MMSE Approximation for the Sparse Prior

Dror Simon

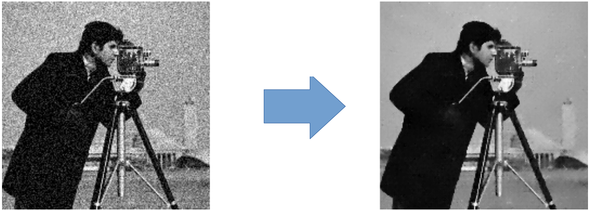
Computer Science, Technion, Israel

January 9, 2019

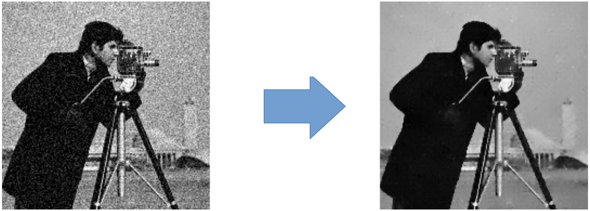
Joint work with Jeremias Sulam, Yaniv Romano, Yue M. Lu and Michael Elad

Noise Removal

Noise Removal

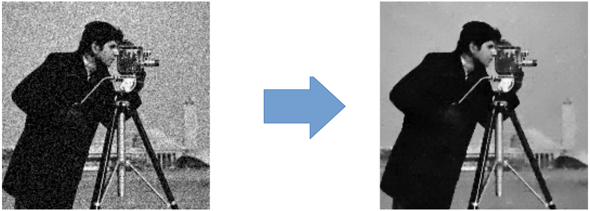


Noise Removal



Why denoising?

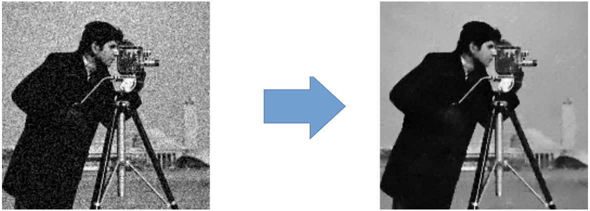
Noise Removal



Why denoising?

- A simple testing ground for novel concepts in signal processing.

Noise Removal



Why denoising?

- A simple testing ground for novel concepts in signal processing.
- Can be generalized to other, more complicated applications.

Noise Removal

Noisy Signal

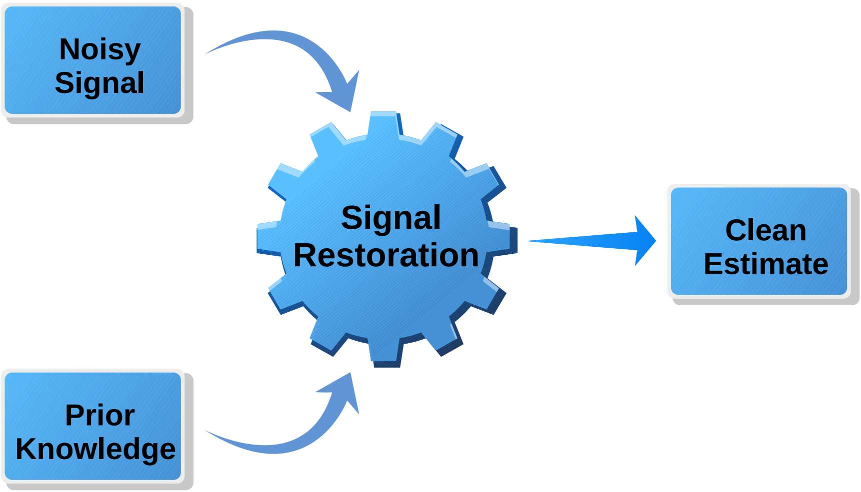
Noise Removal

Noisy Signal

Clean Estimate

Noise Removal

Noise Removal

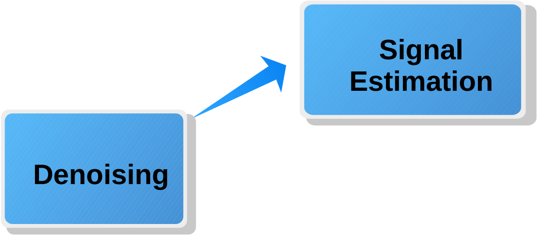


Noise Removal – Bayesian Standpoint

Noise Removal – Bayesian Standpoint

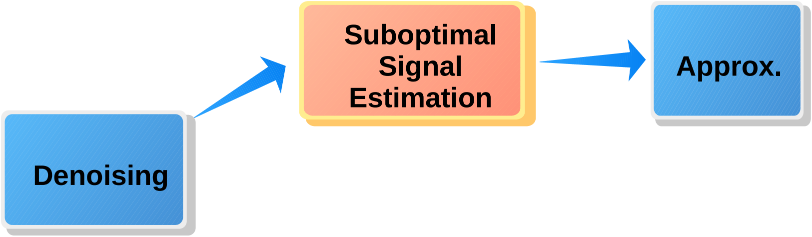
Denoising

Noise Removal – Bayesian Standpoint

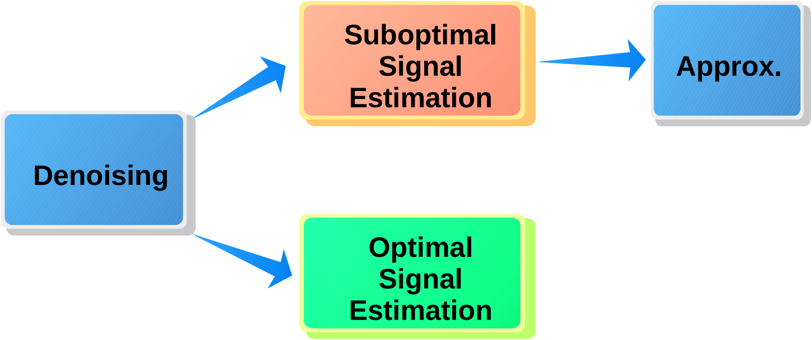


Noise Removal – Bayesian Standpoint

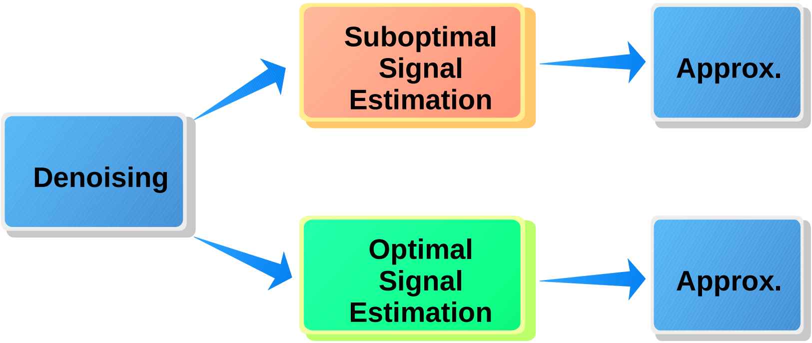
Noise Removal – Bayesian Standpoint



Noise Removal – Bayesian Standpoint



Noise Removal – Bayesian Standpoint

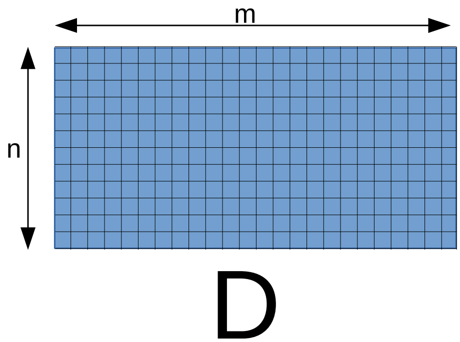


Outline

Outline

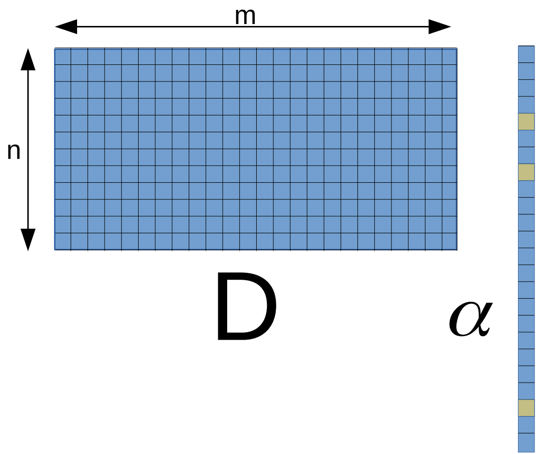
The Generative Model

$\mathbf{D} \in \mathbb{R}^{n \times m}$ is a dictionary with normalized columns.



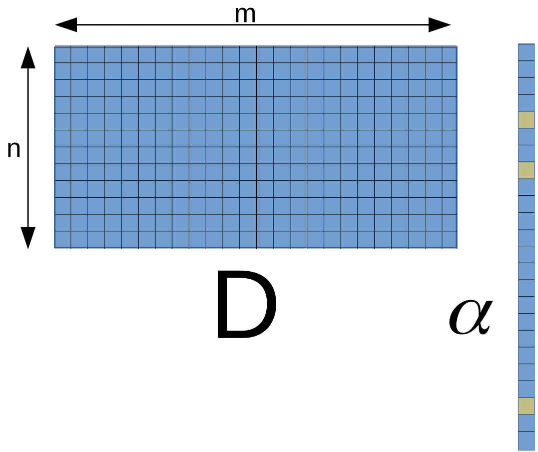
The Generative Model

Each element i in $\boldsymbol{\alpha}$ is non zero with probability $p_i \ll 1$.

