\boldsymbol{\theta}^* - \boldsymbol{\theta}\|_2 \leq \frac{C_0}{\sqrt{S}}\|\boldsymbol{\theta} - \boldsymbol{\theta}_\mathbf{S}\|_1 + C_1\varepsilon$$

$\boldsymbol{\theta}_S$ is created by retaining the S largest magnitude elements of $\boldsymbol{\theta}$, and setting the rest to 0.

Comments on Theorem 3

- Theorem 3 is a direct extension of Theorem 2 for the case of noisy measurements.
- It states that the solution of the convex program (see previous slides) gives a reconstruction error which is the sum of two terms: (1) the error of an oracle solution where the oracle told us the S largest coefficients of the signal $\mathbf{f}$ at the correct indices, and (2) a term proportional to the noise variance.
- The constants C_0 and C_1 are very small (less than or equal to 5.5 and 6 respectively for $\delta_{2S} = 0.25$), they are increasing functions of just δ_{2S}.

Sketch of the proof

- The proof can be found in a paper by Candes "The restricted isometry property and its implications for compressed sensing", published in 2008.

- The proof as such is just between 1 and 2 pages long.

Sketch of the proof

- The proof uses various properties of vectors in Euclidean space.

- The Cauchy Schwartz inequality: $|\mathbf{v} \cdot \mathbf{w}| \leq \|\mathbf{v}\|_2 \|\mathbf{w}\|_2$

- The triangle inequality: $\|\mathbf{v} + \mathbf{w}\|_2 \leq \|\mathbf{v}\|_2 + \|\mathbf{w}\|_2$

- Reverse triangle inequality: $\left| \|\mathbf{v}\|_2 - \|\mathbf{w}\|_2 \right| \leq \|\mathbf{v} - \mathbf{w}\|_2$

Sketch of the proof

- Relationship between various norms:

$$\|\mathbf{v}\|_2 \le \|\mathbf{v}\|_1 = |\mathbf{v}| \cdot \mathbf{1} \le \sqrt{n}\|\mathbf{v}\|_2$$

$$\|\mathbf{v}\|_1 \le \sqrt{k}\|\mathbf{v}\|_2 \text{ if } \mathbf{v} \text{ is a } k\text{-sparse vector}$$

- Refer to Theorem 3. For the sake of simplicity alone, we shall assume Ψ to be the identity matrix.

- Hence $\mathbf{x} = \boldsymbol{\theta}$. Even if Ψ were not identity, the proof as such does not change.

Sketch of the proof: Step 1

This result is called the **Tube constraint**.

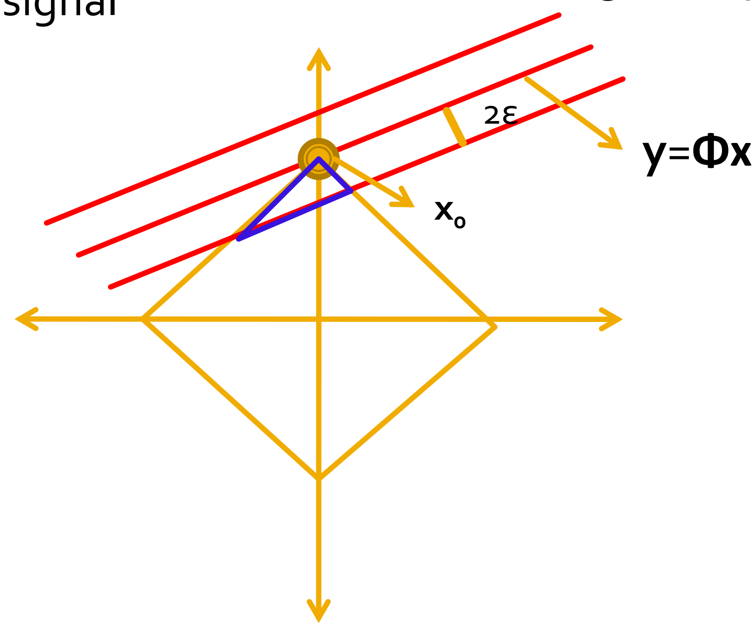
- We have:

$$\left\|\boldsymbol{\Phi}(\mathbf{x}^* - \mathbf{x}_0)\right\|_2 \le \boxed{\left\|\boldsymbol{\Phi}\mathbf{x}^* - \mathbf{y}\right\|_2 + \left\|\boldsymbol{\Phi}\mathbf{x}_0 - \mathbf{y}\right\|_2} \le 2\varepsilon$$

Triangle inequality

Given constraint + feasibility of solution **x***

In the following, $\mathbf{x}_o$ = true signal



Sketch of the proof: Preparing for steps 2 and 3

- Define vector $\mathbf{h} = \mathbf{x}^* - \mathbf{x}_0$.

- Decompose **h** into vectors $\mathbf{h}_{T0}$, $\mathbf{h}_{T1}$, $\mathbf{h}_{T2}$,… which are all at the most *s*-sparse.
- T_o is the set of indices corresponding to the *s* largest absolute value elements of $\mathbf{x}_o$,
- T_1 is the set of indices corresponding to the *s* largest absolute value elements of $\mathbf{h}_{(To\text{-}c)} = \mathbf{h}\text{-}\mathbf{h}_{To}$,
- T_2 is the set of indices corresponding to the *s* largest absolute value elements of $\mathbf{h}_{(To \cup T1)\text{-}c} = \mathbf{h}\text{-}\mathbf{h}_{To}\text{-}\mathbf{h}_{T1}$,
- and so on.

