Sharp Thresholds in Statistical Estimation

David Donoho, Iain Johnstone, Andrea Montanari
Mohsen Bayati, Adel Javanmard, Arian Maleki

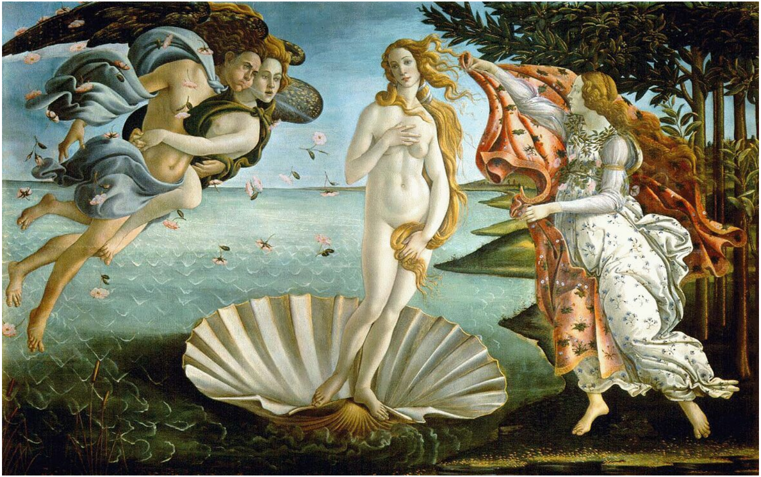
Stanford University

November 4, 2011

Why are graphical models useful?

Represent/exploit structure in
high-dimensional, highly-structured objects.

A high-dimensional, highly-structured object

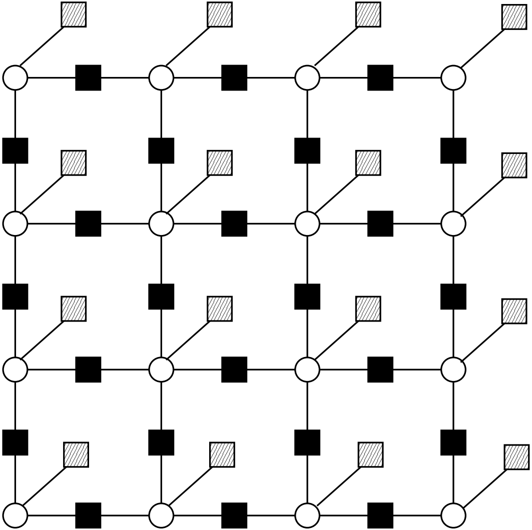


Structure?

Example

Nearby pixels have 'similar' colors (with the exception of edges!)

Structure?



What is ‘structure’ good for?

- Compression
- Denoising
- Parsimonious sensing/sampling
- Inference
- Interpretation
- Comprehension

Outline

- The hidden connection between denoising and compressed sensing
- Better tradeoff from better denoisers
- Better tredeoff via spatial coupling

Outline

- The hidden connection between denoising and compressed sensing
- Better tradeoff from better denoisers
- Better tredeoff via spatial coupling (cf. Marc's talk)

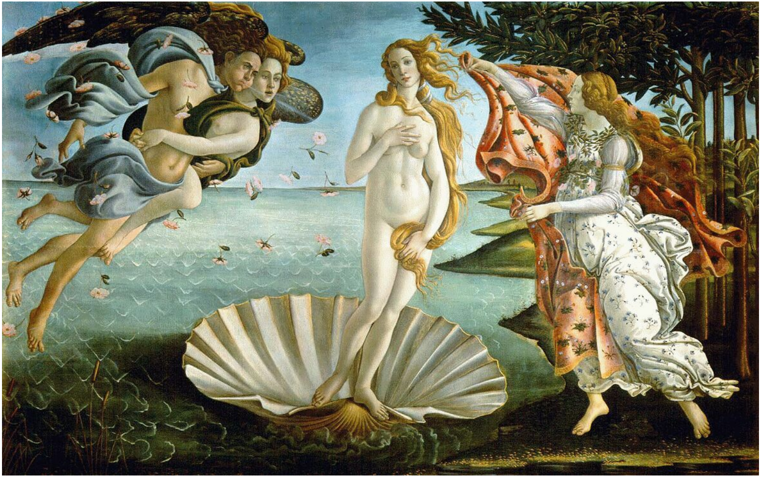
Outline

- The hidden connection between denoising and compressed sensing
- Better tradeoff from better denoisers
- Better tredeoff via spatial coupling (can dobetter than Marc ? ;-))

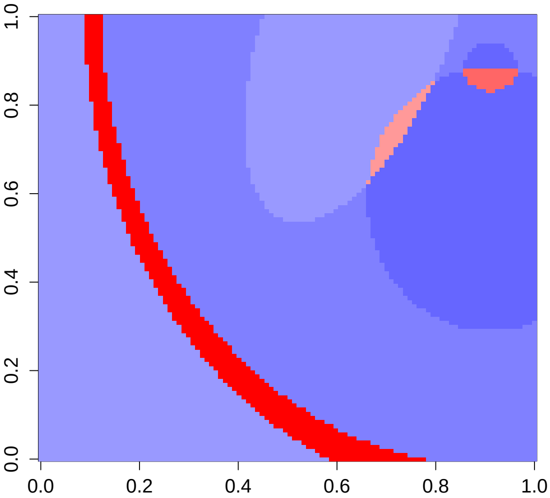
The hidden connection

[Donoho, Johnstone, Montanari, arXiv/*monday*]
my webpage

I would have loved to use this as running example



Something a tad less interesting

$x =$  $\in \mathbb{R}^{n=108\times 108}$

Corrupting it

$$y = x + \sigma z$$

$$z = (z_1, \ldots, z_n) \in \mathbb{R}^n, \qquad z_i \sim \mathrm{N}(0,1)$$

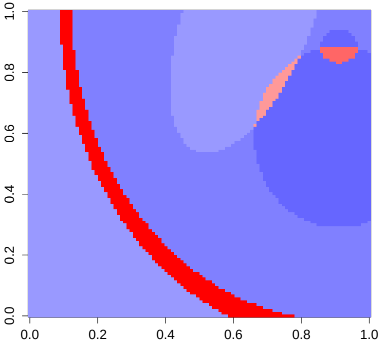
Corrupting it

$$y = x + \sigma z$$

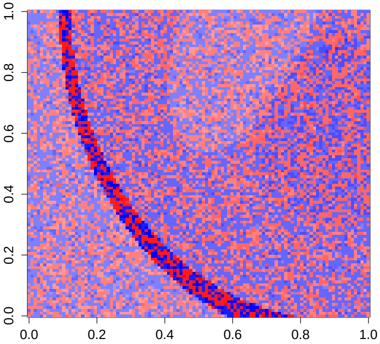
$$z = (z_1, \ldots, z_n) \in \mathbb{R}^n, \qquad z_i \sim \mathrm{N}(0, 1)$$

Corrupting it ($\sigma = 0.1$)

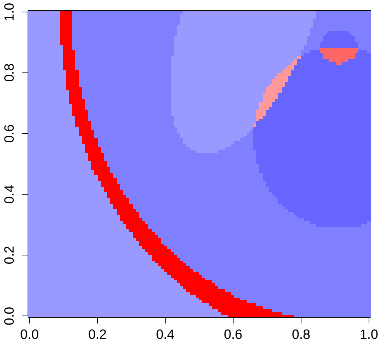
$x =$

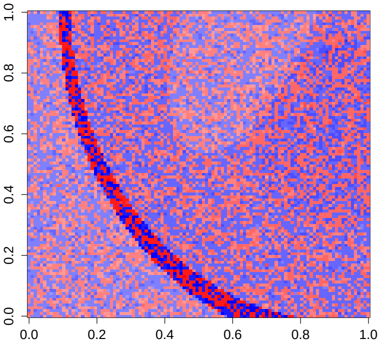


$y =$



Corrupting it ($\sigma = 0.1$)

$x =$ 

$y =$ 

Can we use the image structure to alleviate the noise?

Denoiser

$$\eta(\,\cdot\,;\lambda) : \mathbb{R}^n \to \mathbb{R}^n$$

$\lambda =$ tuning parameter

$y \mapsto \widehat{x} = \eta(y;\lambda)$ reconstruction

Encodes our idea of structure.