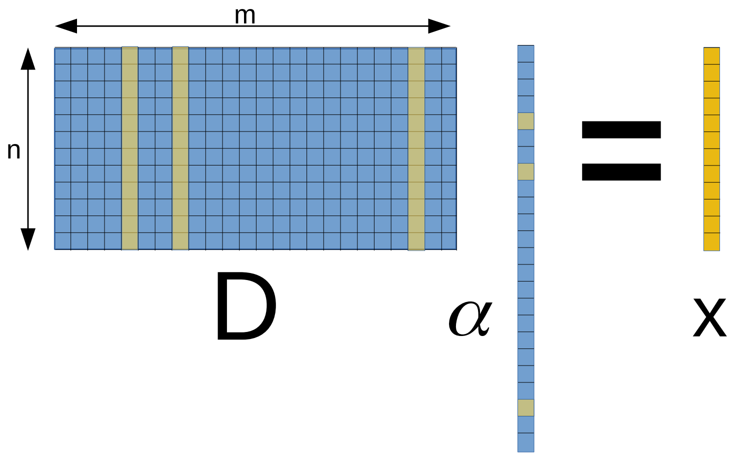


The Generative Model

The non-zero elements of the sparse representation, denoted by $\boldsymbol{\alpha}_s$, are sampled from a Gaussian distribution $\boldsymbol{\alpha}_s | s \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\boldsymbol{\alpha}}^2 \mathbf{I}_{|s|}\right)$.

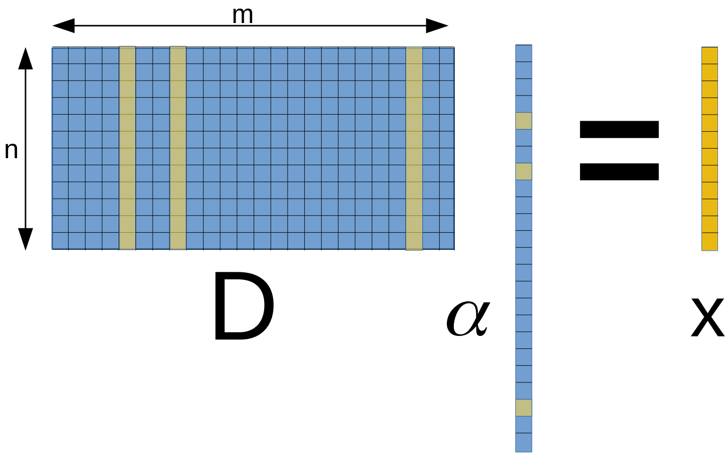


The product $\mathbf{D}\alpha$ leads to a signal $\mathbf{x}$.



The Generative Model

We are given noisy measurements $\mathbf{y} = \mathbf{D}\boldsymbol{\alpha} + \boldsymbol{\nu}$, where $\boldsymbol{\nu}$ is a white Gaussian noise $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\boldsymbol{\nu}}^2 \mathbf{I}_n\right)$.



The Generative Model – Results

[1] Turek, Javier S., Irad Yavneh, and Michael Elad, 2011. "On MMSE and MAP denoising under sparse representation modeling over a unitary dictionary."

The Generative Model – Results

- The prior probability of a support (Bernoulli):
  $p(s) = \prod_{i \in s} p_i \prod_{j \notin s} (1 - p_j)$.

[1] Turek, Javier S., Irad Yavneh, and Michael Elad, 2011. "On MMSE and MAP denoising under sparse representation modeling over a unitary dictionary."

The Generative Model – Results

- The prior probability of a support (Bernoulli): $p(s) = \prod_{i \in s} p_i \prod_{j \notin s} (1 - p_j)$.
- When the support is known, $\mathbf{y}$ and $\boldsymbol{\alpha}_s$ are jointly Gaussian $\mathbf{y} = \mathbf{D}_s \boldsymbol{\alpha}_s + \boldsymbol{\nu}$, leading to[1]

[1] Turek, Javier S., Irad Yavneh, and Michael Elad, 2011. "On MMSE and MAP denoising under sparse representation modeling over a unitary dictionary."

The Generative Model – Results

- The prior probability of a support (Bernoulli): $p(s) = \prod_{i \in s} p_i \prod_{j \notin s} (1 - p_j)$.
- When the support is known, $\mathbf{y}$ and $\boldsymbol{\alpha}_s$ are jointly Gaussian $\mathbf{y} = \mathbf{D}_s \boldsymbol{\alpha}_s + \boldsymbol{\nu}$, leading to[1]
  - $\mathbf{y}|s$ is Gaussian: $\mathbf{y}|s \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_s)$.

[1] Turek, Javier S., Irad Yavneh, and Michael Elad, 2011. "On MMSE and MAP denoising under sparse representation modeling over a unitary dictionary."

The Generative Model – Results

- The prior probability of a support (Bernoulli): $p(s) = \prod_{i \in s} p_i \prod_{j \notin s} (1 - p_j)$.
- When the support is known, $\mathbf{y}$ and $\boldsymbol{\alpha}_s$ are jointly Gaussian $\mathbf{y} = \mathbf{D}_s \boldsymbol{\alpha}_s + \boldsymbol{\nu}$, leading to[1]
  - $\mathbf{y}|s$ is Gaussian: $\mathbf{y}|s \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_s)$.
  - $\mathbf{y}|\boldsymbol{\alpha}_s, s$ is Gaussian: $\mathbf{y}|\boldsymbol{\alpha}_s, s \sim \mathcal{N}\left(\mathbf{D}_s \boldsymbol{\alpha}_s, \sigma_{\boldsymbol{\nu}}^2 \mathbf{I}_n\right)$.

[1] Turek, Javier S., Irad Yavneh, and Michael Elad, 2011. "On MMSE and MAP denoising under sparse representation modeling over a unitary dictionary."

The Generative Model – Results

- The prior probability of a support (Bernoulli): $p(s) = \prod_{i \in s} p_i \prod_{j \notin s} (1 - p_j)$.
- When the support is known, $\mathbf{y}$ and $\boldsymbol{\alpha}_s$ are jointly Gaussian $\mathbf{y} = \mathbf{D}_s \boldsymbol{\alpha}_s + \boldsymbol{\nu}$, leading to[1]
  - $\mathbf{y}|s$ is Gaussian: $\mathbf{y}|s \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_s)$.
  - $\mathbf{y}|\boldsymbol{\alpha}_s, s$ is Gaussian: $\mathbf{y}|\boldsymbol{\alpha}_s, s \sim \mathcal{N}\left(\mathbf{D}_s \boldsymbol{\alpha}_s, \sigma_{\boldsymbol{\nu}}^2 \mathbf{I}_n\right)$.
  - $\boldsymbol{\alpha}_s|\mathbf{y}, s$ is Gaussian: $\boldsymbol{\alpha}_s|\mathbf{y}, s \sim \mathcal{N}\left(\frac{1}{\sigma_\nu^2} \mathbf{Q}_s^{-1} \mathbf{D}_s^T \mathbf{y}, \mathbf{Q}_s^{-1}\right)$.

$$\mathbf{C}_s = \sigma_\alpha^2 \mathbf{D}_s \mathbf{D}_s^T + \sigma_\nu^2 \mathbf{I}_n, \quad \mathbf{Q}_s = \frac{1}{\sigma_\alpha^2} \mathbf{I}_{|s|} + \frac{1}{\sigma_\nu} \mathbf{D}_s^T \mathbf{D}_s$$

[1] Turek, Javier S., Irad Yavneh, and Michael Elad, 2011. "On MMSE and MAP denoising under sparse representation modeling over a unitary dictionary."

Outline

Bayesian Estimators

The goal: Estimate α given the noisy measurements $\mathbf{y}$,
i.e. denoise the signal.

Bayesian Estimators

The goal: Estimate α given the noisy measurements $\mathbf{y}$, i.e. denoise the signal.

Many estimators can be proposed. We focus our attention on three:

Bayesian Estimators

The goal: Estimate α given the noisy measurements $\mathbf{y}$,
i.e. denoise the signal.

Many estimators can be proposed. We focus our attention on three:

1. The oracle estimator.

Bayesian Estimators

The goal: Estimate α given the noisy measurements $\mathbf{y}$,
i.e. denoise the signal.

Many estimators can be proposed. We focus our attention on three:

1. The oracle estimator.
2. The Maximum A-posteriori Probability (MAP) estimator.

Bayesian Estimators

The goal: Estimate α given the noisy measurements $\mathbf{y}$,
i.e. denoise the signal.

Many estimators can be proposed. We focus our attention on three:

1. The oracle estimator.
2. The Maximum A-posteriori Probability (MAP) estimator.
3. The Minimum Mean Square Error (MMSE) estimator.

The Oracle Estimator

The Oracle Estimator

- Assumes knowledge of the support s.

The Oracle Estimator

- Assumes knowledge of the support s.
- $\alpha_s | \mathbf{y}, s$ is Gaussian $\implies$ it's MAP and MMSE estimators are identical.

The Oracle Estimator

- Assumes knowledge of the support s.
- $\boldsymbol{\alpha}_s | \mathbf{y}, s$ is Gaussian $\Longrightarrow$ it's MAP and MMSE estimators are identical.

Oracle Estimator

$$\widehat{\boldsymbol{\alpha}}_s^{\text{Oracle}} = \mathbb{E}\left\{\boldsymbol{\alpha}_s | s, \mathbf{y}\right\} = \frac{1}{\sigma_\nu^2}\mathbf{Q}_s^{-1}\mathbf{D}_s^T\mathbf{y}$$

The Oracle Estimator

- Assumes knowledge of the support s.
- $\boldsymbol{\alpha}_s | \mathbf{y}, s$ is Gaussian $\Longrightarrow$ it's MAP and MMSE estimators are identical.

Oracle Estimator

$$\widehat{\boldsymbol{\alpha}}_s^{\text{Oracle}} = \mathbb{E}\left\{\boldsymbol{\alpha}_s | s, \mathbf{y}\right\} = \frac{1}{\sigma_\nu^2} \mathbf{Q}_s^{-1} \mathbf{D}_s^T \mathbf{y}$$

- Cannot be obtained in practice.

The Oracle Estimator

- Assumes knowledge of the support s.
- $\boldsymbol{\alpha}_s | \mathbf{y}, s$ is Gaussian $\Longrightarrow$ it's MAP and MMSE estimators are identical.