Sketch of the proof: Step 2

- We will assume $\mathbf{x_o}$ is s-sparse (later we will relax this to assume it is compressible).
- We now establish the so-called **cone constraint**.

$$\left\|x^*\right\|_1 = \left\|x_0 + h\right\|_1 \le \left\|x_0\right\|_1$$

The vector **h** has its origin at $\mathbf{x_o}$ and it lies in the intersection of the L1 ball and the tube.

$$\therefore \sum_{i \in T_0} |x_{0i} + h_i| + \sum_{i \in T_{0-c}} |x_{0i} + h_i| \le \left\|x_0\right\|_1$$

0-valued

$$\therefore \left\|x_0\right\|_1 - \left\|h_{T0}\right\|_1 + \left\|h_{T0-c}\right\|_1 \le \left\|x_0\right\|_1$$

$$\therefore \left\|h_{T0-c}\right\|_1 \le \left\|h_{T0}\right\|_1$$

The vector **h** must also necessarily obey this constraint – the cone constraint.

Sketch of the proof: Preparing for steps 3 and onwards

- We will now prove that such a vector **h** is orthogonal to the null-space of **Φ**.
- In fact, we will prove that $\|\mathbf{\Phi h}\|_2 \approx \|\mathbf{h}\|_2$.
- In other words, we will prove that the magnitude of **h** is not much greater than 2ε, which means that the solution **x*** of the optimization problem is "close enough" to $\mathbf{x_0}$.

Sketch of the proof: Step 3

- In step 3, we use a bunch of algebraic manipulations to prove that the magnitude of **h** outside of $T_0 \cup T_1$ is upper bounded by the magnitude of **h** on $T_0 \cup T_1$.

- In other words, we prove that:

$$\|\mathbf{h}_{(T0 \cup T1)c}\|_2 \le \|\mathbf{h}_{T0}\|_2 \le \|\mathbf{h}_{T0 \cup T1}\|_2$$

- The algebra involves various inequalities mentioned earlier.

Sketch of the proof: Steps 4 and 5

- We now prove that the magnitude of **h** on $T_0 \cup T_1$ is upper bounded by a reasonable quantity.
- For this, we show using the RIP of $\mathbf{\Phi}$ of order 2*s* and a series of manipulations that:

$$(1-\delta_{2s})\|h_{T0\cup T1}\|_2^2 \le \|\Phi h_{T0\cup T1}\|_2^2 \le \|h_{T0\cup T1}\|_2 (2\varepsilon\sqrt{1+\delta_{2s}} + \sqrt{2}\delta_{2s}\|h_{T0}\|_2)$$

- This implies that

$$\|h_{T0\cup T1}\|_2 \le \left(\frac{2\varepsilon\sqrt{1+\delta_{2s}}}{1-\delta_{2s}} + \frac{\sqrt{2}\delta_{2s}}{1-\delta_{2s}}\|h_{T0\cup T1}\|_2 \right) \rightarrow \|h_{T0\cup T1}\|_2 \le \frac{2\sqrt{1+\delta_{2s}}\varepsilon}{1-\delta_{2s}(\sqrt{2}+1)} \rightarrow \|h\|_2 \le \frac{4\varepsilon\sqrt{1+\delta_{2s}}}{1-\delta_{2s}(\sqrt{2}+1)}$$

The proof for compressible signals

- The steps change a bit. The cone constraint changes to:

$$\left\|x^*\right\|_1 = \left\|x_0 + h\right\|_1 \le \left\|x_0\right\|_1$$

$$\therefore \sum_{i \in T_0} | x_{0i} + h_i | + \sum_{i \in T_{0-c}} | x_{0i} + h_i | \le \left\|x_0\right\|_1$$

$$\therefore \left\|x_{0,T0}\right\|_1 - \left\|h_{T0}\right\|_1 + \left\|h_{T0-c}\right\|_1 - \left\|x_{0,T0-c}\right\|_1 \le \left\|x_0\right\|_1$$

$$\therefore \left\|h_{T0-c}\right\|_1 \le \left\|h_{T0}\right\|_1 + 2\left\|x_{0,T0-c}\right\|_1$$

The proof for compressible signals

- All the other steps remain as is, except the last one which produces the following bound:

$$\|h\|_2 \le \frac{2\sqrt{1+\delta_{2s}}}{1-\delta_{2s}(1+\sqrt{2})}\varepsilon + \frac{1+\delta_{2s}(\sqrt{2}-1)}{1-\delta_{2s}(\sqrt{2}+1)} \frac{\|x_0 - x_{0,T0}\|_1}{\sqrt{s}}$$

Sketch of the proof: Step 3

- Step 3 of the proof uses the following corollary of the RIP for two *s*-sparse unit vectors with disjoint support: $|\mathbf{\Phi x}_1 \bullet \mathbf{\Phi x}_2| \le \delta_{2s}$