Denoiser

$$\eta(\,\cdot\,;\lambda) : \mathbb{R}^n \to \mathbb{R}^n$$

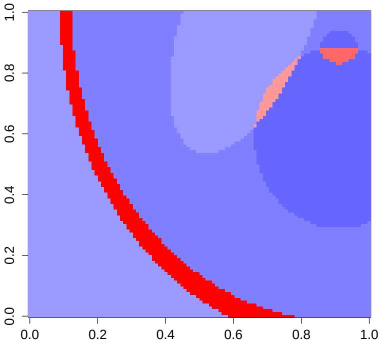
$\lambda =$ tuning parameter

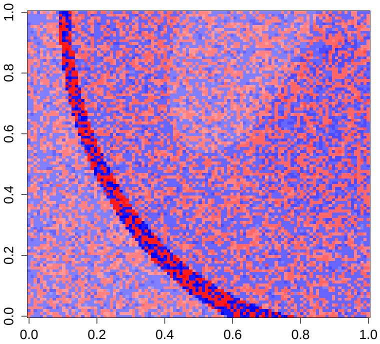
$y \mapsto \widehat{x} = \eta(y;\lambda)$ reconstruction

Encodes our idea of structure.

Structure?

Few edges, and mostly uniform color

$x =$ 

$y =$ 

Structure?

Few edges, and mostly uniform color

$$\text{minimize} \qquad \mathcal{C}_{\lambda,y}(x) \equiv \frac{1}{2}\|y - x\|_2^2 + \lambda\|x\|_{\rm TV}\,,$$

$$\|x\|_{\rm TV} \equiv \sum_{(i,j)\in \text{Edges}(2\text{DGrid})} |x_i - x_j|\,.$$

$$\eta(y;\lambda) \equiv \arg\min_{x\in\mathbb{R}^n} \mathcal{C}_{\lambda,y}(x)\,.$$

[Rudin,Osher,Fatemi 1992]

Structure?

Few edges, and mostly uniform color

$$\text{minimize} \qquad \mathcal{C}_{\lambda,y}(x) \equiv \frac{1}{2}\|y - x\|_2^2 + \lambda\|x\|_{\mathrm{TV}}\,,$$

$$\|x\|_{\mathrm{TV}} \equiv \sum_{(i,j)\in \mathrm{Edges}(\mathrm{2DGrid})} |x_i - x_j|\,.$$

$$\eta(y;\lambda) \equiv \arg\min_{x\in\mathbb{R}^n} \mathcal{C}_{\lambda,y}(x)\,.$$

[Rudin,Osher,Fatemi 1992]

Structure?

Few edges, and mostly uniform color

$$\text{minimize} \qquad \mathcal{C}_{\lambda,y}(x) \equiv \frac{1}{2}\|y - x\|_2^2 + \lambda\|x\|_{\mathrm{TV}}\,,$$

$$\|x\|_{\mathrm{TV}} \equiv \sum_{(i,j)\in \mathrm{Edges(2DGrid)}} |x_i - x_j|\,.$$

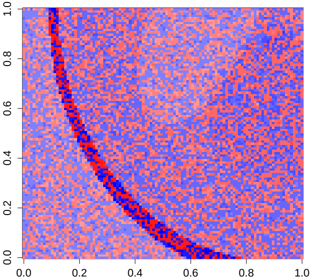
$$\eta(y;\lambda) \equiv \arg\min_{x\in\mathbb{R}^n} \mathcal{C}_{\lambda,y}(x)\,.$$

[Rudin,Osher,Fatemi 1992]

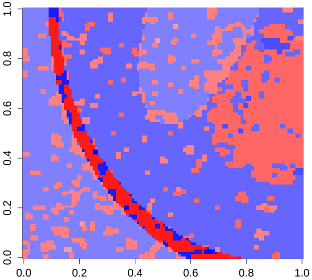
Does it work?

Does it work? ($\sigma = 0.1$)

$y =$

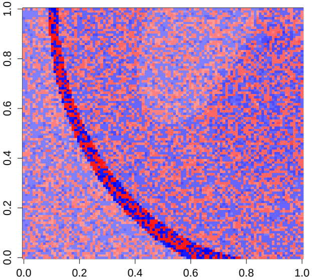


$\eta(y; \lambda) =$

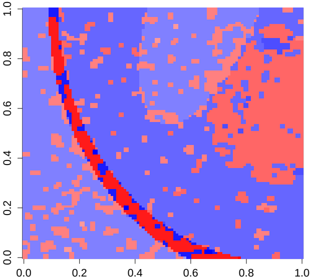


How do we pick λ? ($\sigma = 0.1$)

$y =$

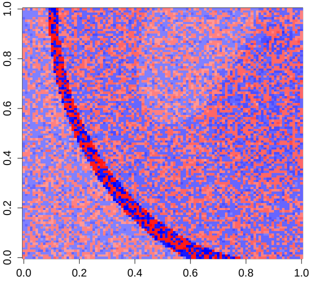


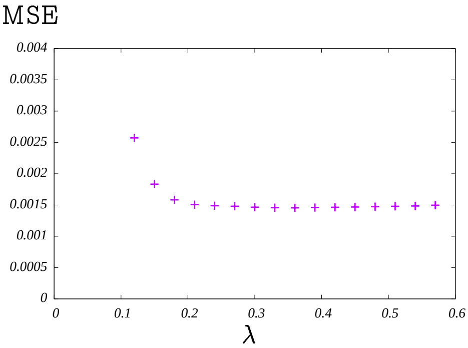
$\eta(y; \lambda) =$



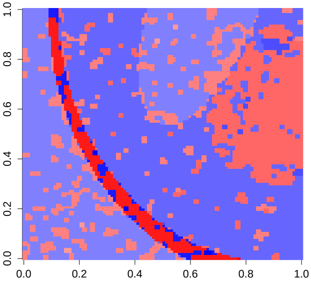
How do we pick λ? ($\sigma = 0.1$)

$y =$



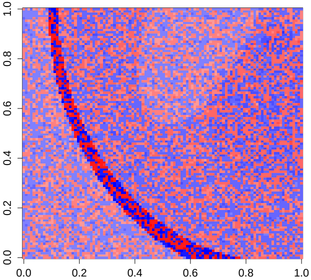


$\eta(y;\lambda) =$

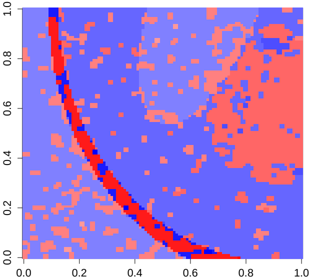


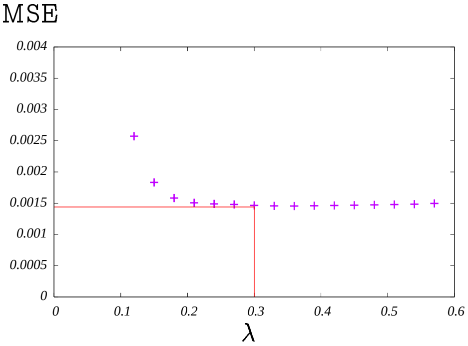
How do we pick λ? ($\sigma = 0.1$)

$y =$



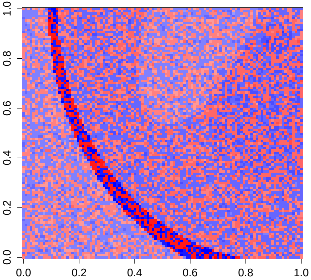
$\eta(y; \lambda) =$



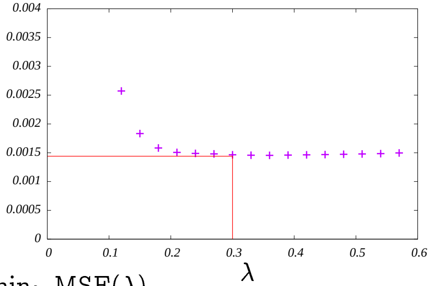


How do we pick λ? ($\sigma = 0.1$)

$y =$

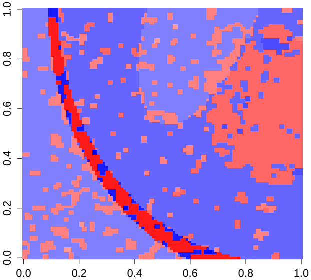


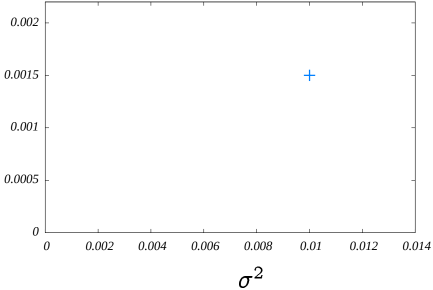
MSE



$\min_\lambda$ MSE(λ)

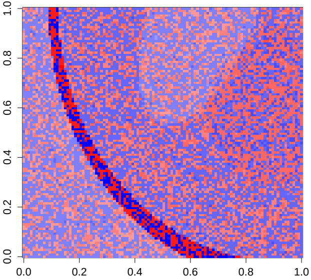
$\eta(y;\lambda) =$



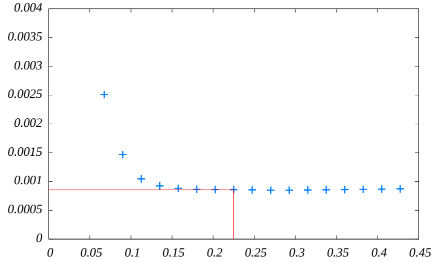


How do we pick λ? ($\sigma = 0.075$)