Oracle Estimator

$$\widehat{\boldsymbol{\alpha}}_s^{\text{Oracle}} = \mathbb{E}\left\{\boldsymbol{\alpha}_s | s, \mathbf{y}\right\} = \frac{1}{\sigma_\nu^2} \mathbf{Q}_s^{-1} \mathbf{D}_s^T \mathbf{y}$$

- Cannot be obtained in practice.
- We will use it as a basic building block later in this talk.

The MAP Estimator

The MAP Estimator

Instead of approximating the representation α, we approximate the support itself.

The MAP Estimator

Instead of approximating the representation α, we approximate the support itself.

MAP Support Estimator

$$\widehat{s}_{\text{MAP}} = \arg\max_{s} p\left(s|\mathbf{y}\right)$$

The MAP Estimator

Instead of approximating the representation α, we approximate the support itself.

MAP Support Estimator

$$\widehat{s}_{\text{MAP}} = \arg\max_{s} p\left(s|\mathbf{y}\right) = \arg\max_{s} p\left(s\right) p\left(\mathbf{y}|s\right)$$

The MAP Estimator

Instead of approximating the representation α, we approximate the support itself.

MAP Support Estimator

$$\begin{aligned}\widehat{s}_{\text{MAP}} &= \arg\max_{s} p\left(s|\mathbf{y}\right) = \arg\max_{s} p\left(s\right) p\left(\mathbf{y}|s\right) \\ &= \arg\max_{s} -\frac{1}{2}\mathbf{y}^T \mathbf{C}_s^{-1}\mathbf{y} - \frac{1}{2}\log\det\left(\mathbf{C}_s\right) \\ &\quad + \sum_{i\in s}\log\left(p_i\right) + \sum_{j\notin s}\log\left(1-p_j\right)\end{aligned}$$

The MAP Estimator

Instead of approximating the representation α, we approximate the support itself.

MAP Support Estimator

$$
\begin{aligned}
\widehat{s}_{\mathrm{MAP}} &= \arg\max_{s} p\left(s|\mathbf{y}\right) = \arg\max_{s} p\left(s\right) p\left(\mathbf{y}|s\right) \\
&= \arg\max_{s} -\frac{1}{2}\mathbf{y}^T\mathbf{C}_s^{-1}\mathbf{y} - \frac{1}{2}\log\det\left(\mathbf{C}_s\right) \\
&\qquad + \sum_{i\in s}\log\left(p_i\right) + \sum_{j\notin s}\log\left(1-p_j\right)
\end{aligned}
$$

$$s \in \{0,1\}^m$$

The MAP Estimator

Instead of approximating the representation $\boldsymbol{\alpha}$, we approximate the support itself.

MAP Support Estimator

$$
\begin{aligned}
\widehat{s}_{\text{MAP}} &= \arg\max_{s} p\left(s|\mathbf{y}\right) = \arg\max_{s} p\left(s\right) p\left(\mathbf{y}|s\right) \\
&= \arg\max_{s} -\frac{1}{2}\mathbf{y}^T \mathbf{C}_s^{-1}\mathbf{y} - \frac{1}{2}\log\det\left(\mathbf{C}_s\right) \\
&\qquad + \sum_{i\in s}\log\left(p_i\right) + \sum_{j\notin s}\log\left(1-p_j\right)
\end{aligned}
$$

$$s \in \{0,1\}^m$$

The sparse representation $\widehat{\boldsymbol{\alpha}}_{\text{MAP}}$, is obtained using the oracle estimator on the recovered support: $\widehat{\boldsymbol{\alpha}}_{\text{MAP}} = \widehat{\boldsymbol{\alpha}}^{\text{Oracle}}_{s_{\text{MAP}}}$.

The MMSE Estimator

The Minimum Mean Square Error (MMSE) estimator minimizes the MSE of the estimator.

The MMSE Estimator

The Minimum Mean Square Error (MMSE) estimator minimizes the MSE of the estimator.

MMSE Estimator

$$\widehat{\boldsymbol{\alpha}}_{\text{MMSE}} = \arg\min_{\widehat{\boldsymbol{\alpha}}(\mathbf{y})} \mathbb{E}\left\{ \|\widehat{\boldsymbol{\alpha}}(\mathbf{y}) - \boldsymbol{\alpha}\|_2^2 \middle| \mathbf{y} \right\}$$

The MMSE Estimator

The Minimum Mean Square Error (MMSE) estimator minimizes the MSE of the estimator.

MMSE Estimator

$$\widehat{\boldsymbol{\alpha}}_{\mathsf{MMSE}} = \underset{\widehat{\boldsymbol{\alpha}}(\mathbf{y})}{\arg\min}\, \mathbb{E}\left\{ \left\| \widehat{\boldsymbol{\alpha}}\left(\mathbf{y}\right) - \boldsymbol{\alpha} \right\|_2^2 \middle| \mathbf{y} \right\} = \mathbb{E}\left\{ \boldsymbol{\alpha} | \mathbf{y} \right\}$$

The MMSE Estimator

The Minimum Mean Square Error (MMSE) estimator minimizes the MSE of the estimator.

MMSE Estimator

$$\begin{aligned}\widehat{\boldsymbol{\alpha}}_{\text{MMSE}} &= \underset{\widehat{\boldsymbol{\alpha}}(\mathbf{y})}{\arg\min}\, \mathbb{E}\left\{ \left\| \widehat{\boldsymbol{\alpha}}\left(\mathbf{y}\right) - \boldsymbol{\alpha} \right\|_2^2 \middle| \mathbf{y} \right\} = \mathbb{E}\left\{ \boldsymbol{\alpha} | \mathbf{y} \right\} \\ &= \mathbb{E}_{s|\mathbf{y}}\left\{ \mathbb{E}_{\boldsymbol{\alpha}|\mathbf{y},s}\left\{ \boldsymbol{\alpha} | \mathbf{y}, s \right\} \right\}\end{aligned}$$

The MMSE Estimator

The Minimum Mean Square Error (MMSE) estimator minimizes the MSE of the estimator.

MMSE Estimator

$$\begin{aligned}\widehat{\boldsymbol{\alpha}}_{\text{MMSE}} &= \underset{\widehat{\boldsymbol{\alpha}}(\mathbf{y})}{\arg\min}\, \mathbb{E}\left\{ \left\| \widehat{\boldsymbol{\alpha}}(\mathbf{y}) - \boldsymbol{\alpha} \right\|_2^2 \middle| \mathbf{y} \right\} = \mathbb{E}\left\{ \boldsymbol{\alpha} | \mathbf{y} \right\} \\ &= \mathbb{E}_{s|\mathbf{y}}\left\{ \mathbb{E}_{\boldsymbol{\alpha}|\mathbf{y},s}\left\{ \boldsymbol{\alpha} | \mathbf{y}, s \right\} \right\} = \sum_{s\in\{0,1\}^m} p(s|\mathbf{y})\, \widehat{\boldsymbol{\alpha}}_s^{\text{Oracle}}\end{aligned}$$

The MMSE Estimator

The Minimum Mean Square Error (MMSE) estimator minimizes the MSE of the estimator.

MMSE Estimator

$$\widehat{\boldsymbol{\alpha}}_{\text{MMSE}} = \underset{\widehat{\boldsymbol{\alpha}}(\mathbf{y})}{\arg\min}\, \mathbb{E}\left\{ \left\| \widehat{\boldsymbol{\alpha}}(\mathbf{y}) - \boldsymbol{\alpha} \right\|_2^2 \middle| \mathbf{y} \right\} = \mathbb{E}\left\{ \boldsymbol{\alpha} | \mathbf{y} \right\}$$

$$= \mathbb{E}_{s|\mathbf{y}}\left\{ \mathbb{E}_{\boldsymbol{\alpha}|\mathbf{y},s}\left\{ \boldsymbol{\alpha} | \mathbf{y}, s \right\} \right\} = \sum_{s \in \{0,1\}^m} p(s|\mathbf{y})\, \widehat{\boldsymbol{\alpha}}_s^{\text{Oracle}}$$

The MMSE is the sum of all the possible oracle estimators, weighted by the probability of the support.

The MMSE Estimator

The Minimum Mean Square Error (MMSE) estimator minimizes the MSE of the estimator.

MMSE Estimator

$$\widehat{\boldsymbol{\alpha}}_{\text{MMSE}} = \underset{\widehat{\boldsymbol{\alpha}}(\mathbf{y})}{\arg\min}\, \mathbb{E}\left\{ \left\| \widehat{\boldsymbol{\alpha}}(\mathbf{y}) - \boldsymbol{\alpha} \right\|_2^2 \middle| \mathbf{y} \right\} = \mathbb{E}\left\{ \boldsymbol{\alpha} | \mathbf{y} \right\}$$
$$= \mathbb{E}_{s|\mathbf{y}}\left\{ \mathbb{E}_{\boldsymbol{\alpha}|\mathbf{y},s}\left\{ \boldsymbol{\alpha} | \mathbf{y}, s \right\} \right\} = \sum_{s \in \{0,1\}^m} p(s|\mathbf{y})\, \widehat{\boldsymbol{\alpha}}_s^{\text{Oracle}}$$

The MMSE is the sum of all the possible oracle estimators, weighted by the probability of the support.

The MMSE estimator is **not sparse** at all!

The MMSE and Map Estimators

- Both estimators are practically impossible to obtain.

The MMSE and Map Estimators

- Both estimators are practically impossible to obtain.
  - MAP – compute the posterior probability of each of the 2^m supports and pick the most probable one.

The MMSE and Map Estimators

- Both estimators are practically impossible to obtain.
  - MAP – compute the posterior probability of each of the 2^m supports and pick the most probable one.
  - MMSE – compute the posterior probability of each support and use them as weights for all the possible oracle estimators.