- Proof of corollary:

$$(1-\delta_{2S})\|\mathbf{x}_1 - \mathbf{x}_2\|^2 \le \|\Phi(\mathbf{x}_1 - \mathbf{x}_2)\|^2 \le (1+\delta_{2s})\|\mathbf{x}_1 - \mathbf{x}_2\|^2, \text{ by RIP}$$

$$|\mathbf{\Phi x}_1 \bullet \mathbf{\Phi x}_2| = \frac{1}{4}\left|\|\mathbf{\Phi x}_1 + \mathbf{\Phi x}_2\|^2 - \|\mathbf{\Phi x}_1 - \mathbf{\Phi x}_2\|^2\right|$$

$$\le \frac{1}{4}\left((1+\delta_{2s})\|\mathbf{x}_1 + \mathbf{x}_2\|^2 - (1-\delta_{2s})\|\mathbf{x}_1 - \mathbf{x}_2\|^2\right)$$

$$\le \frac{1}{4}\left((1+\delta_{2s})(\|\mathbf{x}_1\|^2 + \|\mathbf{x}_2\|^2 + 2\mathbf{x}_1 \bullet \mathbf{x}_2) - (1-\delta_{2s})(\|\mathbf{x}_1\|^2 + \|\mathbf{x}_2\|^2 - 2\mathbf{x}_1 \bullet \mathbf{x}_2)\right)$$

$$\le \delta_{2s} \because \|\mathbf{x}_1\|^2 = \|\mathbf{x}_2\|^2 = 1, x_1 \bullet \mathbf{x}_2 = 0$$

Sketch of the proof: Step 3

- This step also uses the following corollary of the RIP for two *s*-sparse unit vectors with disjoint support: $|\mathbf{\Phi x_1} \bullet \mathbf{\Phi x_2}| \le \delta_{2s}$

- What if the original vectors $\mathbf{x_1}$ and $\mathbf{x_2}$ were not unit-vectors, but both were *s*-sparse?

$$\frac{|\mathbf{\Phi x_1} \bullet \mathbf{\Phi x_2}|}{\|\mathbf{x_1}\|_2 \|\mathbf{x_2}\|_2} \le \delta_{2s} \rightarrow |\mathbf{\Phi x_1} \bullet \mathbf{\Phi x_2}| \le \delta_{2s} \|\mathbf{x_1}\|_2 \|\mathbf{x_2}\|_2$$

Comments on the theorem

- The bound is

$$\|\mathbf{h}\|_2 \le \frac{4\sqrt{1+\delta_{2s}}}{1-(1+\sqrt{2})\delta_{2s}}\varepsilon + \frac{1}{\sqrt{s}}\frac{1+\delta_{2s}(\sqrt{2}-1)}{1-(1+\sqrt{2})\delta_{2s}}\|\mathbf{x_0}-\mathbf{x_{0,T0}}\|_1$$

- Note the requirement that δ_{2s} should be less than $2^{0.5}-1$.
- You can prove that the two constant factors – one before ε and the other before $|\mathbf{x_0}-\mathbf{x_{0,T0}}|_1$, are both increasing functions of δ_{2s} in the domain [0,1].
- So sensing matrices with smaller values of δ_{2s} are always nicer!

Theorem 6 (Cai et al): Improvement over Theorem 3

- Suppose the matrix **A=$\boldsymbol{\Phi\Psi}$** of size *m* by *n* (where sensing matrix $\boldsymbol{\Phi}$ has size *m* by *n*, and basis matrix $\boldsymbol{\Psi}$ has size *n* by *n*) has RIP property of order *S* where $\delta_S < 0.307$. Let the solution of the following be denoted as $\boldsymbol{\theta}^*$, (for signal **f** = $\boldsymbol{\Psi\theta}$, measurement vector **y**=$\boldsymbol{\Phi\Psi\theta}$):

$$\min \|\boldsymbol{\theta}\|_1 \text{ such that } \|\mathbf{y} - \boldsymbol{\Phi\Psi\theta}\|_2^2 \leq \varepsilon$$

  Then we have:

$$\|\boldsymbol{\theta}^* - \boldsymbol{\theta}\|_2 \leq \frac{1}{(0.307-\delta_k)\sqrt{S}}\|\boldsymbol{\theta} - \boldsymbol{\theta}_S\|_1 + \frac{1}{0.307-\delta_k}\varepsilon$$

$\boldsymbol{\theta}_S$ is created by retaining the S largest magnitude elements of $\boldsymbol{\theta}$, and setting the rest to 0.

Ψ Need not be orthonormal

- Theorems 3,5,6 refer to orthonormal bases for the signal to have sparse or compressible representations.

- However that is not a necessary condition.

- There exist the so-called **"over-complete bases"** in which the number of columns exceeds the number of rows (n x K, $K > n$).

- Such matrices afford even sparser signal representations.

Ψ Need not be orthonormal

- Why? We explain with an example.
- A cosine wave (with grid-aligned frequency) will have a sparse representation in the DCT basis $\mathbf{V}_1$.
- An impulse signal has sparse representation in the identity basis $\mathbf{V}_2$.
- Now consider a signal which is the superposition of a small number of cosines and impulses.
- The combined signal has sparse representation in neither the DCT basis nor the identity basis.
- But the combined signal will have a sparse representation in the combined dictionary $[\mathbf{V}_1\ \mathbf{V}_2]$.

Designing Sensing Matrices in CS

Designing Compressed Sensing Matrices

- We know that certain classes of random matrices satisfy the RIP with very high probability.
- However, we also know that small RICs are desirable.
- This gives rise to the question: Can we design matrices with **smaller RIC** than a randomly generated matrix?

Designing Compressed Sensing Matrices

- Unfortunately, there is no known efficient algorithm for even computing the RIC given a fixed matrix!

- But we know that the mutual coherence of $\mathbf{\Phi\Psi}$ is an upper bound to the RIC: $\delta_s \leq \mu(s-1)$

- So we can design a CS matrix by starting with a random one, and then performing **a gradient descent on the mutual coherence** to reach a matrix with a smaller mutual coherence!