$y =$

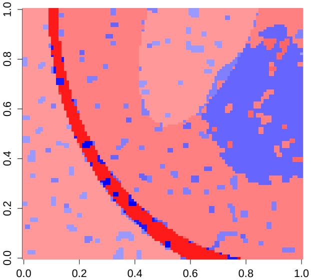


MSE

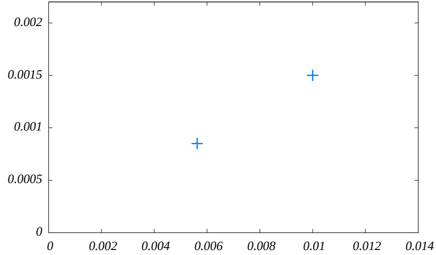


λ

$\eta(y;\lambda) =$



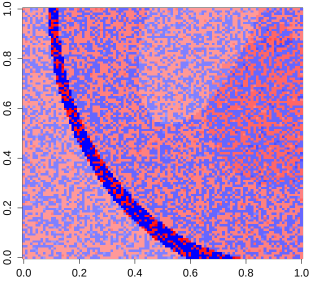
$\min_\lambda \ \mathrm{MSE}(\lambda)$



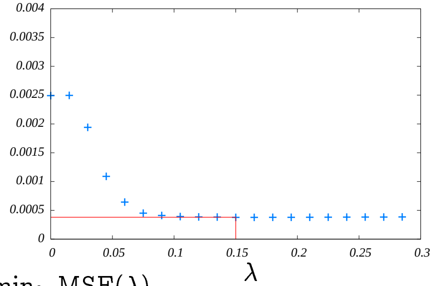
σ^2

How do we pick λ? ($\sigma = 0.05$)

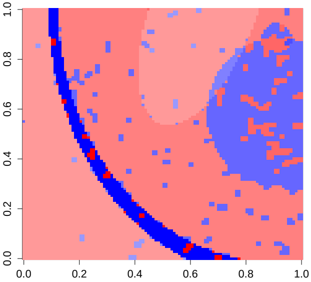
$y =$



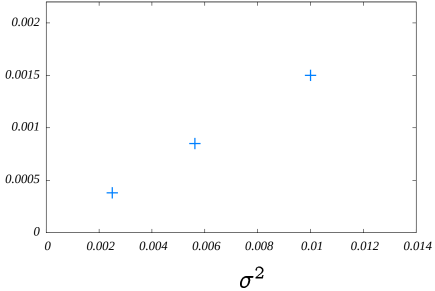
MSE



$\eta(y;\lambda) =$

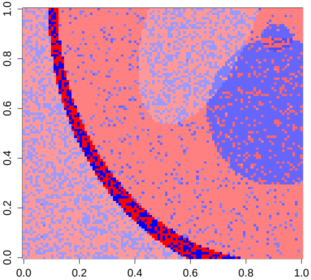


$\min_\lambda \text{ MSE}(\lambda)$

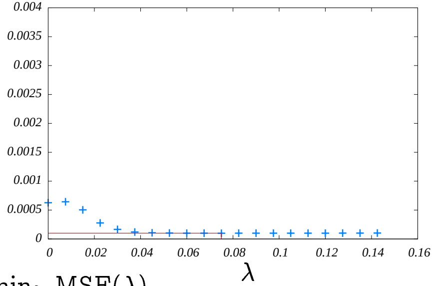


How do we pick λ? ($\sigma = 0.025$)

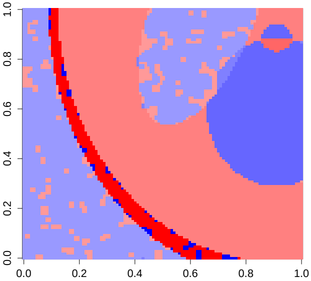
$y =$



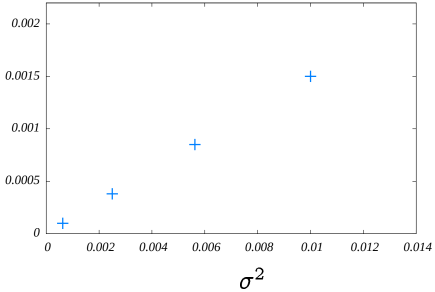
MSE



$\eta(y;\lambda) =$



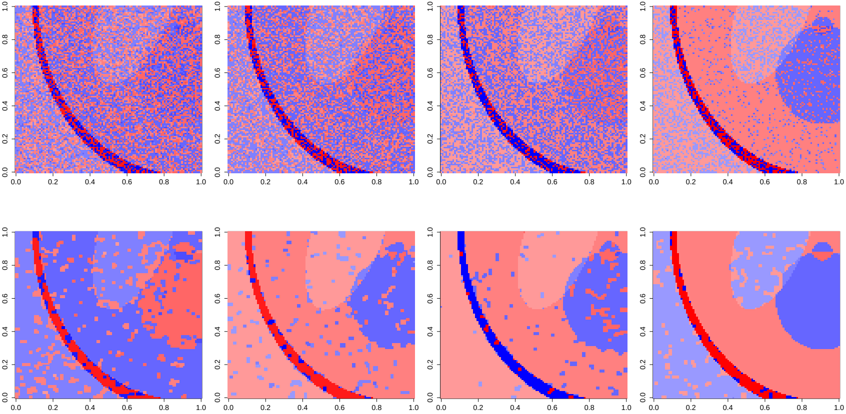
$\min_\lambda$ MSE(λ)



Everything together

y

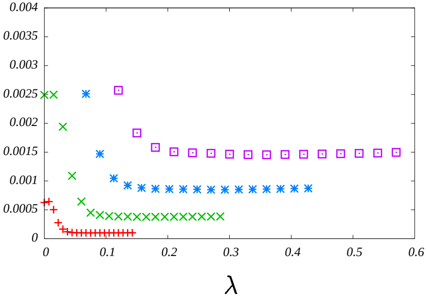
$\eta(y;\lambda)$



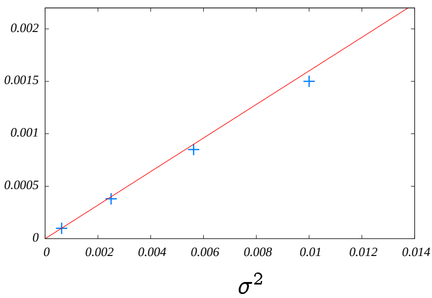
$\sigma = 0.1$ $\sigma = 0.075$ $\sigma = 0.05$ $\sigma = 0.025$

Everything together

MSE

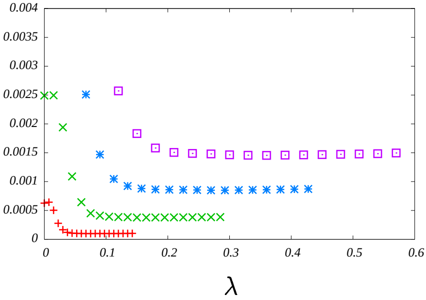


$\min_\lambda$ MSE(λ)

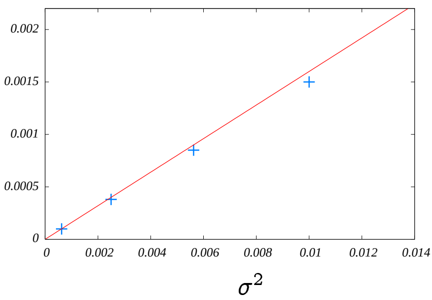


Everything together

MSE



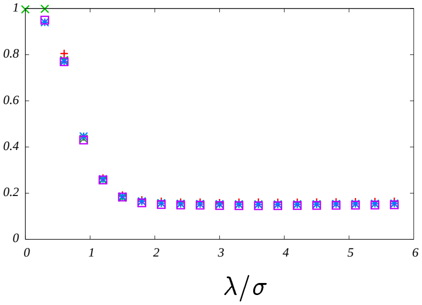
$\min_\lambda$ MSE(λ)



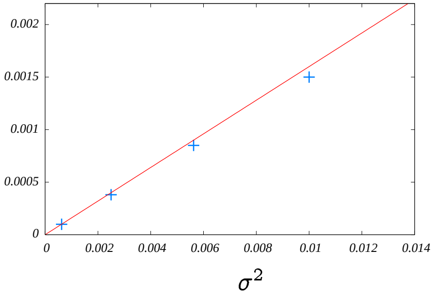
$$\min_\lambda \mathrm{MSE} \simeq M(x)\,\sigma^2, \qquad M(x) \approx 0.17$$

Everything together

MSE/σ^2



$\min_\lambda\ \mathrm{MSE}(\lambda)$



$$\min_\lambda \mathrm{MSE} \simeq M(x)\,\sigma^2, \qquad M(x) \approx 0.17$$

Let's do something more modern!