The MMSE and Map Estimators

- Both estimators are practically impossible to obtain.
  - MAP – compute the posterior probability of each of the 2^m supports and pick the most probable one.
  - MMSE – compute the posterior probability of each support and use them as weights for all the possible oracle estimators.
- How is this issue resolved?

The MMSE and Map Estimators

- Both estimators are practically impossible to obtain.
  - MAP – compute the posterior probability of each of the 2^m supports and pick the most probable one.
  - MMSE – compute the posterior probability of each support and use them as weights for all the possible oracle estimators.
- How is this issue resolved?
  - MAP – use an approximation algorithm (greedy or relaxed) to recover a likely support, and then use the oracle.

The MMSE and Map Estimators

- Both estimators are practically impossible to obtain.
  - MAP – compute the posterior probability of each of the 2^m supports and pick the most probable one.
  - MMSE – compute the posterior probability of each support and use them as weights for all the possible oracle estimators.
- How is this issue resolved?
  - MAP – use an approximation algorithm (greedy or relaxed) to recover a likely support, and then use the oracle.
  - MMSE – usually avoided.

The MMSE and Map Estimators

- Both estimators are practically impossible to obtain.
  - MAP – compute the posterior probability of each of the 2^m supports and pick the most probable one.
  - MMSE – compute the posterior probability of each support and use them as weights for all the possible oracle estimators.
- How is this issue resolved?
  - MAP – use an approximation algorithm (greedy or relaxed) to recover a likely support, and then use the oracle.
  - MMSE – usually avoided.
- Can we do better?

Outline

Outline

Random OMP

Elad, Michael, and Irad Yavneh, 2009. "A plurality of sparse representations is better than the sparsest one alone."

Random OMP

- A modified version of the OMP algorithm.

Elad, Michael, and Irad Yavneh, 2009. "A plurality of sparse representations is better than the sparsest one alone."

Random OMP

- A modified version of the OMP algorithm.
  - OMP: picks the atom most correlated with the current residual.

Elad, Michael, and Irad Yavneh, 2009. "A plurality of sparse representations is better than the sparsest one alone."

Random OMP

- A modified version of the OMP algorithm.
  - OMP: picks the atom most correlated with the current residual.
  - RandOMP: weights the unpicked atoms according to their correlation with the residual, and chooses randomly.

Elad, Michael, and Irad Yavneh, 2009. "A plurality of sparse representations is better than the sparsest one alone."

Random OMP

- A modified version of the OMP algorithm.
  - OMP: picks the atom most correlated with the current residual.
  - RandOMP: weights the unpicked atoms according to their correlation with the residual, and chooses randomly.
- Repeated many times leading to a variety of solutions.

Elad, Michael, and Irad Yavneh, 2009. "A plurality of sparse representations is better than the sparsest one alone."

Random OMP

- A modified version of the OMP algorithm.
  - OMP: picks the atom most correlated with the current residual.
  - RandOMP: weights the unpicked atoms according to their correlation with the residual, and chooses randomly.
- Repeated many times leading to a variety of solutions.
- Averages the solutions to retrieve a final estimate.

Elad, Michael, and Irad Yavneh, 2009. "A plurality of sparse representations is better than the sparsest one alone."

Random OMP

- A modified version of the OMP algorithm.
  - OMP: picks the atom most correlated with the current residual.
  - RandOMP: weights the unpicked atoms according to their correlation with the residual, and chooses randomly.
- Repeated many times leading to a variety of solutions.
- Averages the solutions to retrieve a final estimate.
- Asymptotically converges with the MMSE estimator when:
  - The dictionary is unitary.
  - The cardinality of the sparse representation is 1.

Elad, Michael, and Irad Yavneh, 2009. "A plurality of sparse representations is better than the sparsest one alone."

Random OMP

- A modified version of the OMP algorithm.
  - OMP: picks the atom most correlated with the current residual.
  - RandOMP: weights the unpicked atoms according to their correlation with the residual, and chooses randomly.
- Repeated many times leading to a variety of solutions.
- Averages the solutions to retrieve a final estimate.
- Asymptotically converges with the MMSE estimator when:
  - The dictionary is unitary.
  - The cardinality of the sparse representation is 1.
- Empirically achieves better MSE than OMP even when these conditions are not met.

Elad, Michael, and Irad Yavneh, 2009. "A plurality of sparse representations is better than the sparsest one alone."

Random OMP

- A modified version of the OMP algorithm.
  - OMP: picks the atom most correlated with the current residual.
  - RandOMP: weights the unpicked atoms according to their correlation with the residual, and chooses randomly.
- Repeated many times leading to a variety of solutions.
- Averages the solutions to retrieve a final estimate.
- Asymptotically converges with the MMSE estimator when:
  - The dictionary is unitary.
  - The cardinality of the sparse representation is 1.
- Empirically achieves better MSE than OMP even when these conditions are not met.

Other methods exist (Schniter, P. et al. 2008 "Fast Bayesian matching pursuit")

Elad, Michael, and Irad Yavneh, 2009. "A plurality of sparse representations is better than the sparsest one alone."

MMSE Approximation Methods

Why do these methods work?

MMSE Approximation Methods

Why do these methods work?

- $p(s|\mathbf{y})$ has an exponential nature.

MMSE Approximation Methods

Why do these methods work?

- $p(s|\mathbf{y})$ has an exponential nature. $\Rightarrow$ The estimator is dominated by a small set of solutions.

MMSE Approximation Methods

Why do these methods work?

- $p(s|\mathbf{y})$ has an exponential nature. $\Rightarrow$ The estimator is dominated by a small set of solutions.

$$\widehat{\boldsymbol{\alpha}}_{\text{MMSE}} = \sum_{s\in\{0,1\}^m} p\left(s|\mathbf{y}\right)\widehat{\boldsymbol{\alpha}}_s^{\text{Oracle}} \approx \sum_{s\in\omega\subset\{0,1\}^m} p\left(s|\mathbf{y}\right)\widehat{\boldsymbol{\alpha}}_s^{\text{Oracle}}$$

MMSE Approximation Methods

Why do these methods work?

- $p(s|\mathbf{y})$ has an exponential nature. $\Rightarrow$ The estimator is dominated by a small set of solutions.

$$\widehat{\boldsymbol{\alpha}}_{\text{MMSE}} = \sum_{s\in\{0,1\}^m} p\left(s|\mathbf{y}\right)\widehat{\boldsymbol{\alpha}}_s^{\text{Oracle}} \approx \sum_{s\in\omega\subset\{0,1\}^m} p\left(s|\mathbf{y}\right)\widehat{\boldsymbol{\alpha}}_s^{\text{Oracle}}$$

- These methods find a “dominant” subset ω of supports and approximate their weights (posterior probabilities).

MMSE Approximation Methods

Why do these methods work?

- $p(s|\mathbf{y})$ has an exponential nature. $\Rightarrow$ The estimator is dominated by a small set of solutions.

$$\widehat{\alpha}_{\text{MMSE}} = \sum_{s \in \{0,1\}^m} p\left(s|\mathbf{y}\right) \widehat{\alpha}_s^{\text{Oracle}} \approx \sum_{s \in \omega \subset \{0,1\}^m} p\left(s|\mathbf{y}\right) \widehat{\alpha}_s^{\text{Oracle}}$$

- These methods find a "dominant" subset ω of supports and approximate their weights (posterior probabilities).

These methods operate in a greedy fashion.

MMSE Approximation Methods

Why do these methods work?

- $p(s|\mathbf{y})$ has an exponential nature. $\Rightarrow$ The estimator is dominated by a small set of solutions.

$$\widehat{\boldsymbol{\alpha}}_{\text{MMSE}} = \sum_{s\in\{0,1\}^m} p\left(s|\mathbf{y}\right)\widehat{\boldsymbol{\alpha}}_s^{\text{Oracle}} \approx \sum_{s\in\omega\subset\{0,1\}^m} p\left(s|\mathbf{y}\right)\widehat{\boldsymbol{\alpha}}_s^{\text{Oracle}}$$

- These methods find a "dominant" subset ω of supports and approximate their weights (posterior probabilities).

These methods operate in a greedy fashion.

$\Rightarrow$ **They are impractical for high dimensional signals**.

Outline

Outline

Stochastic Resonance

Definition

- Originally, was suggested as an explanation to the periodic recurrence of ice ages.
- Today, broadly applied to describe a more general phenomenon where presence of noise in a *nonlinear* system provides a better response.

Stochastic Resonance

Definition

- Originally, was suggested as an explanation to the periodic recurrence of ice ages.
- Today, broadly applied to describe a more general phenomenon where presence of noise in a *nonlinear* system provides a better response.

Noise **improves** system performance?

Dither – Noise Has a Constructive Value

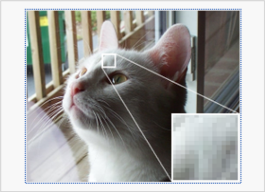
Dither

- In signal quantization, additive noise is used to create stochastic quantization error.
- For example in images, dither prevents color banding which creates unpleasant images.

Dither – Noise Has a Constructive Value

Dither

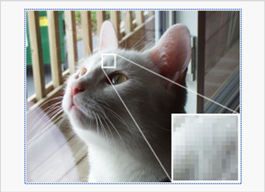
- In signal quantization, additive noise is used to create stochastic quantization error.
- For example in images, dither prevents color banding which creates unpleasant images.



Dither – Noise Has a Constructive Value

Dither

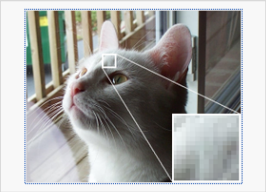
- In signal quantization, additive noise is used to create stochastic quantization error.
- For example in images, dither prevents color banding which creates unpleasant images.



Dither – Noise Has a Constructive Value

Dither

- In signal quantization, additive noise is used to create stochastic quantization error.
- For example in images, dither prevents color banding which creates unpleasant images.



Dither – Noise Has a Constructive Value

Dither

- In signal quantization, additive noise is used to create stochastic quantization error.
- For example in images, dither prevents color banding which creates unpleasant images.