Designing Compressed Sensing Matrices

- The procedure is summarized below:

Randomly pick a m by n matrix $\mathbf{\Phi}$.

Repeat until convergence

{

$$\mathbf{\Phi} \leftarrow \mathbf{\Phi} - \alpha \frac{\partial}{\partial \mathbf{\Phi}} (\mu(\mathbf{\Phi\Psi}))$$

}

Pick the step-size adaptively so that you actually descend on the mutual coherence.

$$\mu(\mathbf{\Phi\Psi}) = \max_{i \neq j} \left| \frac{(\mathbf{\Phi\Psi})_i}{\left\|(\mathbf{\Phi\Psi})_i\right\|_2} \bullet \frac{(\mathbf{\Phi\Psi})_j}{\left\|(\mathbf{\Phi\Psi})_j\right\|_2} \right|$$

Designing Compressed Sensing Matrices

- The aforementioned is one example of a procedure to “design” a CS matrix – as opposed to picking one randomly.
- Note that mutual coherence has one more advantage over RIC – the former is **not tied to any particular sparsity level**!
- But one must bear in mind that the mutual coherence is an upper bound to the RIC!

Designing CS matrices: Method 1

- The main problem is how to find a derivative of the “max” function which is non-differentiable!

- Use the softmax function which is differentiable:

$$\lim_{\beta \to \infty} \frac{1}{\beta} \log\left(\sum_{i=1}^{n} \exp(\beta x_i) \right) = \max\{x_i\}_{i=1}^{n}$$

Designing CS matrices: Method 1

- This method can now be used to design CS matrices.
- As the mutual coherence function is expected to be very non-convex, one must run a multi-start strategy in practice.
- For each "start", you begin with a random $\mathbf{\Phi}$, do a gradient descent on μ till convergence.
- Repeat this procedure many times, each time beginning from a different randomly chosen initial condition.
- Choose the value of μ that is the smallest – among all these starts.

Designing CS matrices: Method 1

- In many cases, one needs additional constraints on the matrix $\mathbf{\Phi}$.

- For example, in a Hitomi video camera architecture, $\mathbf{\Phi}$ is a concatenation of non-negative and diagonal matrices.

- The non-negativity can be imposed by means of **projected** gradient descent.

- See the next slide for the modified algorithm to maintain non-negativity.

Designing CS matrices: Method 1

Randomly pick a m by n matrix $\mathbf{\Phi}$.

Repeat until convergence

{

$$\mathbf{\Phi} \leftarrow \mathbf{\Phi} - \alpha \frac{\partial}{\partial \mathbf{\Phi}}(\mu(\mathbf{\Phi\Psi}))$$

Set negative entries in $\mathbf{\Phi}$ to 0.

}

Pick the step-size adaptively so that you actually descend on the mutual coherence **after** setting the negative entries to 0.

https://arxiv.org/abs/1609.02135

Figure 9: Demosaicing close-ups, examples {1, 2, 3}. Clockwise: inputs, reconstructions with {random, {circularly, non-circularly} designed} matrices

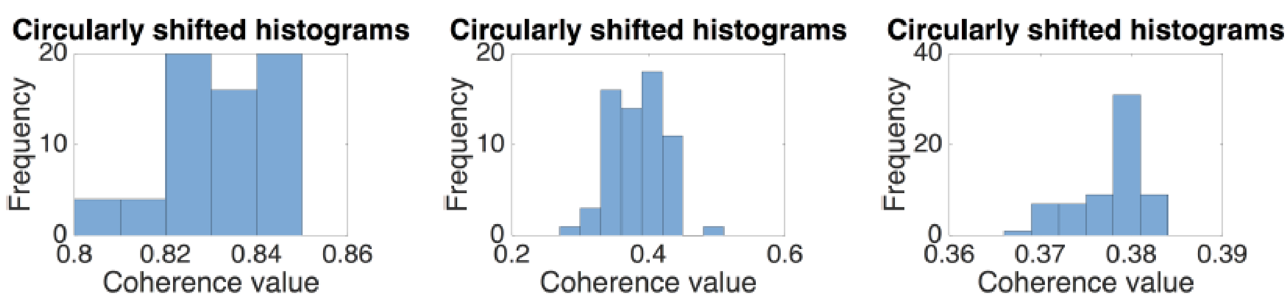


Figure 15: Left to right: Circularly-shifted coherence histograms for {random, non-circularly optimized, circularly optimized} matrices

Designing CS Matrices: Method 2

- This method does not directly target μ but instead considers the Gram matrix $\mathbf{D}^T\mathbf{D}$ where
  $\mathbf{D} = \mathbf{\Phi}\mathbf{\Psi}$ with all columns unit-normalized.

- The aim is to design $\mathbf{\Phi}$ in such a way that the Gram matrix resembles the identity matrix as much as possible, in other words we want:
  $$\mathbf{\Psi}^T\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{\Psi} \approx \mathbf{I}$$