Use structure for parsimonious sensing

(compressed sensing)

[Lots of papers to be cited]

Let's do something more modern!

Use structure for parsimonious sensing
(compressed sensing)

[Lots of papers to be cited]

Let's do something more modern!

$$y = Ax$$

$y \in \mathbb{R}^m, \quad m \ll n$

$A \in \mathbb{R}^{m \times n} \quad$ sensing matrix

Let's do something more modern!

$$y = Ax$$

$$y \in \mathbb{R}^m, \quad m \ll n$$

$$A \in \mathbb{R}^{m \times n} \quad \text{sensing matrix}$$

Let's do something more modern!

$$y = Ax$$

$$y \in \mathbb{R}^m, \quad m \ll n$$

$$A \in \mathbb{R}^{m \times n} \quad \text{sensing matrix}$$

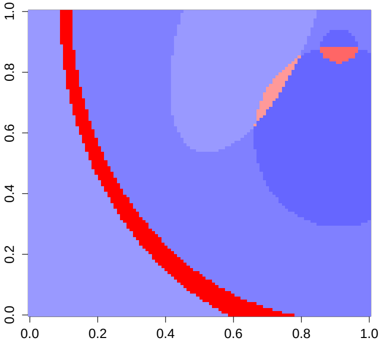
Let's do something more modern!

Throughout: $A_{ij} \sim_{i.i.d.} \mathrm{N}(0,1)$

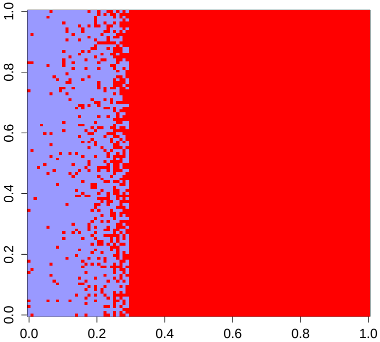
(other distibutions OK: eg Uniform($\{+1,-1\}$))

Sensing it ($\delta = m/n = 0.27$)

$x =$



$y =$



Using structure to recover the image

Few edges, and mostly uniform color

$$\begin{aligned} \text{minimize} \quad & \|x\|_{\mathrm{TV}}, \\ \text{subject to} \quad & y = Ax, \end{aligned}$$

Solution $\widehat{x}(y)$: Estimate of x

Using structure to recover the image

Few edges, and mostly uniform color

$$\begin{array}{rl} \text{minimize} & \|x\|_{\mathrm{TV}}, \\ \text{subject to} & y = Ax, \end{array}$$

Solution $\widehat{x}(y)$: Estimate of x

Using structure to recover the image

Few edges, and mostly uniform color

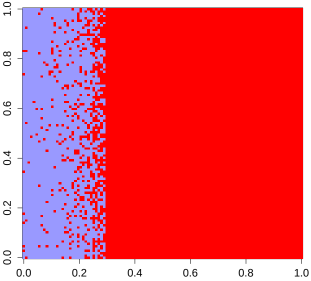
$$\begin{array}{ll} \text{minimize} & \|x\|_{\mathrm{TV}}, \\ \text{subject to} & y = Ax, \end{array}$$

Solution $\widehat{x}(y)$: Estimate of x

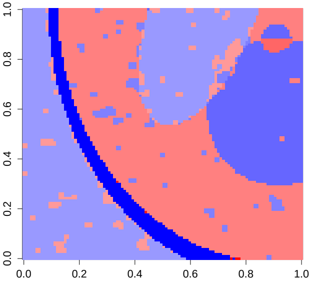
Does it work?

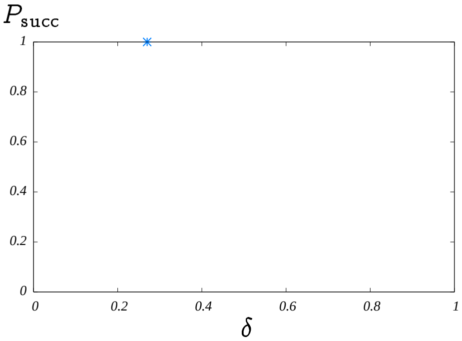
Does it work? ($\delta = m/n = 0.27$)

$y =$



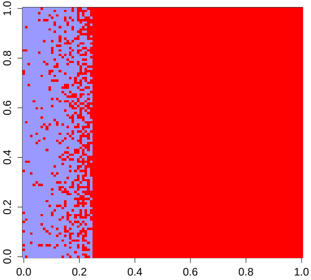
$\widehat{x}(y) =$



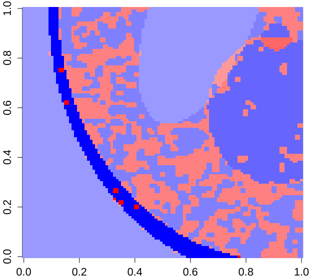


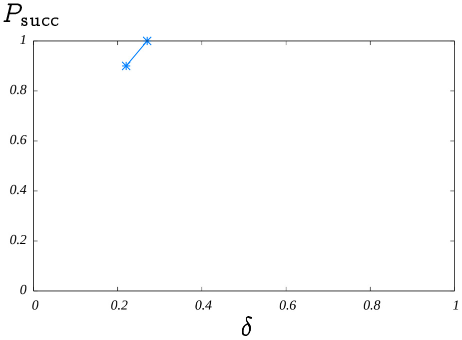
Does it work? ($\delta = m/n = 0.22$)

$y =$



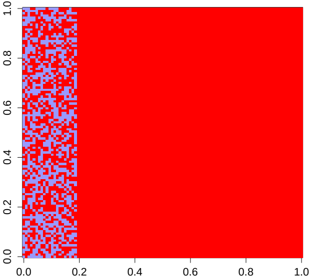
$\widehat{x}(y) =$



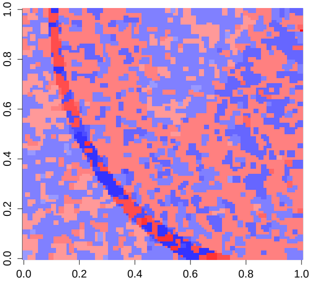


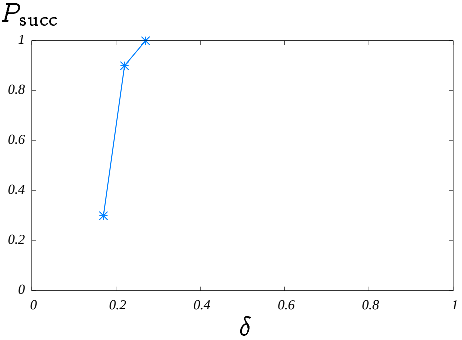
Does it work? ($\delta = m/n = 0.17$)

$y =$



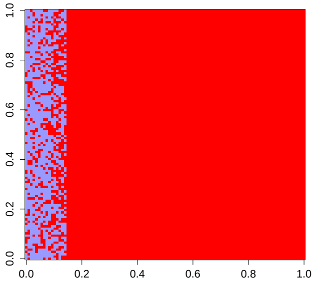
$\hat{x}(y) =$



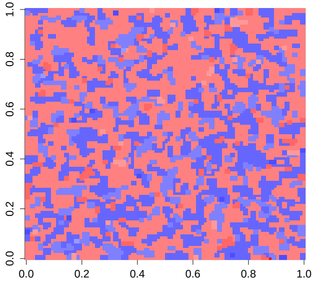


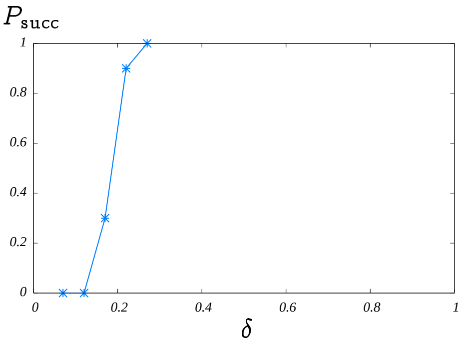
Does it work? ($\delta = m/n = 0.12$)

$y =$



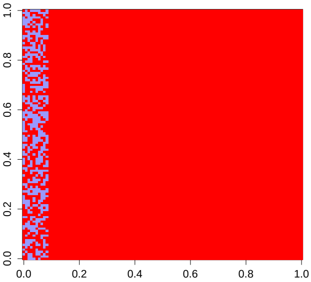
$\widehat{x}(y) =$



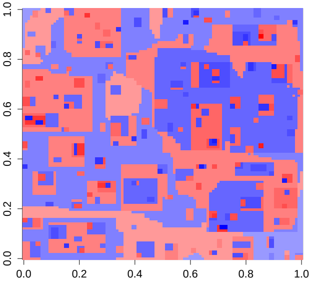


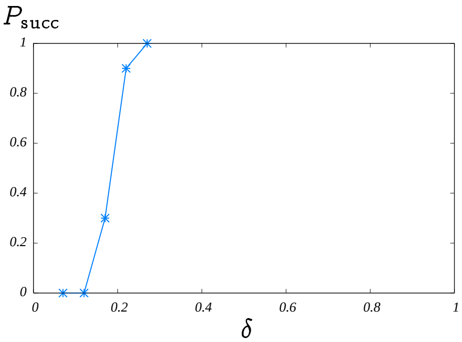
Does it work? ($\delta = m/n = 0.07$)