Noise

Outline

Outline

The Proposed Algorithm

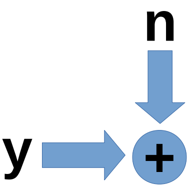
$$\mathbf{y}$$

The Proposed Algorithm

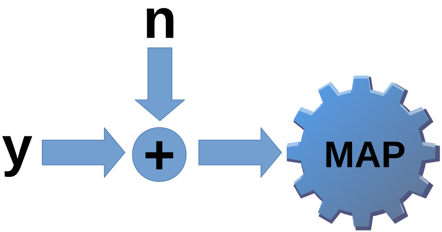
n

y

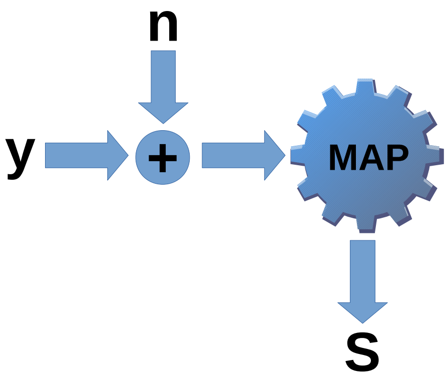
The Proposed Algorithm



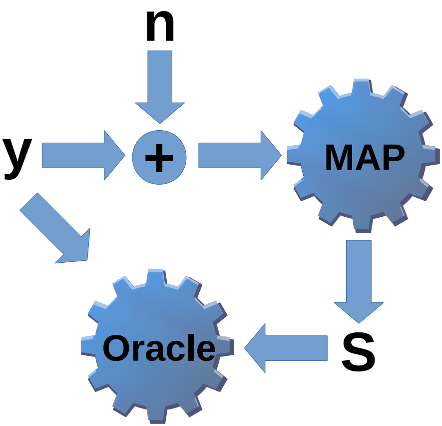
The Proposed Algorithm



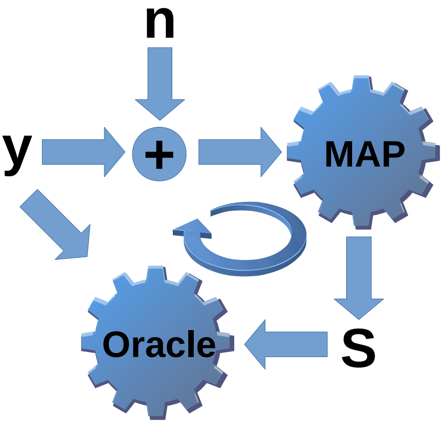
The Proposed Algorithm



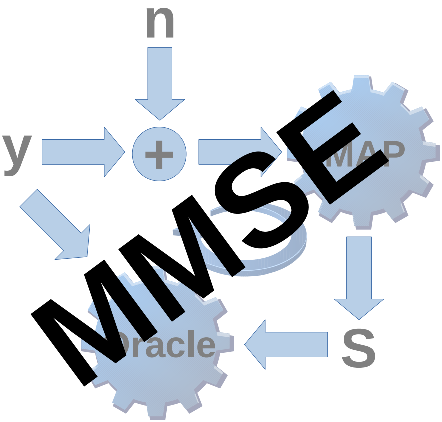
The Proposed Algorithm



The Proposed Algorithm



The Proposed Algorithm



The Proposed Algorithm

Algorithm 2: Stochastic Resonance MMSE Approximation

input : $\mathbf{y}, \mathbf{D}$, `PursuitMethod`, σ_n, K

The Proposed Algorithm

Algorithm 2: Stochastic Resonance MMSE Approximation

input : $\mathbf{y}, \mathbf{D}$, `PursuitMethod`, σ_n, K

output: $\widehat{\alpha}$

The Proposed Algorithm

Algorithm 2: Stochastic Resonance MMSE Approximation

input : $\mathbf{y}, \mathbf{D}$, `PursuitMethod`, σ_n, K

output: $\widehat{\alpha}$

for $k \in 1 \ldots K$ **do**

The Proposed Algorithm

Algorithm 2: Stochastic Resonance MMSE Approximation

input : $\mathbf{y}$, $\mathbf{D}$, `PursuitMethod`, σ_n, K

output: $\widehat{\alpha}$

for $k \in 1 \ldots K$ **do**

  $\mathbf{n}_k \leftarrow \texttt{SampleNoise}(\sigma_n)$

The Proposed Algorithm

Algorithm 2: Stochastic Resonance MMSE Approximation

input : $\mathbf{y}$, $\mathbf{D}$, `PursuitMethod`, σ_n, K

output: $\widehat{\alpha}$

for $k \in 1 \ldots K$ **do**

  $\mathbf{n}_k \leftarrow \texttt{SampleNoise}(\sigma_n)$

  $\tilde{\alpha}_k \leftarrow \texttt{PursuitMethod}(\mathbf{y} + \mathbf{n}_k, \mathbf{D})$

The Proposed Algorithm

Algorithm 2: Stochastic Resonance MMSE Approximation

input : $\mathbf{y}, \mathbf{D}$, `PursuitMethod`, σ_n, K

output: $\widehat{\alpha}$

for $k \in 1...K$ **do**

$\mathbf{n}_k \leftarrow \texttt{SampleNoise}(\sigma_n)$

$\tilde{\alpha}_k \leftarrow \texttt{PursuitMethod}(\mathbf{y} + \mathbf{n}_k, \mathbf{D})$

$\widehat{S}_k \leftarrow \texttt{Support}(\tilde{\alpha}_k)$

The Proposed Algorithm

Algorithm 2: Stochastic Resonance MMSE Approximation

input : $\mathbf{y}, \mathbf{D}$, `PursuitMethod`, σ_n, K

output: $\widehat{\alpha}$

for $k \in 1 \dots K$ **do**

$\mathbf{n}_k \leftarrow \texttt{SampleNoise}(\sigma_n)$

$\tilde{\alpha}_k \leftarrow \texttt{PursuitMethod}(\mathbf{y} + \mathbf{n}_k, \mathbf{D})$

$\widehat{S}_k \leftarrow \texttt{Support}(\tilde{\alpha}_k)$

$\widehat{\alpha}_k \leftarrow \widehat{\alpha}^{\text{Oracle}}_{\widehat{S}_K}(\mathbf{y})$

end

The Proposed Algorithm

Algorithm 2: Stochastic Resonance MMSE Approximation

input : $\mathbf{y}, \mathbf{D}, \texttt{PursuitMethod}, \sigma_n, K$

output: $\widehat{\alpha}$

for $k \in 1...K$ **do**

$\mathbf{n}_k \leftarrow \texttt{SampleNoise}(\sigma_n)$

$\tilde{\alpha}_k \leftarrow \texttt{PursuitMethod}(\mathbf{y} + \mathbf{n}_k, \mathbf{D})$

$\widehat{S}_k \leftarrow \texttt{Support}(\tilde{\alpha}_k)$

$\widehat{\alpha}_k \leftarrow \widehat{\alpha}^{\text{Oracle}}_{\widehat{S}_K}(\mathbf{y})$

end

$\widehat{\alpha} \leftarrow \frac{1}{K} \sum_{k=1}^{K} \widehat{\alpha}_k$

The Proposed Algorithm

Approximation In Large Dimensions

The Proposed Algorithm

Approximation In Large Dimensions

The pursuit itself is independent from the rest of the process
$\Longrightarrow$ Relaxation methods are just as applicable.

The Proposed Algorithm – Does It Work?

- $\mathbf{D} \in \mathbb{R}^{50 \times 100}$ a normalized random dictionary.

The Proposed Algorithm – Does It Work?

- $\mathbf{D} \in \mathbb{R}^{50\times 100}$ a normalized random dictionary.
- $\|\boldsymbol{\alpha}\|_0 = 1, \boldsymbol{\alpha}_s \sim \mathcal{N}(0,1)$.

The Proposed Algorithm – Does It Work?

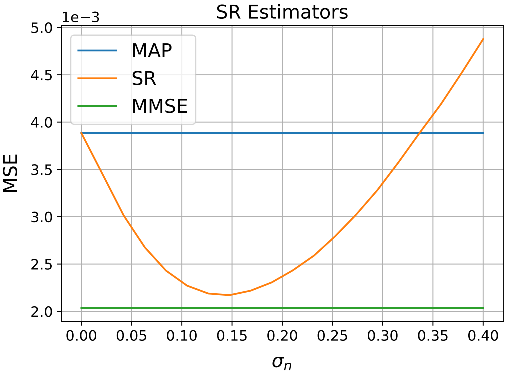
- $\mathbf{D} \in \mathbb{R}^{50 \times 100}$ a normalized random dictionary.
- $\|\boldsymbol{\alpha}\|_0 = 1, \boldsymbol{\alpha}_s \sim \mathcal{N}(0, 1)$.
- $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_\nu^2 \mathbf{I}_{50}\right), \sigma_\nu = 0.2$.

The Proposed Algorithm – Does It Work?

- $\mathbf{D} \in \mathbb{R}^{50\times 100}$ a normalized random dictionary.
- $\|\boldsymbol{\alpha}\|_0 = 1, \boldsymbol{\alpha}_s \sim \mathcal{N}(0,1)$.
- $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_\nu^2 \mathbf{I}_{50}\right), \sigma_\nu = 0.2$.
- 100 iterations of stochastic resonance.

The Proposed Algorithm – Does It Work?

- $\mathbf{D} \in \mathbb{R}^{50\times 100}$ a normalized random dictionary.
- $\|\boldsymbol{\alpha}\|_0 = 1, \boldsymbol{\alpha}_s \sim \mathcal{N}(0,1)$.
- $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_\nu^2 \mathbf{I}_{50}\right), \sigma_\nu = 0.2$.
- 100 iterations of stochastic resonance.