Design CS matrices:

$$E = \| \tilde{\mathbf{D}}^T \tilde{\mathbf{D}} - \mathbf{I} \|_F^2 = \mathbf{Tr}\{(\tilde{\mathbf{D}}^T \tilde{\mathbf{D}} - \mathbf{I})(\tilde{\mathbf{D}}^T \tilde{\mathbf{D}} - \mathbf{I})^T\}$$

$$\nabla E \equiv \frac{\partial E}{\partial \tilde{d}_{ij}} = 4\tilde{\mathbf{D}}(\tilde{\mathbf{D}}^T \tilde{\mathbf{D}} - \mathbf{I})$$

$$\tilde{\mathbf{D}}_{(i+1)} = \tilde{\mathbf{D}}_{(i)} - \hat{\eta}\tilde{\mathbf{D}}_{(i)}(\tilde{\mathbf{D}}_{(i)}^T \tilde{\mathbf{D}}_{(i)} - \mathbf{I})$$

$$\hat{\Phi} = \arg\min_{\tilde{\Phi}} \| \Psi^T \Phi^T \Phi \Psi - \mathbf{I} \|_F^2$$

$$\frac{\partial E}{\partial \phi_{ij}} = 4\Phi\Psi(\Psi^T \Phi^T \Phi\Psi - \mathbf{I})\Psi^T$$

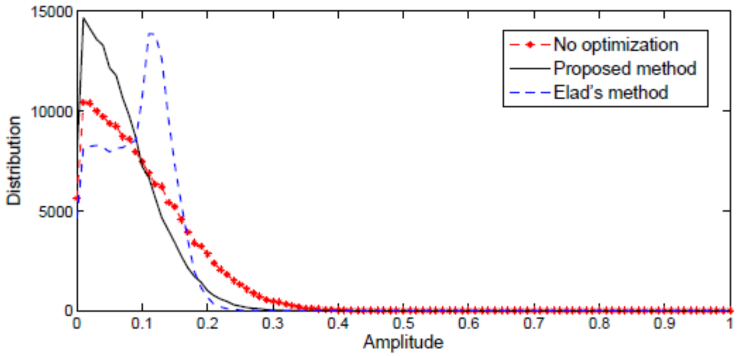
$$\Phi_{(i+1)} = \Phi_{(i)} - \eta\Phi_{(i)}\Psi(\Psi^T \Phi_{(i)}^T \Phi_{(i)}\Psi - \mathbf{I})\Psi^T.$$

Algorithm 1: Gradient-descent optimization.

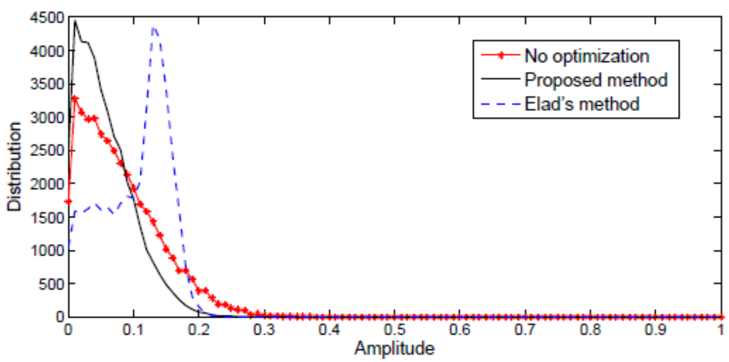
```
Input: Sparse representation basis Ψ_nn (if necessary),
       Stepsize η, Maximum number of iterations K.
Output: Measurement matrix Φ_pn.
begin
    Initialize D to a random matrix.
    for k=1 to K do
        for j=1 to n do
            d^j ⟵ d^j / ||d^j||_2
        end
        D ⟵ D − ηD(D^T D − I)
    end
    if Ψ_nn has been given as input then
        Φ̂ ⟵ DΨ^{-1}
    else
        Φ̂ ⟵ D
    end
end
```

Abdoghasemi et al, On optimization of the measurement matrix for compressive sensing, EUSIPCO 2010

Design CS matrices:

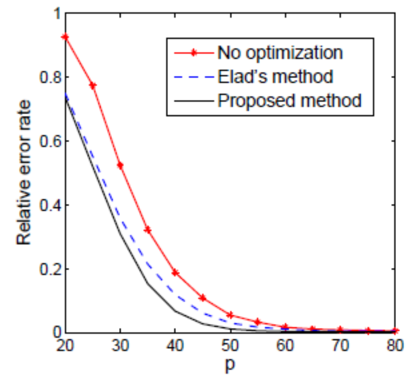


(a) Φ: Random, and Ψ: Random.

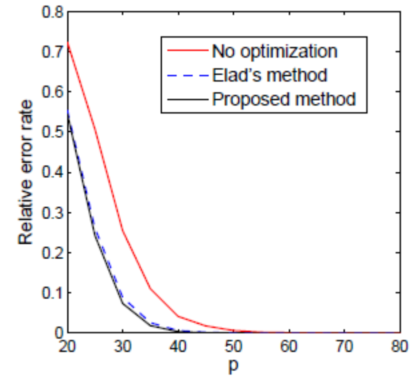


(b) Φ: Random, and Ψ: DCT.

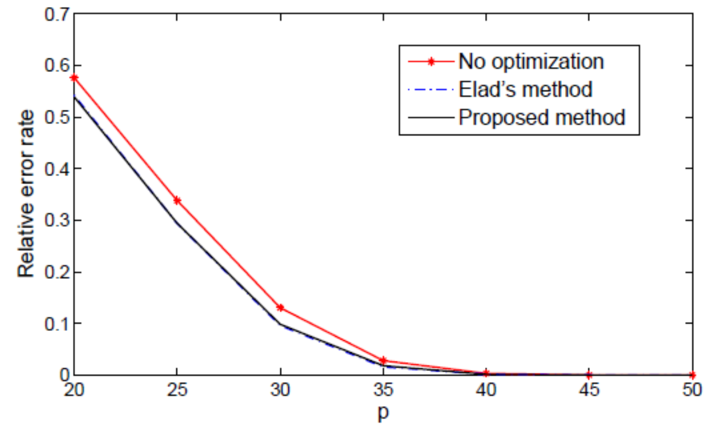
Figure 1: Distribution of off-diagonal elements of **G**.