$y =$



$\widehat{x}(y) =$

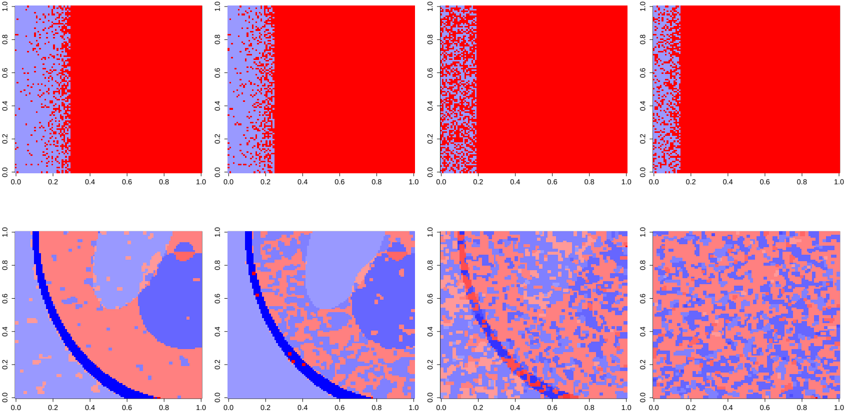




Everything together

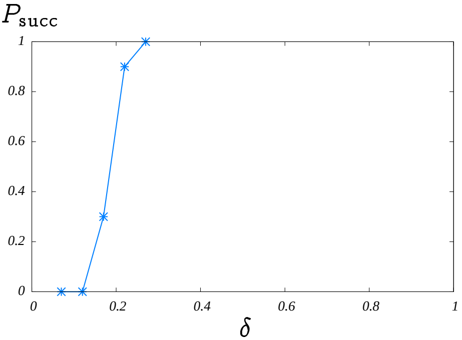
y

$\widehat{x}(y)$

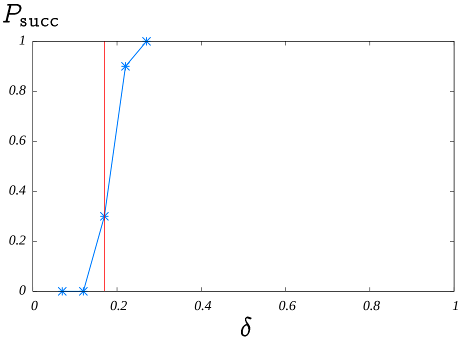


$\delta = 0.27$ $\delta = 0.22$ $\delta = 0.17$ $\delta = 0.12$

Everything together



Everything together



Recover with high probability $\Leftrightarrow \quad \delta < \delta(x), \qquad \delta(x) \approx 0.18$

Is this a coincidence?

$$\min_\lambda \mathrm{MSE} \simeq M(x)\,\sigma^2, \qquad M(x) \approx 0.17$$

$$\text{Recover with high probability} \Leftrightarrow \delta < \delta(x), \qquad \delta(x) \approx 0.18$$

A general mathematical formulation

Denoiser

$$J_\lambda : \mathbb{R}^n \to \mathbb{R} \qquad \text{regularization function}$$

$$\eta(y;\lambda) \equiv \arg\min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}\|y-x\|^2 + J_\lambda(x) \right\}$$

Compressed sensing reconstruction

$$\begin{aligned} \text{minimize} \quad & J_\lambda(x) \\ \text{subject to} \quad & y = Ax \end{aligned}$$

Performance parameters

Minimax denoising error

$$M_n(x) \;=\; \inf_{\lambda} \sup_{\sigma>0} \frac{1}{\sigma^2 n} \mathbb{E}\Big\{\|\eta(x+\sigma\, z;\lambda) - x\|_2^2\Big\}\,,$$

Compressed sensing threshold

$$\delta_n(x) \;=\; \inf\Big\{\,\delta > 0\;:\quad \mathbb{P}\{\widehat{x}(y) = x\} < 0.001\Big\}$$

Performance parameters

Minimax denoising error

$$M_n(x) \;=\; \inf_{\lambda}\sup_{\sigma>0}\frac{1}{\sigma^2 n}\,\mathbb{E}\Big\{\|\eta(x+\sigma\,z;\lambda)-x\|_2^2\Big\}\,,$$

Compressed sensing threshold

$$\delta_n(x) \;=\; \inf\Big\{\,\delta>0\;:\quad \mathbb{P}\{\widehat{x}(y)=x\}<0.001\Big\}$$

Loosely speaking

Conjecture

For large n

$$M_n(x) \approx \delta_n(x)$$

More precisely: Class of instances

$$\mathcal{F}_{n,\varepsilon} \equiv \text{class of images } x \in \mathbb{R}^n,$$

$$\varepsilon \equiv \text{generalized sparsity parameter.}$$

More precisely

$$M(\varepsilon) \equiv \lim_{n\to\infty} \sup_{x\in\mathcal{F}_{n,\varepsilon}} M_n(x),$$
$$\delta(\varepsilon) \equiv \lim_{n\to\infty} \sup_{x\in\mathcal{F}_{n,\varepsilon}} \delta_n(x).$$

Conjecture

For a number of signal classes $\mathcal{F}_{n,\varepsilon}$ and denoisers

$$M(\varepsilon) = \delta(\varepsilon)$$

More precisely

$$
\begin{aligned}
M(\varepsilon) &\equiv \lim_{n\to\infty} \sup_{x\in\mathcal{F}_{n,\varepsilon}} M_n(x)\,, \\
\delta(\varepsilon) &\equiv \lim_{n\to\infty} \sup_{x\in\mathcal{F}_{n,\varepsilon}} \delta_n(x)\,.
\end{aligned}
$$

Conjecture

For a number of signal classes $\mathcal{F}_{n,\varepsilon}$ and denoisers

$$M(\varepsilon) = \delta(\varepsilon)$$

What is known?

Proved for

- $\mathcal{F}_{n,\varepsilon}$ = vectors with at most $n\varepsilon$ nonzeros
- $\eta = \ell_1$ denoising

[slightly weaker form: Bayati-Montanar 2011, Donoho-Maleki-Montanari 2011]

Extensively tested for

- $\mathcal{F}_{n,\varepsilon}$ = sparse, block-sparse, monotone, TV class
- η = soft, firm, minimax, block-thresholding, James-Stein, monoreg, TV-denoising.

The missing connection: The approximate message passing (AMP) algorithm

Iterative CS reconstruction algorithm:

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^T r^t; \lambda_t) \\ r^t &= y - Ax^t + \mathsf{b}_t\, r^{t-1} \end{aligned}$$

Where

- λ_t any sequence of thresholds
- $\mathsf{b}_t \equiv \operatorname{div} \eta(x^{t-1} + A^T r^{t-1}; \lambda_{t-1})/m$ (Onsager term)

The missing connection: The approximate message passing (AMP) algorithm

Iterative CS reconstruction algorithm:

$$
\begin{aligned}
x^{t+1} &= \eta(x^t + A^T r^t; \lambda_t) \\
r^t &= y - Ax^t + \mathsf{b}_t\, r^{t-1}
\end{aligned}
$$

Where

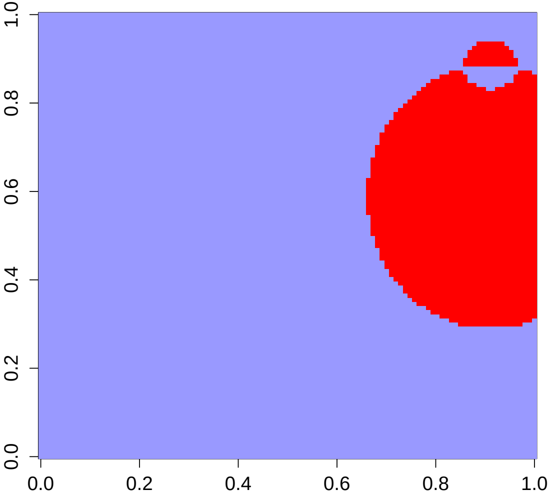
- λ_t any sequence of thresholds
- $\mathsf{b}_t \equiv \operatorname{div} \eta(x^{t-1} + A^T r^{t-1}; \lambda_{t-1})/m$ (Onsager term)

Better tradeoff from better denoisers

[Donoho, Johnstone, Montanari, arXiv/*monday*]

my webpage

Even less interesting image

$x =$  $\in \mathbb{R}^{n=108\times 108}$

'Mostly black' image.