Outline

Unitary Case - Estimators

When $\mathbf{D}$ is a unitary matrix $\left(\mathbf{D}^T\mathbf{D} = \mathbf{I}\right)$, the oracle, MAP and MMSE estimators are element-wise shrinkage operators:

c and λ_{MAP} depend on p_i, σ_α and σ_ν.

Unitary Case - Estimators

When $\mathbf{D}$ is a unitary matrix $\left(\mathbf{D}^T\mathbf{D} = \mathbf{I}\right)$, the oracle, MAP and MMSE estimators are element-wise shrinkage operators:

Oracle

$$\widehat{\boldsymbol{\alpha}}_s^{\text{Oracle}}(\mathbf{y}) = c^2\mathbf{D}_s^T\mathbf{y} = c^2\boldsymbol{\beta}_s$$

---

c and λ_{MAP} depend on p_i, σ_α and σ_ν.

Unitary Case - Estimators

When $\mathbf{D}$ is a unitary matrix $\left(\mathbf{D}^T\mathbf{D} = \mathbf{I}\right)$, the oracle, MAP and MMSE estimators are element-wise shrinkage operators:

MAP

$$\widehat{\boldsymbol{\alpha}}_{\text{MAP}}(\mathbf{y}) = \mathcal{H}_{\lambda_{\text{MAP}}}\left(\mathbf{D}^T\mathbf{y}\right) = \mathcal{H}_{\lambda_{\text{MAP}}}(\boldsymbol{\beta})$$

c and λ_{MAP} depend on p_i, σ_α and σ_ν.

Unitary Case - Estimators

When $\mathbf{D}$ is a unitary matrix $\left(\mathbf{D}^T\mathbf{D} = \mathbf{I}\right)$, the oracle, MAP and MMSE estimators are element-wise shrinkage operators:

MAP

$$\widehat{\boldsymbol{\alpha}}_{\text{MAP}}(\mathbf{y}) = \mathcal{H}_{\lambda_{\text{MAP}}}\left(\mathbf{D}^T\mathbf{y}\right) = \mathcal{H}_{\lambda_{\text{MAP}}}(\boldsymbol{\beta})$$

$$\mathcal{H}_{\lambda_{\text{MAP}}}(\beta_i) = \begin{cases} c^2\beta_i & \text{if } |\beta_i| \geq \lambda_{\text{MAP}}, \\ 0 & \textit{else} \end{cases}$$

c and λ_{MAP} depend on p_i, σ_α and σ_ν.

Unitary Case - Estimators

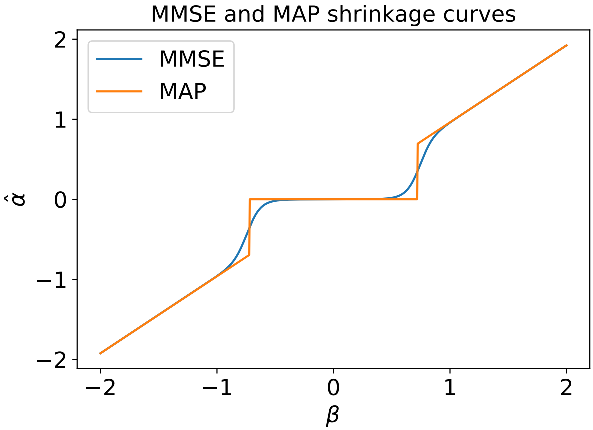
When $\mathbf{D}$ is a unitary matrix $\left(\mathbf{D}^T\mathbf{D} = \mathbf{I}\right)$, the oracle, MAP and MMSE estimators are element-wise shrinkage operators:

MMSE

$$\widehat{\alpha}_i^{MMSE}(\beta_i) = \frac{\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta_i^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}}{1+\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta_i^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}} c^2\beta_i$$

c and λ_{MAP} depend on p_i, σ_α and σ_ν.

Unitary Case – Estimators



Unitary Case – Stochastic Resonance

To use our proposed algorithm we need to provide a pursuit. Algorithm 1

To use our proposed algorithm we need to provide a pursuit. Algorithm 1

The MAP is attainable.

To use our proposed algorithm we need to provide a pursuit. Algorithm 1

The MAP is attainable. $\Longrightarrow$ Use it!

To use our proposed algorithm we need to provide a pursuit. Algorithm 1

The MAP is attainable. $\Longrightarrow$ Use it!

Subtractive hard-thresholding $\mathcal{H}^{-}(\cdot)$

$$\mathcal{H}^{-}(\beta, \tilde{n}) = \begin{cases} & \text{if } |\beta + \tilde{n}| \geq \lambda_{\text{MAP}}, \\ 0 & \text{otherwise.} \end{cases}$$

Unitary Case – Stochastic Resonance

To use our proposed algorithm we need to provide a pursuit. Algorithm 1

The MAP is attainable. $\Longrightarrow$ Use it!

Subtractive hard-thresholding $\mathcal{H}^{-}(\cdot)$

$$\mathcal{H}^{-}(\beta, \tilde{n}) = \begin{cases} c^2\beta & \text{if } |\beta + \tilde{n}| \geq \lambda_{\mathrm{MAP}}, \\ 0 & \text{otherwise.} \end{cases}$$

Unitary Case – Stochastic Resonance

To use our proposed algorithm we need to provide a pursuit. Algorithm 1

The MAP is attainable. $\Longrightarrow$ Use it!

Subtractive hard-thresholding $\mathcal{H}^{-}(\cdot)$

$$\mathcal{H}^{-}(\beta, \tilde{n}) = \begin{cases} c^2\beta & \text{if } |\beta + \tilde{n}| \geq \lambda_{\mathrm{MAP}}, \\ 0 & \text{otherwise.} \end{cases}$$

Unitary Case – Stochastic Resonance

To use our proposed algorithm we need to provide a pursuit. Algorithm 1

The MAP is attainable. $\Longrightarrow$ Use it!

Subtractive hard-thresholding $\mathcal{H}^{-}(\cdot)$

$$\mathcal{H}^{-}(\beta, \tilde{n}) = \begin{cases} c^2\beta & \text{if } |\beta + \tilde{n}| \geq \lambda, \\ 0 & \text{otherwise.} \end{cases}$$

Unitary Case – Asymptotic Estimator

What happens as $K \to \infty$?

Unitary Case – Asymptotic Estimator

What happens as $K \to \infty$?

$$\widehat{\alpha}\left(\beta, \tilde{n}\right) = \mathcal{H}^{-}\left(\beta, \tilde{n}\right) = \begin{cases} c^2\beta & \text{if } |\beta + \tilde{n}| \geq \lambda, \\ 0 & \text{otherwise.} \end{cases}$$

Unitary Case – Asymptotic Estimator

What happens as $K \to \infty$?

$$\widehat{\alpha}\left(\beta, \tilde{n}\right) = \mathcal{H}^{-}\left(\beta, \tilde{n}\right) = \begin{cases} c^2\beta & \text{if } \left|\beta + \tilde{n}\right| \geq \lambda, \\ 0 & \text{otherwise.} \end{cases}$$

$$\widehat{\alpha} = \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \widehat{\alpha}_k = \mathbb{E}_n \left\{\widehat{\alpha}_k\right\}$$

Unitary Case – Asymptotic Estimator

What happens as $K \to \infty$?

$$\widehat{\alpha}\left(\beta, \tilde{n}\right) = \mathcal{H}^{-}\left(\beta, \tilde{n}\right) = \begin{cases} c^2 \beta & \text{if } \left|\beta + \tilde{n}\right| \geq \lambda, \\ 0 & \text{otherwise.} \end{cases}$$

$$\widehat{\alpha} = \lim_{K \to \infty} \frac{1}{K} \sum_{k=1}^{K} \widehat{\alpha}_k = \mathbb{E}_n \left\{ \widehat{\alpha}_k \right\} = \int_{-\infty}^{\infty} \mathcal{H}^{-}\left(\beta, \tilde{n}\right) p\left(\tilde{n}\right) d\tilde{n}$$

Unitary Case – Asymptotic Estimator

What happens as $K \to \infty$?

$$\widehat{\alpha}\left(\beta, \tilde{n}\right) = \mathcal{H}^{-}\left(\beta, \tilde{n}\right) = \begin{cases} c^2\beta & \text{if } |\beta + \tilde{n}| \geq \lambda, \\ 0 & \text{otherwise.} \end{cases}$$

$$\widehat{\alpha} = \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \widehat{\alpha}_k = \mathbb{E}_n\left\{\widehat{\alpha}_k\right\} = \int_{-\infty}^{\infty} \mathcal{H}^{-}\left(\beta, \tilde{n}\right) p\left(\tilde{n}\right) d\tilde{n}$$

$$= \ldots$$

Unitary Case – Asymptotic Estimator

What happens as $K \to \infty$?

$$\widehat{\alpha}\left(\beta, \tilde{n}\right) = \mathcal{H}^{-}\left(\beta, \tilde{n}\right) = \begin{cases} c^2\beta & \text{if } \left|\beta + \tilde{n}\right| \geq \lambda, \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{aligned} \widehat{\alpha} = \lim_{K \to \infty} \frac{1}{K} \sum_{k=1}^{K} \widehat{\alpha}_k = \mathbb{E}_n\left\{\widehat{\alpha}_k\right\} &= \int_{-\infty}^{\infty} \mathcal{H}^{-}\left(\beta, \tilde{n}\right) p\left(\tilde{n}\right) d\tilde{n} \\ &= \ldots \\ &= c^2\beta\left[Q\left(\frac{\lambda + \beta}{\sigma_n}\right) + Q\left(\frac{\lambda - \beta}{\sigma_n}\right)\right] \end{aligned}$$

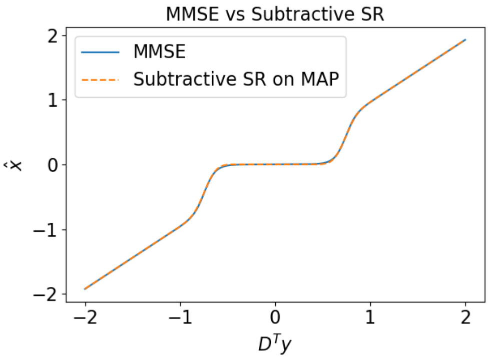
$Q(x) = \int_x^{\infty} e^{-\frac{t^2}{2}} dt$

Unitary Case – Empirical Performance

How does this estimator perform?