(a)



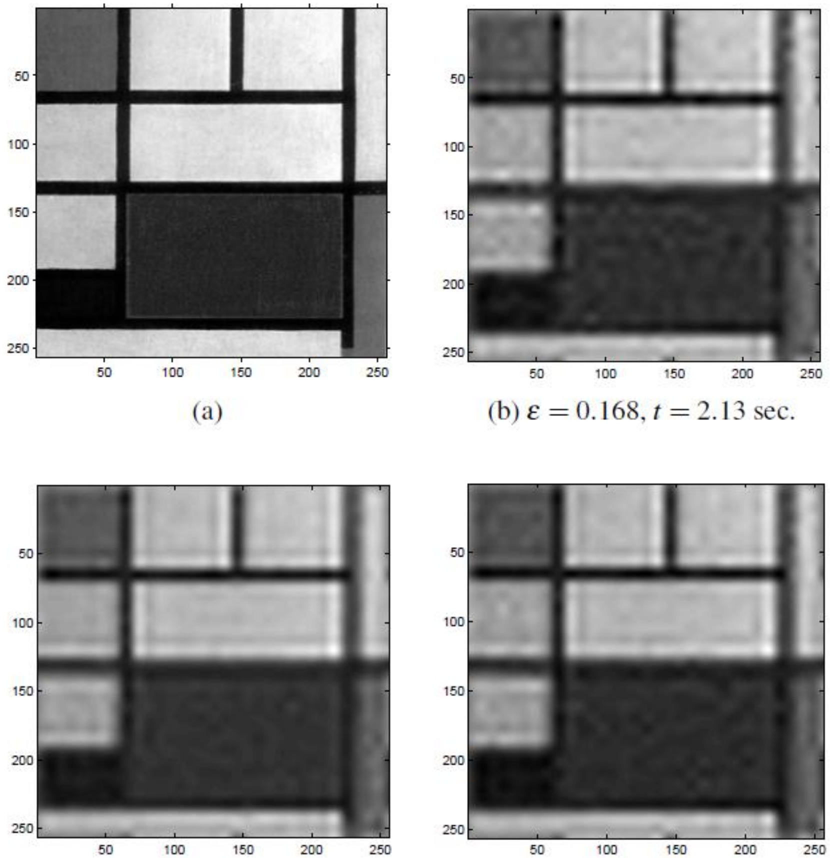
(b)



(c)

Figure 2: Relative error rate vs. the number of measurements p, using (a) IHT, (b) OMP, and (c) BP for reconstruction.

Design CS matrices:



(a)

(b) $\varepsilon = 0.168, t = 2.13$ sec.

(c) $\varepsilon = 0.157, t = 19.37$ sec.

(d) $\varepsilon = 0.163, t = 444.3$ sec.

Figure 4: Reconstruction of Mondrian image using HALS_CS method. (a) Original image. Reconstruction with (b) no optimization, (c) proposed optimization, and (d) Elad's optimization of the measurement matrix.

Abdoghasemi et al, On optimization of the measurement matrix for compressive sensing, EUSIPCO 2010

Results

- Leading to smaller values of column-column dot products
- Better reconstruction errors
- For more details refer to Abdoghasemi et al, On optimization of the measurement matrix for compressive sensing, EUSIPCO 2010.

Compressive Classification

Problem definition

- We have extensively examined the issue of reconstruction of signals or images from compressive reconstructions – algorithms, systems as well as theory (theorems).
- Now imagine you had compressive measurements for each of a set of *K* classes of images.
- The task is to classify the measurements into one of the *K* categories ***without*** intermediate reconstruction.
- This is called as the problem of **compressive classification**.

Maximum Likelihood Classifier

- Consider a vector **y** which is a noisy measurement of vector **x** in the following way:
  $\mathbf{y} = \mathbf{x} + \mathbf{n}$, $\mathbf{n} \sim N(0,\sigma^2)$.
- Let us suppose that **x** belongs to one of *P* classes, each class containing a single representative vector $\mathbf{s}_i$, $1 <= i <= P$.
- The likelihood that **y** is a noisy sample from the i^{th} class is given as:

$$p(\mathbf{y} \mid \mathbf{s}_\mathbf{i}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{s}_\mathbf{i}\|^2}{2\sigma^2} \right)$$

Maximum Likelihood Classifier

- The maximum likelihood classifiers assigns to **y** the class *j* such that

$$j = \arg\max_{k \in \{1,2,\ldots,P\}} p(\mathbf{y} \mid \mathbf{s_k})$$

$$p(\mathbf{y} \mid \mathbf{s_i}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\|\mathbf{y} - \mathbf{s_i}\|^2}{2\sigma^2} \right)$$

- By using the –log, we see that this reduces to a nearest neighbour classifier with Euclidean distance.

Generalized Maximum Likelihood Classifier (GMLC)

- Consider the earlier problem was an image classification problem where we had *P* image templates in a database.
- We observed Gaussian noisy versions of one of these templates and wanted to determine which one it was.
- Now in addition, let us suppose that the noisy version of the image were acquired in some different "pose", i.e. with some translations and/or rotation.
- Let the pose parameters be denoted by a vector $\boldsymbol{\theta}$ which belong to a set of values Θ.
- In such a case, this becomes a joint problem – of classification as well as pose estimation.