Corrupting it

$$y = x + \sigma z$$

$$z = (z_1, \ldots, z_n) \in \mathbb{R}^n, z_i \sim \mathrm{N}(0, 1)$$

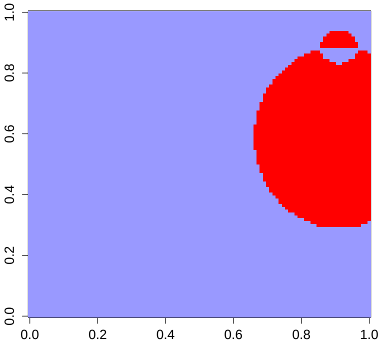
Corrupting it

$$y = x + \sigma z$$

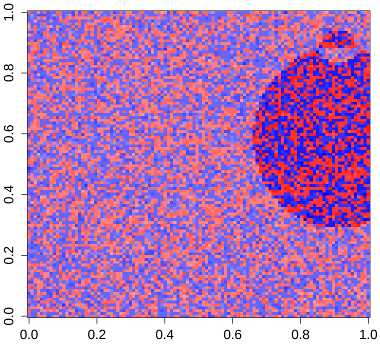
$$z = (z_1, \dots, z_n) \in \mathbb{R}^n, z_i \sim \mathrm{N}(0, 1)$$

Corrupting it ($\sigma = 0.1$)

$x =$



$y =$



Can we use the image structure to alleviate the noise?

Structure?

Mostly black, some white blocks

$$\text{minimize} \qquad C_{\lambda,y}(x) \equiv \frac{1}{2}\|y - x\|_2^2 + \lambda\|x\|_{\ell_2-\ell_1}\,,$$

$$\|x\|_{\ell_2-\ell_1} \equiv \sum_{B\in\text{Blocks}} \|x_B\|_2\,.$$

Blocks $\equiv$ Partition of image into blocks of size 9×9

$$\eta(y;\lambda) \equiv \arg\min_{x\in\mathbb{R}^n} C_{\lambda,y}(x)\,.$$

[Hall, Kerkyacharian, Picard 1998]

Structure?

Mostly black, some white blocks

$$\text{minimize} \qquad \mathcal{C}_{\lambda,y}(x) \equiv \frac{1}{2}\|y - x\|_2^2 + \lambda\|x\|_{\ell_2-\ell_1}\,,$$

$$\|x\|_{\ell_2-\ell_1} \equiv \sum_{B\in\text{Blocks}} \|x_B\|_2\,.$$

Blocks $\equiv$ Partition of image into blocks of size 9×9

$$\eta(y;\lambda) \equiv \arg\min_{x\in\mathbb{R}^n} \mathcal{C}_{\lambda,y}(x)\,.$$

[Hall, Kerkyacharian, Picard 1998]

Structure?

Mostly black, some white blocks

$$\text{minimize} \qquad \mathcal{C}_{\lambda,y}(x) \equiv \frac{1}{2}\|y - x\|_2^2 + \lambda \|x\|_{\ell_2 - \ell_1}\,,$$

$$\|x\|_{\ell_2-\ell_1} \equiv \sum_{B \in \text{Blocks}} \|x_B\|_2\,.$$

Blocks $\equiv$ Partition of image into blocks of size 9×9

$$\eta(y;\lambda) \equiv \arg\min_{x \in \mathbb{R}^n} \mathcal{C}_{\lambda,y}(x)\,.$$

[Hall, Kerkyacharian, Picard 1998]

Explicit form of the denoiser

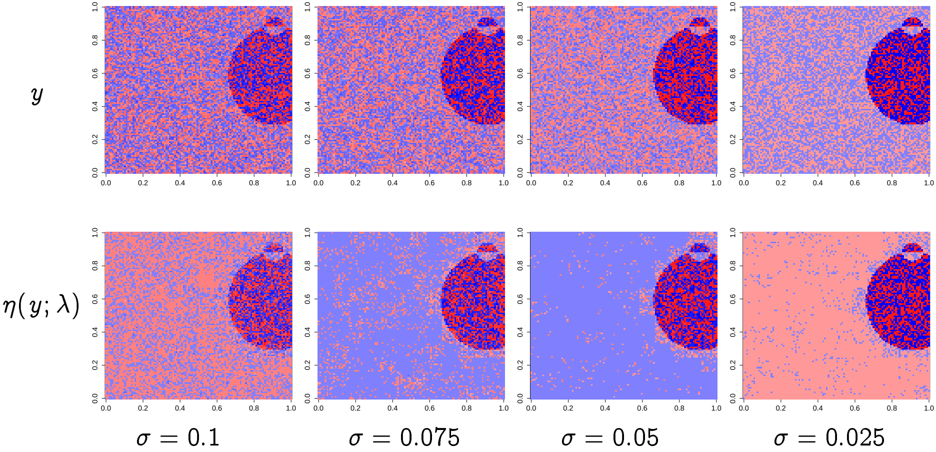
Seprable across blocks ($x_{B(i)} \in \mathbb{R}^{\ell\times\ell}$, $\ell = 9$)

$$\eta(x_{B(1)}, \dots, x_{B(K)}; \lambda) = \Big(\eta(x_{B(1)}; \lambda), \dots, \eta(x_{B(K)}; \lambda)\Big)$$

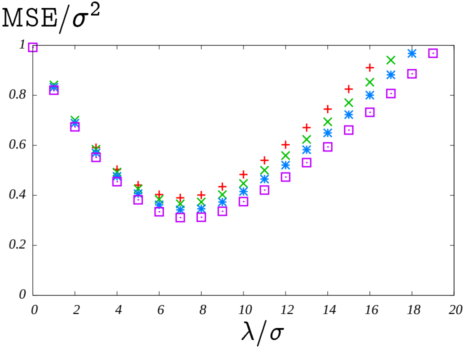
Soft thresholding inside each block

$$\eta(x_B; \lambda) = \Big(1 - \frac{\lambda}{\|x_B\|_2}\Big)_+ x_B\,.$$

Does it work?

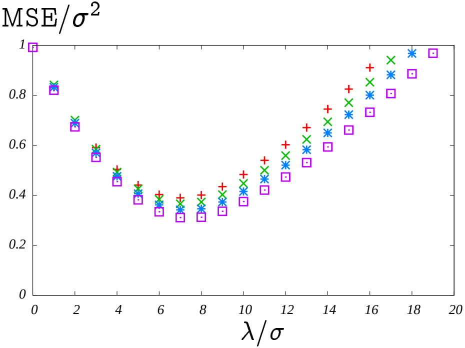


Everything together



$$\min_\lambda \text{MSE} \simeq M(x)\,\sigma^2, \qquad M(x) \approx 0.35$$

Everything together



$$\min_\lambda \text{MSE} \simeq M(x)\,\sigma^2, \qquad M(x) \approx 0.35$$

Compressed sensing

$$y = Ax$$

$y \in \mathbb{R}^m, \quad m \ll n$

$A \in \mathbb{R}^{m \times n}$ sensing matrix

Compressed sensing

$$y = Ax$$

$$y \in \mathbb{R}^m, \quad m \ll n$$

$$A \in \mathbb{R}^{m \times n} \quad \text{sensing matrix}$$

Compressed sensing

$$y = Ax$$

$$y \in \mathbb{R}^m, \quad m \ll n$$

$$A \in \mathbb{R}^{m \times n} \quad \text{sensing matrix}$$

Using structure to recover the image

Mostly black, some white blocks

$$\begin{array}{ll} \text{minimize} & \|x\|_{\ell_2-\ell_1}, \\ \text{subject to} & y = Ax, \end{array}$$

Solution $\widehat{x}(y)$: Estimate of x

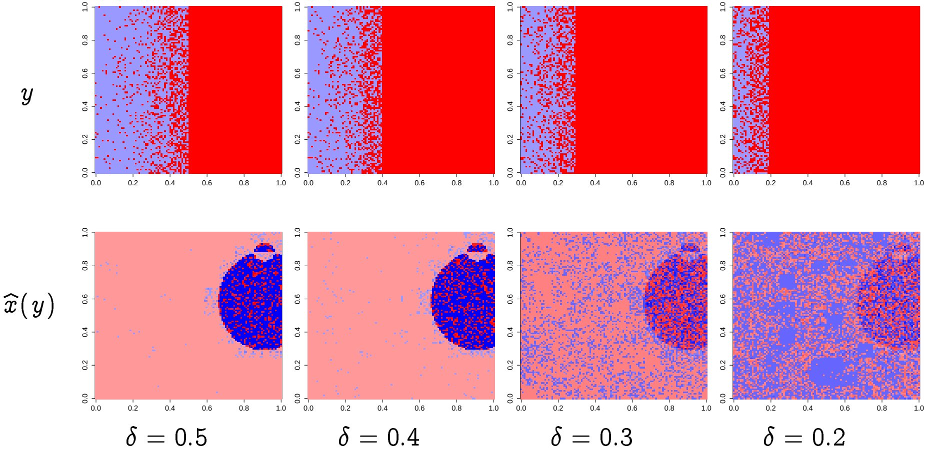
Using structure to recover the image

Mostly black, some white blocks

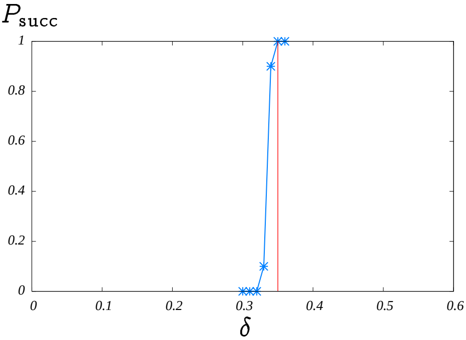
$$\begin{aligned} \text{minimize} \quad & \|x\|_{\ell_2 - \ell_1}, \\ \text{subject to} \quad & y = Ax, \end{aligned}$$

Solution $\widehat{x}(y)$: Estimate of x

Does it work?