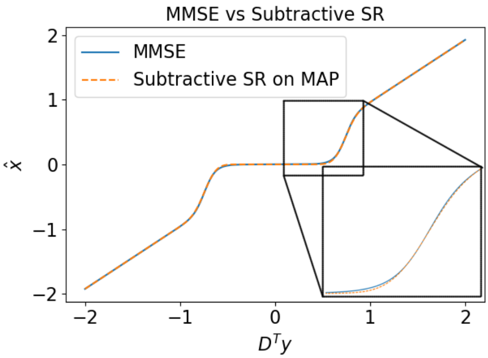
Unitary Case – Empirical Performance

How does this estimator perform?



Unitary Case – Empirical Performance

How does this estimator perform?



Stochastic Resonance vs. MMSE

Are the two the same?

Stochastic Resonance & MMSE

$$\hat{\alpha}_{\text{stochastic}} = c^2\beta\left[Q\left(\frac{\lambda+\beta}{\sigma_n}\right) + Q\left(\frac{\lambda-\beta}{\sigma_n}\right)\right]$$

$$\hat{\alpha}_{MMSE} = \frac{\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}}{1+\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}}c^2\beta$$

Stochastic Resonance vs. MMSE

Are the two the same?

Stochastic Resonance & MMSE

$$\hat{\alpha}_{\text{stochastic}} = c^2\beta\left[Q\left(\frac{\lambda+\beta}{\sigma_n}\right)+Q\left(\frac{\lambda-\beta}{\sigma_n}\right)\right]$$

$$\hat{\alpha}_{MMSE} = \frac{\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}}{1+\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}}c^2\beta$$

No... But are they close?

Stochastic Resonance vs. MMSE

Are the two the same?

Stochastic Resonance & MMSE

$$\hat{\alpha}_{\text{stochastic}} = c^2\beta\left[Q\left(\frac{\lambda+\beta}{\sigma_n}\right) + Q\left(\frac{\lambda-\beta}{\sigma_n}\right)\right]$$

$$\hat{\alpha}_{MMSE} = \frac{\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}}{1+\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}}c^2\beta$$

No... But are they close?

Empirically yes, but for the right choice of parameters.

Stochastic Resonance vs. MMSE

Are the two the same?

Stochastic Resonance & MMSE

$$\hat{\alpha}_{\text{stochastic}} = c^2\beta\left[Q\left(\frac{\lambda+\beta}{\sigma_n}\right) + Q\left(\frac{\lambda-\beta}{\sigma_n}\right)\right]$$

$$\hat{\alpha}_{MMSE} = \frac{\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}}{1+\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}}c^2\beta$$

No... But are they close?

Empirically yes, but for the right choice of parameters.

We can set the parameters by using SURE.

Stochastic Resonance vs. MMSE

Are the two the same?

Stochastic Resonance & MMSE

$$\hat{\alpha}_{\text{stochastic}} = c^2\beta\left[Q\left(\frac{\lambda+\beta}{\sigma_n}\right)+Q\left(\frac{\lambda-\beta}{\sigma_n}\right)\right]$$

$$\hat{\alpha}_{MMSE} = \frac{\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}}{1+\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}}c^2\beta$$

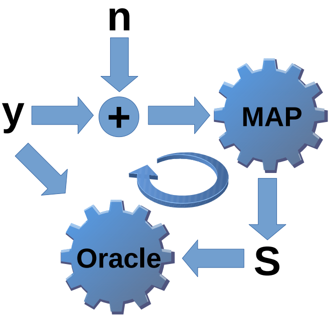
No... But are they close?

Empirically yes, but for the right choice of parameters.

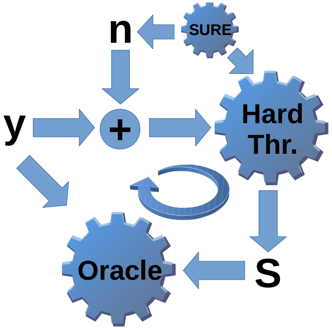
We can set the parameters by using SURE.

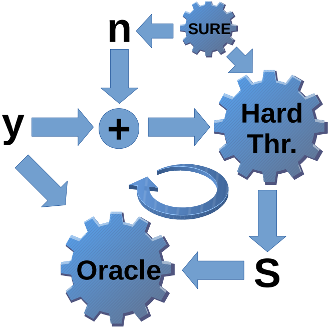
More information in our paper.

Unitary Case – Summary



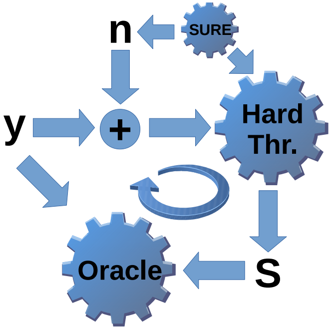
Unitary Case – Summary





Applicable when the closed form solution of the MMSE is not attainable (i.e. when p_i is not known).

Unitary Case – Summary



Applicable when the closed form solution of the MMSE is not attainable (i.e. when p_i is not known).

What about non-unitary cases?

Outline

The General Dictionary Case

The General Dictionary Case

- The general case is harder to analyze.

The General Dictionary Case

- The general case is harder to analyze.
- The MAP estimator is exhaustive $\Longrightarrow$ use a pursuit algorithm instead.

The General Dictionary Case

- The general case is harder to analyze.
- The MAP estimator is exhaustive $\Longrightarrow$ use a pursuit algorithm instead.
- The performance (and the analysis) depends on the pursuit used.

The General Dictionary Case

- The general case is harder to analyze.
- The MAP estimator is exhaustive $\Longrightarrow$ use a pursuit algorithm instead.
- The performance (and the analysis) depends on the pursuit used.
- We separate to two cases:

The General Dictionary Case

- The general case is harder to analyze.
- The MAP estimator is exhaustive $\Longrightarrow$ use a pursuit algorithm instead.
- The performance (and the analysis) depends on the pursuit used.
- We separate to two cases:
  - The generative model's parameters are known.

The General Dictionary Case

- The general case is harder to analyze.
- The MAP estimator is exhaustive $\Longrightarrow$ use a pursuit algorithm instead.
- The performance (and the analysis) depends on the pursuit used.
- We separate to two cases:
  - The generative model's parameters are known.
  - The generative model's parameters are not known.

General Case – Known Parameters

General Case – Known Parameters

- The parameters are known $\Longrightarrow$ use them.

General Case – Known Parameters

- The parameters are known $\Longrightarrow$ use them.
- Replace the arithmetic mean in the proposed algorithm with a weighted sum.

General Case – Known Parameters

- The parameters are known $\Longrightarrow$ use them.
- Replace the arithmetic mean in the proposed algorithm with a weighted sum.
- Similar to FBMP.

General Case – Known Parameters

- The parameters are known $\Longrightarrow$ use them.
- Replace the arithmetic mean in the proposed algorithm with a weighted sum.
- Similar to FBMP.
- But unlike FBMP - no greedy measures required.

General Case – Known Parameters

- The parameters are known $\Longrightarrow$ use them.
- Replace the arithmetic mean in the proposed algorithm with a weighted sum.
- Similar to FBMP.
- But unlike FBMP - no greedy measures required.

  $\Longrightarrow$ Applicable for large dimensions.

General Case – Known Parameters

- The parameters are known $\Longrightarrow$ use them.
- Replace the arithmetic mean in the proposed algorithm with a weighted sum.
- Similar to FBMP.
- But unlike FBMP - no greedy measures required.
  $\Longrightarrow$ Applicable for large dimensions.
- In our paper we prove it is equivalent to a Monte Carlo importance sampling simulation.

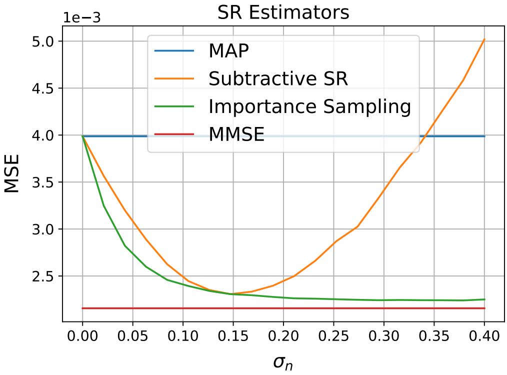
General Case – Known Parameters

- The parameters are known $\Longrightarrow$ use them.
- Replace the arithmetic mean in the proposed algorithm with a weighted sum.
- Similar to FBMP.
- But unlike FBMP - no greedy measures required.
  $\Longrightarrow$ Applicable for large dimensions.
- In our paper we prove it is equivalent to a Monte Carlo importance sampling simulation.
  $\Longrightarrow$ Asymptotically converges to the MMSE!

General Case – Known Parameters

- $\mathbf{D} \in \mathbb{R}^{50\times 100}$ a normalized random dictionary.
- $\|\boldsymbol{\alpha}\|_0 = 1, \boldsymbol{\alpha}_s \sim \mathcal{N}(0, 1)$.
- $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_\nu^2 \mathbf{I}_{50}\right), \sigma_\nu = 0.2$.
- 35 iterations of stochastic resonance.

- $\mathbf{D} \in \mathbb{R}^{50\times100}$ a normalized random dictionary.
- $\|\boldsymbol{\alpha}\|_0 = 1, \boldsymbol{\alpha}_s \sim \mathcal{N}(0,1)$.
- $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_\nu^2 \mathbf{I}_{50}\right), \sigma_\nu = 0.2$.
- 35 iterations of stochastic resonance.



General Case – Unknown Parameters

- Parameters' values are lacking.