Generalized Maximum Likelihood Classifier (GMLC)

- The problem is solved as follows:

$$j = \arg\max_{k \in \{1,2,\ldots,P\}} p(\mathbf{y} \mid \mathbf{s_k}, \tilde{\boldsymbol{\theta}}_\mathbf{k})$$

$$p(\mathbf{y} \mid \mathbf{s_i}, \tilde{\boldsymbol{\theta}}_\mathbf{i}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(- \frac{\left\| \mathbf{y} - \mathbf{s_i}(\tilde{\boldsymbol{\theta}}_\mathbf{i}) \right\|^2}{2\sigma^2} \right)$$

$$\tilde{\boldsymbol{\theta}}_\mathbf{i} = \operatorname{argmax}_{\boldsymbol{\theta} \in \Theta} p(\mathbf{y} \mid \mathbf{s_i}, \boldsymbol{\theta})$$

This denotes image $\mathbf{s_i}$ deformed by pose parameter tilde(θ)$_i$. By "deformation" we mean rotation and/or translation in this example. In general, it could mean any other type of transformation including geometric scaling, blurring, etc.

Matched filter: special case of GMLC

- If the parameters **θ** denoted pure translation, then one way to classify **y** is to determine for which *i*, the following quantity is maximized:

  $$\int \mathbf{s}_{\mathbf{i}}(\mathbf{t}-\boldsymbol{\theta})\mathbf{y}(\mathbf{t})d\mathbf{t}$$

  where **t** = (x,y) denotes spatial coordinates.

- This is called a **matched filter** and it is equivalent to GMLC if all the candidate signals $\mathbf{s}_\mathrm{i}$ had the same magnitude, and we assume additive white Gaussian noise.

Compressive Classification: MLC

- Consider the following compressive acquisition model:

$$y = \Phi x + \eta, \eta \sim N(0, \sigma^2),$$
$$y \in R^m, x \in R^n, \Phi \in R^{m \times n}, m << n$$

- The MLC for this case is now:

$$j = \arg\max_{k \in \{1,2,\ldots,P\}} p(\mathbf{y} \mid \Phi \mathbf{s_k})$$
$$p(\mathbf{y} \mid \Phi \mathbf{s_i}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left\| \mathbf{y} - \Phi \mathbf{s_i} \right\|^2}{2\sigma^2} \right)$$

Compressive Classification: GMLC

- But the compressive measurement **y** could be acquired from an image which was acquired in a different pose than any of the images in the database.
- The GMLC is now given as:

$$j = \arg\max_{k \in \{1,2,\ldots,P\}} p(\mathbf{y} \mid \Phi \mathbf{s_k}, \tilde{\boldsymbol{\theta}}_\mathbf{k})$$

$$p(\mathbf{y} \mid \Phi \mathbf{s_i}, \tilde{\boldsymbol{\theta}}_\mathbf{i}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(- \frac{\left\| \mathbf{y} - \Phi \mathbf{s_i}(\tilde{\boldsymbol{\theta}}_\mathbf{i}) \right\|^2}{2\sigma^2} \right)$$

$$\tilde{\boldsymbol{\theta}}_\mathbf{i} = \mathrm{argmax}_{\boldsymbol{\theta} \in \Theta}\; p(\mathbf{y} \mid \Phi \mathbf{s_i}, \boldsymbol{\theta})$$

This has an interesting name – the **smashed filter** (derived from the name "matched filter"), taking into account the compressive nature of the measurements.

RIP and classification

- These are all essentially nearest neighbour classifiers with Euclidean distance.
- What is special about these classifiers?
- Let us assume that the sensing matrix **Φ** (and hence **ΦU** for any orthonormal **U**) obey the restricted isometry property.
- Then for *k*-sparse signals $\mathbf{s_1}$ and $\mathbf{s_2}$ and RIC δ_{2k}, we have:

$$(1-\delta_{2k})\|s_1 - s_2\|^2 \le \|\Phi(s_1 - s_2)\|^2 \le (1+\delta_{2k})\|s_1 - s_2\|^2$$

RIP and classification

- If m is sufficiently large and $\mathbf{\Phi}$ obeys RIP, we know that the distance between any two sparse vectors in the original n-dimensional space is preserved when you multiply them by $\mathbf{\Phi}$**.**
- In other words, under the conditions of sparsity and RIP, the distance between two compressive measurements is approximately equal to the distance between the original signals.
- The RIP helps classification here – but m needs to be carefully chosen – for an appropriate k.
- Basically this means that the nearest neighbour algorithm from compressive measurements will work almost as well as from the original measurements.

Theorem by Baraniuk and Wakin

- Consider a manifold with *K* degrees of freedom in R^n. Consider $0 < \varepsilon < 1$, $0 < \rho < 1$ and a random matrix $\boldsymbol{\Phi}$ in $R^{m \times n}$ with orthonormal rows where m is $O(K \log(n) \log(1/\rho)/\varepsilon^2)$. If $m \leq n$, then with probability at least $1-\rho$, we have the following:

$$(1-\varepsilon)\sqrt{\frac{m}{n}} \leq \frac{\left\|\Phi(x_1 - x_2)\right\|_2}{\left\|x_1 - x_2\right\|_2} \leq (1+\varepsilon)\sqrt{\frac{m}{n}}$$

where x_1 and x_2 are any two points on the manifold.

Theorem by Baraniuk and Wakin

- Clearly for large *m*, the distances will be preserved better – as per this theorem.
- This theorem makes no assumption on signal sparsity, but just on the dimension of the manifold.