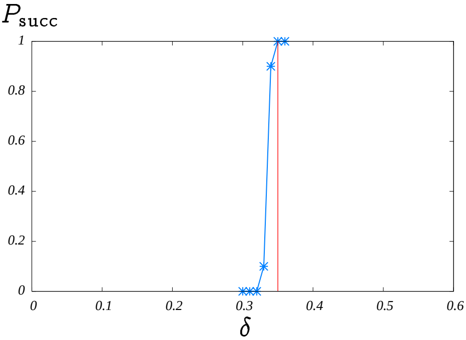


Everything together



Recover with high probability $\Leftrightarrow$ $\delta < \delta(x)$, $\delta(x) \approx 0.35$

Everything together



Recover with high probability $\Leftrightarrow \quad \delta < \delta(x), \qquad \delta(x) \approx 0.35$

Another coincidence?

$$\min_\lambda \mathrm{MSE} \simeq M(x)\,\sigma^2, \qquad M(x) \approx 0.35$$

$$\text{Recover with high probability} \Leftrightarrow \quad \delta < \delta(x), \qquad \delta(x) \approx 0.35$$

Idea

Better denoiser $\Rightarrow$ Better compressed sensing algorithm

A better denoiser

Block-soft thresholding

$$\eta(y_B;\lambda) = \left(1 - \frac{\lambda}{\|y_B\|_2}\right)_+ x_B \,.$$

The James-Stein positive-part estimator

$$\eta(y_B;\lambda) \;=\; \left(1 - \frac{\lambda}{\|y_B\|_2^2}\right)_+ y_B \,.$$

A better denoiser

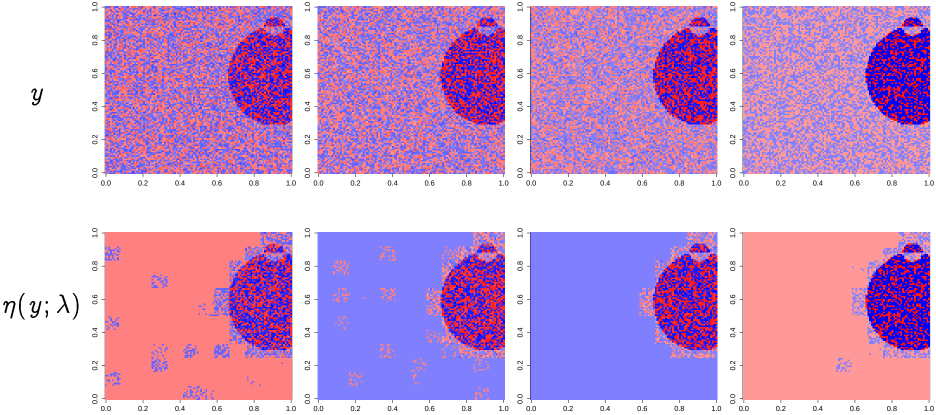
Block-soft thresholding

$$\eta(y_B;\lambda) = \Big(1 - \frac{\lambda}{\|y_B\|_2}\Big)_+ x_B\,.$$

The James-Stein positive-part estimator

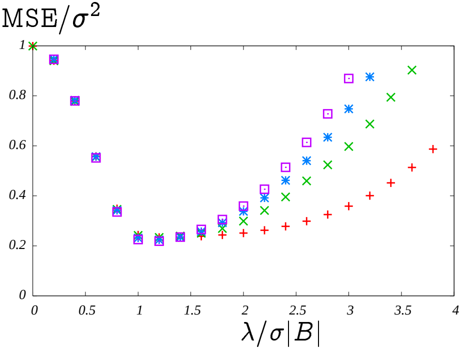
$$\eta(y_B;\lambda) \;\;=\;\; \Big(1 - \frac{\lambda}{\|y_B\|_2^2}\Big)_+ y_B\,.$$

Does it work?



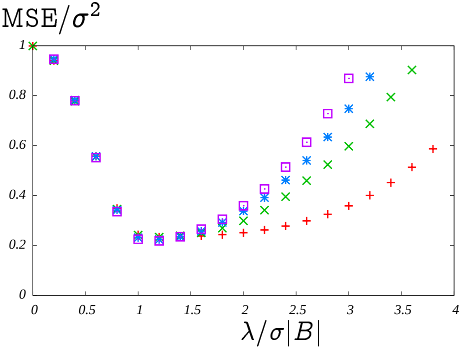
$\sigma = 0.1$ $\sigma = 0.075$ $\sigma = 0.05$ $\sigma = 0.025$

Does it work?



$$\min_\lambda \text{MSE} \simeq M(x)\,\sigma^2, \qquad M(x) \approx 0.2$$

Does it work?



$$\min_\lambda \text{MSE} \simeq M(x)\,\sigma^2, \qquad M(x) \approx 0.2$$

Compressed sensing

$$y = Ax$$

What about the algorithm???

AMP with James-Stein denoiser

Compressed sensing

$$y = Ax$$

What about the algorithm???

AMP with James-Stein denoiser

Compressed sensing

$$y = Ax$$

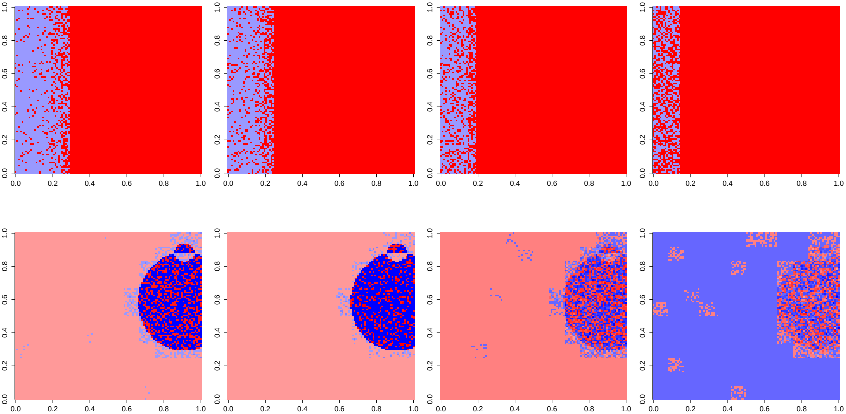
What about the algorithm???

AMP with James-Stein denoiser

Does it work?

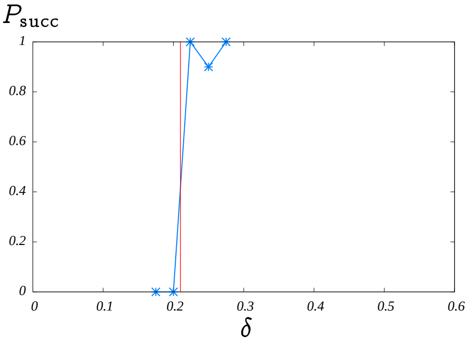
y

$\widehat{x}(y)$



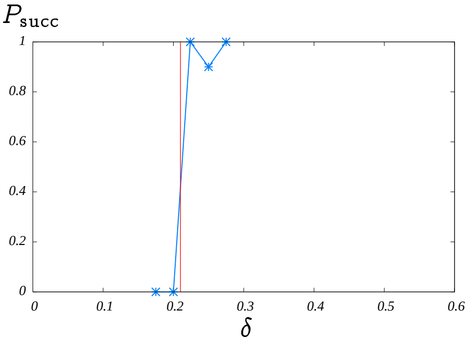
$\delta = 0.3$ $\delta = 0.25$ $\delta = 0.2$ $\delta = 0.15$

Does it work?



Recover with high probability $\Leftrightarrow \quad \delta < \delta(x), \qquad \delta(x) \approx 0.21$

Does it work?