Results

(a) Tank (b) School Bus (c) Truck

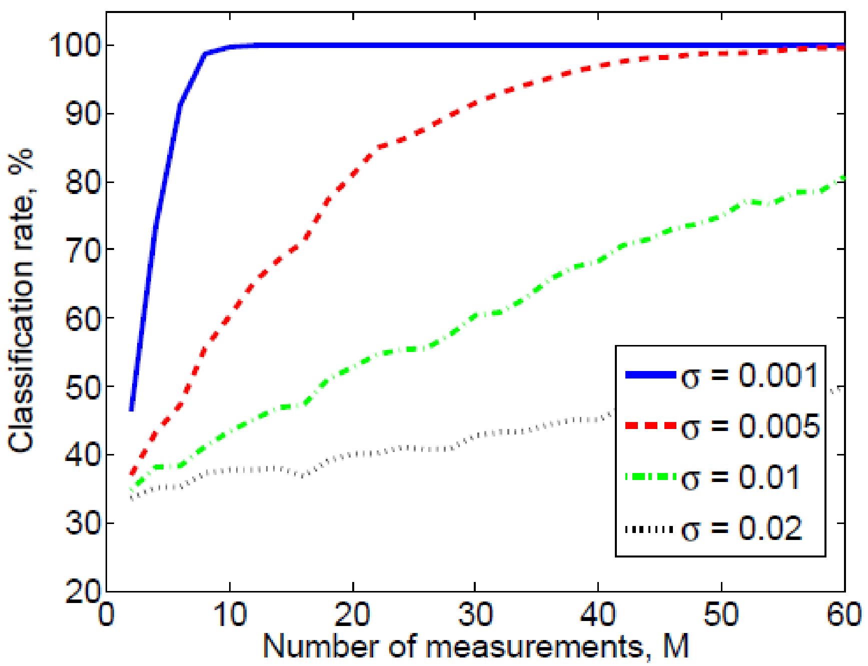
Figure 3. *Models used for classification experiments.*

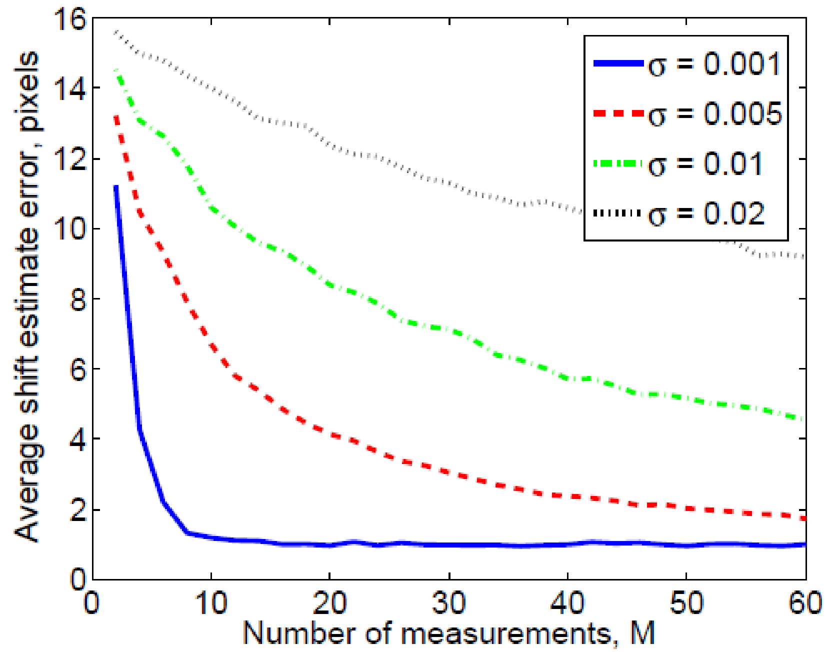
Image source: [Davenport et al, "The smashed filter for compressive classification and target recognition"]

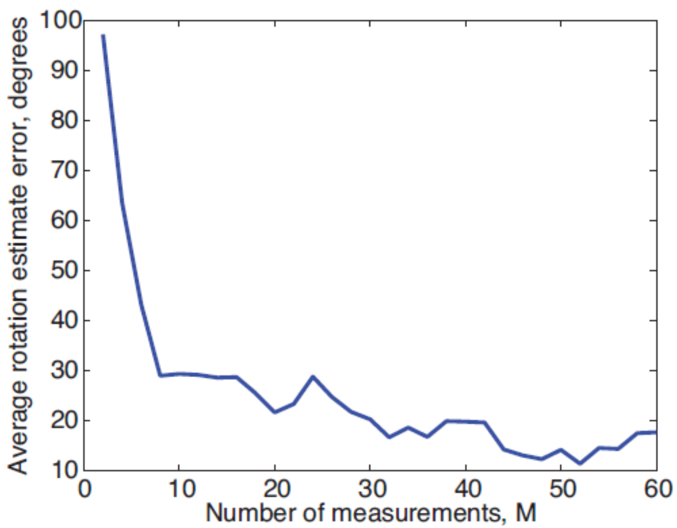
Image size: 128 x 128

Compressive measurements taken by a Rice single pixel camera. Though the sensing matrix of the camera does not obey RIP since it contains values that are 0 or 1, it can be converted into a matrix with entries that are either -1 or +1. This is by taking two measurements of the same scene, where the second measurement is taken by flipping the 0 and 1 values in the first sensing matrix.

Results







[Image source: Davenport et al, "The smashed filter for compressive classification and target recognition"]()