Recover with high probability $\Leftrightarrow$ $\delta < \delta(x)$, $\delta(x) \approx 0.21$

JamesStein-AMP: Summarizing

$$M(x) \approx \delta(x) \approx 0.2 \qquad (< 0.35!)$$

Better denoiser $\Rightarrow$ Better reconstruction algorithm.

Adaptive!

JamesStein-AMP: Summarizing

$$M(x) \approx \delta(x) \approx 0.2 \qquad (< 0.35!)$$

Better denoiser $\Rightarrow$ Better reconstruction algorithm.

Adaptive!

JamesStein-AMP: Summarizing

$$M(x) \approx \delta(x) \approx 0.2 \qquad (< 0.35!)$$

Better denoiser $\Rightarrow$ Better reconstruction algorithm.

Adaptive!

A lot better!

Theorem (Donoho, Johnstone, Montanari)

For ε-block sparse vector, JS-AMP reconstructs the signal correctly for any

$$\delta > \varepsilon \, ,$$

provided $|B| \geq B_*(\varepsilon, \delta)$.

'As many measurements as equations'

A lot better!

Theorem (Donoho, Johnstone, Montanari)

For ε-block sparse vector, JS-AMP reconstructs the signal correctly for any

$$\delta > \varepsilon \, ,$$

provided $|B| \geq B_*(\varepsilon, \delta)$.

'As many measurements as equations'

Better tradeoff using spatial coupling

[Donoho, Javanmard, Montanari, arXiv/*in a couple of weeks*]

An example

$$\begin{aligned} x &= (x_1, \ldots, x_n), \quad x_i \sim_{i.i.d.} p_X \,, \\ y &= Ax \,, \quad y \in \mathbb{R}^m \,, \end{aligned}$$

$$p_X = 0.2\,\delta_0 + 0.3\,\delta_1 + 0.2\,\delta_{-1} + 0.2\,\delta_3 + 0.1\,\text{Uniform}(-2, 2).$$

p_X is known! Non-adaptive!

An example

$$
\begin{aligned}
x &= (x_1, \dots, x_n), \quad x_i \sim_{i.i.d.} p_X \,, \\
y &= Ax \,, \quad y \in \mathbb{R}^m \,,
\end{aligned}
$$

$$
p_X = 0.2\,\delta_0 + 0.3\,\delta_1 + 0.2\,\delta_{-1} + 0.2\,\delta_3 + 0.1\,\text{Uniform}(-2, 2).
$$

p_X is known! Non-adaptive!

An example

$$
\begin{aligned}
x &= (x_1, \ldots, x_n), \quad x_i \sim_{i.i.d.} p_X \,, \\
y &= Ax \,, \quad y \in \mathbb{R}^m \,,
\end{aligned}
$$

$$
p_X = 0.2\,\delta_0 + 0.3\,\delta_1 + 0.2\,\delta_{-1} + 0.2\,\delta_3 + 0.1\,\text{Uniform}(-2, 2).
$$

p_X is known! Non-adaptive!

How many measurements are needed?

$$p_X = 0.2\,\delta_0 + 0.3\,\delta_1 + 0.2\,\delta_{-1} + 0.2\,\delta_3 + 0.1\,\text{Uniform}(-2,2).$$

- Classical compressed sensing: $0.8\,n < m \lesssim 2.5\,n$
  (e.g. Candés-Recht 2011, adaptive, provably robust)
- Classical compressed sensing: $m = 0.97\,n + o(n)$
  (Donoho 2006, adaptive, Donoho-Maleki-M. 2011 provably robust)
- Marc's talk: $m = 0.8\,n + o(n)$
  (non-adaptive, robust)
- This talk: $m = 0.1\,n + o(n)$
  (non-adaptive, provably robust)

How many measurements are needed?

$$p_X = 0.2\,\delta_0 + 0.3\,\delta_1 + 0.2\,\delta_{-1} + 0.2\,\delta_3 + 0.1\,\text{Uniform}(-2,2).$$

- Classical compressed sensing: $0.8\,n < m \lesssim 2.5\,n$
  (e.g. Candés-Recht 2011, adaptive, provably robust)
- Classical compressed sensing: $m = 0.97\,n + o(n)$
  (Donoho 2006, adaptive, Donoho-Maleki-M. 2011 provably robust)
- Marc's talk: $m = 0.8\,n + o(n)$
  (non-adaptive, robust)
- This talk: $m = 0.1\,n + o(n)$
  (non-adaptive, provably robust)

How many measurements are needed?

$$p_X = 0.2\,\delta_0 + 0.3\,\delta_1 + 0.2\,\delta_{-1} + 0.2\,\delta_3 + 0.1\,\text{Uniform}(-2,2).$$

- Classical compressed sensing: $0.8\,n < m \lesssim 2.5\,n$
  (e.g. Candés-Recht 2011, adaptive, provably robust)
- Classical compressed sensing: $m = 0.97\,n + o(n)$
  (Donoho 2006, adaptive, Donoho-Maleki-M. 2011 provably robust)
- Marc's talk: $m = 0.8\,n + o(n)$
  (non-adaptive, robust)
- This talk: $m = 0.1\,n + o(n)$
  (non-adaptive, provably robust)

How many measurements are needed?

$$p_X = 0.2\,\delta_0 + 0.3\,\delta_1 + 0.2\,\delta_{-1} + 0.2\,\delta_3 + 0.1\,\text{Uniform}(-2,2).$$

- Classical compressed sensing: $0.8\,n < m \lesssim 2.5\,n$
  (e.g. Candés-Recht 2011, adaptive, provably robust)
- Classical compressed sensing: $m = 0.97\,n + o(n)$
  (Donoho 2006, adaptive, Donoho-Maleki-M. 2011 provably robust)
- Marc's talk: $m = 0.8\,n + o(n)$
  (non-adaptive, robust)
- This talk: $m = 0.1\,n + o(n)$
  (non-adaptive, provably robust)

How many measurements are needed?

$$p_X = 0.2\,\delta_0 + 0.3\,\delta_1 + 0.2\,\delta_{-1} + 0.2\,\delta_3 + 0.1\,\text{Uniform}(-2,2).$$

- Classical compressed sensing: $0.8\,n < m \lesssim 2.5\,n$
  (e.g. Candés-Recht 2011, adaptive, provably robust)
- Classical compressed sensing: $m = 0.97\,n + o(n)$
  (Donoho 2006, adaptive, Donoho-Maleki-M. 2011 provably robust)
- Marc's talk: $m = 0.8\,n + o(n)$
  (non-adaptive, robust)
- This talk: $m = 0.1\,n + o(n)$
  (non-adaptive, provably robust)

What is 0.1 here?

Definition (Renyi's Information Dimension)

For $X \sim p_X$, $\langle X \rangle_m$ an m-digits rounding of X

$$\overline{d}(X) \equiv \lim \sup_{m \to \infty} \frac{H(\langle X \rangle_m)}{m} .$$

Example: If

$$p_X = (1 - \varepsilon) \cdot \textit{discrete} + \varepsilon \cdot \textit{abs. continuous},$$

then $\overline{d}(X) = \varepsilon$.

What is 0.1 here?

Definition (Renyi's Information Dimension)

For $X \sim p_X$, $\langle X \rangle_m$ an m-digits rounding of X

$$\overline{d}(X) \equiv \lim\sup_{m \to \infty} \frac{H(\langle X \rangle_m)}{m}.$$

Example: If

$$p_X = (1 - \varepsilon) \cdot \textit{discrete} + \varepsilon \cdot \textit{abs. continuous},$$

then $\overline{d}(X) = \varepsilon$.

Why is this important?

Theorem (Verdu, Wu, 2010)

Under mild regularity hypotheses, non-adaptive compressed sensing is possible if and only if

$$m > \overline{d}(X)\, n + o(n)\,.$$

(equivalently, $\delta > \overline{d}(X) + o(1)$*).*

Shannon-theoretic argment. Exhaustive-search reconstruction.

Why is this important?

Theorem (Donoho-Javanmard-Montanari 2011)

Using spatially-coupled matrices and approximate message passing (AMP) reconstruction can recover x from

$$m > \overline{d}(X)\, n + o(n)\,.$$

measurements. Further, the approach is robust to noise.

Conclusion

- General connection: Sensing $\leftrightarrow$ denoising.
- Better ways to exploit data structure (e.g. graph-structured sparsity).
- Proof techniques/Algorithms $\rightarrow$ Graphical models.

Thanks!

Conclusion

- General connection: Sensing $\leftrightarrow$ denoising.
- Better ways to exploit data structure (e.g. graph-structured sparsity).
- Proof techniques/Algorithms $\rightarrow$ Graphical models.

Thanks!

Conclusion

- General connection: Sensing $\leftrightarrow$ denoising.
- Better ways to exploit data structure (e.g. graph-structured sparsity).
- Proof techniques/Algorithms $\rightarrow$ Graphical models.

Thanks!