General Case – Unknown Parameters

- Parameters' values are lacking. $\implies$ the same MMSE is no longer attainable.

General Case – Unknown Parameters

- Parameters' values are lacking. $\Longrightarrow$ the same MMSE is no longer attainable.
- How can we estimate a support?

- Parameters' values are lacking. $\Longrightarrow$ the same MMSE is no longer attainable.
- How can we estimate a support? Use any pursuit (MAP approximation).

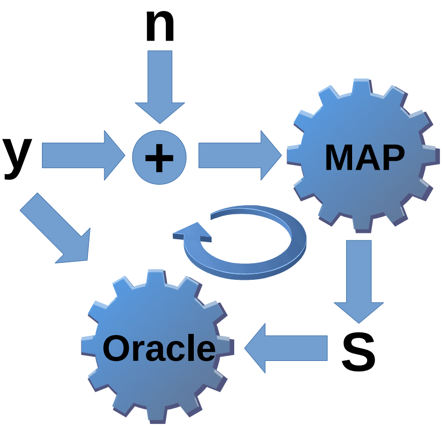
General Case – Unknown Parameters

- Parameters' values are lacking. $\Longrightarrow$ the same MMSE is no longer attainable.
- How can we estimate a support? Use any pursuit (MAP approximation).
- How can we obtain the oracle estimator?

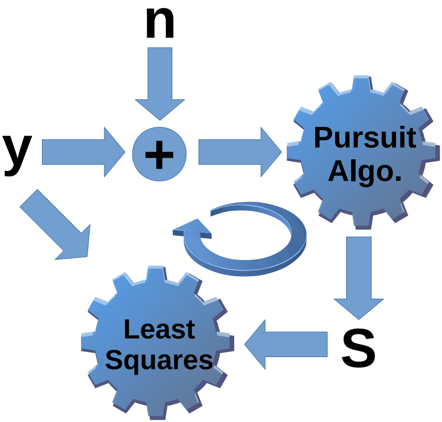
General Case – Unknown Parameters

- Parameters' values are lacking. $\Longrightarrow$ the same MMSE is no longer attainable.
- How can we estimate a support? Use any pursuit (MAP approximation).
- How can we obtain the oracle estimator? Simply use least squares $\widehat{\alpha}\left(s, \mathbf{y}\right)_{\text{oracle}} = \left(\mathbf{D}_s^T \mathbf{D}_s\right)^{-1} \mathbf{D}_s^T \mathbf{y}$.

General Case – Unknown Parameters



General Case – Unknown Parameters

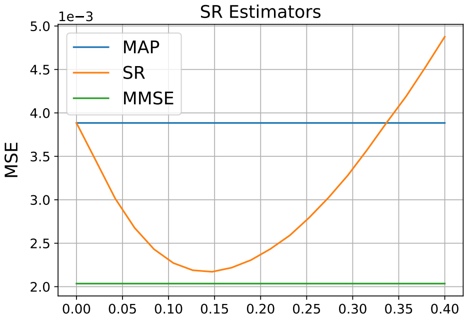


General Case – Unknown Parameters

- $\mathbf{D} \in \mathbb{R}^{50\times 100}$ a normalized random dictionary.
- $\|\boldsymbol{\alpha}\|_0 = 1, \boldsymbol{\alpha}_s \sim \mathcal{N}(0, 1)$.
- $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_\nu^2 \mathbf{I}_{50}\right), \sigma_\nu = 0.2$.
- 100 iterations of stochastic resonance.

General Case – Unknown Parameters

- $\mathbf{D} \in \mathbb{R}^{50\times 100}$ a normalized random dictionary.
- $\|\boldsymbol{\alpha}\|_0 = 1, \boldsymbol{\alpha}_s \sim \mathcal{N}(0,1)$.
- $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_\nu^2 \mathbf{I}_{50}\right), \sigma_\nu = 0.2$.
- 100 iterations of stochastic resonance.



General Case – Unknown Parameters

- $\mathbf{D} \in \mathbb{R}^{50\times 100}$ a normalized random dictionary.
- $\cancel{\|\boldsymbol{\alpha}\|_0 = 1}$ $p_i = 0.05, \boldsymbol{\alpha}_s \sim \mathcal{N}(0, 1)$.
- $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_\nu^2 \mathbf{I}_{50}\right), \sigma_\nu = 0.2$.
- 100 iterations of stochastic resonance.

General Case – Unknown Parameters

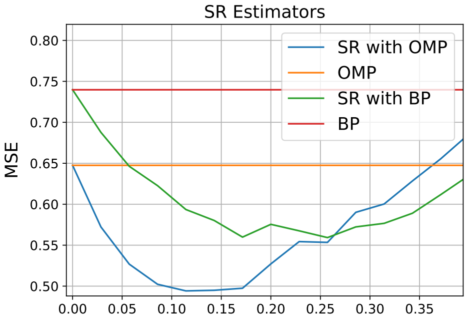
- $\mathbf{D} \in \mathbb{R}^{50\times 100}$ a normalized random dictionary.
- $\cancel{\|\boldsymbol{\alpha}\|_0 = 1}$ $p_i = 0.05, \boldsymbol{\alpha}_s \sim \mathcal{N}(0,1)$.
- $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_\nu^2 \mathbf{I}_{50}\right), \sigma_\nu = 0.2$.
- 100 iterations of stochastic resonance.

Use bounded noise formulation for the pursuit:

$$
\begin{aligned}
&\text{(OMP)} \quad \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}\|_2 \le \epsilon, \\
&\text{(BP)} \quad \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}\|_2 \le \epsilon.
\end{aligned}
$$

General Case – Unknown Parameters

- $\mathbf{D} \in \mathbb{R}^{50\times 100}$ a normalized random dictionary.
- $\cancel{\|\boldsymbol{\alpha}\|_0 = 1}$ $p_i = 0.05, \boldsymbol{\alpha}_s \sim \mathcal{N}(0, 1)$.
- $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_\nu^2 \mathbf{I}_{50}\right), \sigma_\nu = 0.2$.
- 100 iterations of stochastic resonance.



Outline

Image Denoising

Method Used:

[5] Sulam, Jeremias, et al, 2016. "Trainlets: Dictionary learning in high dimensions."
[6] Dai, Wei, and Olgica Milenkovic, 2009. "Subspace pursuit for compressive sensing signal reconstruction."

Image Denoising

Method Used:

- Dataset containing facial images.

[5] Sulam, Jeremias, et al, 2016. "Trainlets: Dictionary learning in high dimensions."
[6] Dai, Wei, and Olgica Milenkovic, 2009. "Subspace pursuit for compressive sensing signal reconstruction."

Method Used:

- Dataset containing facial images.
- Trained a Trainlet[5] dictionary on clean facial images.

[5] Sulam, Jeremias, et al, 2016. "Trainlets: Dictionary learning in high dimensions."
[6] Dai, Wei, and Olgica Milenkovic, 2009. "Subspace pursuit for compressive sensing signal reconstruction."

Image Denoising

Method Used:

- Dataset containing facial images.
- Trained a Trainlet[5] dictionary on clean facial images.
- Pursuit used: Subspace Pursuit (SP)[6].

[5] Sulam, Jeremias, et al, 2016. "Trainlets: Dictionary learning in high dimensions."
[6] Dai, Wei, and Olgica Milenkovic, 2009. "Subspace pursuit for compressive sensing signal reconstruction."

Image Denoising

Method Used:

- Dataset containing facial images.
- Trained a Trainlet[5] dictionary on clean facial images.
- Pursuit used: Subspace Pursuit (SP)[6].

Experiment:

[5] Sulam, Jeremias, et al, 2016. "Trainlets: Dictionary learning in high dimensions."
[6] Dai, Wei, and Olgica Milenkovic, 2009. "Subspace pursuit for compressive sensing signal reconstruction."

Image Denoising

Method Used:

- Dataset containing facial images.
- Trained a Trainlet[5] dictionary on clean facial images.
- Pursuit used: Subspace Pursuit (SP)[6].

Experiment:

- Added noise to an unseen image.

[5] Sulam, Jeremias, et al, 2016. "Trainlets: Dictionary learning in high dimensions."
[6] Dai, Wei, and Olgica Milenkovic, 2009. "Subspace pursuit for compressive sensing signal reconstruction."

Image Denoising

Method Used:

- Dataset containing facial images.
- Trained a Trainlet[5] dictionary on clean facial images.
- Pursuit used: Subspace Pursuit (SP)[6].

Experiment:

- Added noise to an unseen image.
- Use the dictionary and SP to denoise. Tune L parameter to obtain optimal denoising performance.

[5] Sulam, Jeremias, et al, 2016. "Trainlets: Dictionary learning in high dimensions."
[6] Dai, Wei, and Olgica Milenkovic, 2009. "Subspace pursuit for compressive sensing signal reconstruction."

Image Denoising

Method Used:

- Dataset containing facial images.
- Trained a Trainlet[5] dictionary on clean facial images.
- Pursuit used: Subspace Pursuit (SP)[6].

Experiment:

- Added noise to an unseen image.
- Use the dictionary and SP to denoise. Tune L parameter to obtain optimal denoising performance.
- Use SR algorithm using the same SP configuration used in the previous step.

[5] Sulam, Jeremias, et al, 2016. "Trainlets: Dictionary learning in high dimensions."

[6] Dai, Wei, and Olgica Milenkovic, 2009. "Subspace pursuit for compressive sensing signal reconstruction."

Image Denoising – Results

Clean Image.

Image Denoising – Results

Noisy image.
PSNR=16.1 dB.

Clean Image.

Noisy image.
PSNR=16.1 dB.

Subspace Pursuit.
PSNR=26.88 dB.

Clean Image.

Image Denoising – Results

Noisy image.
PSNR=16.1 dB.

Subspace Pursuit.
PSNR=26.88 dB.

Stochastic Resonance.
PSNR=28.76 dB.

Clean Image.

Image Denoising – Results

Noisy image.
PSNR=16.1 dB.

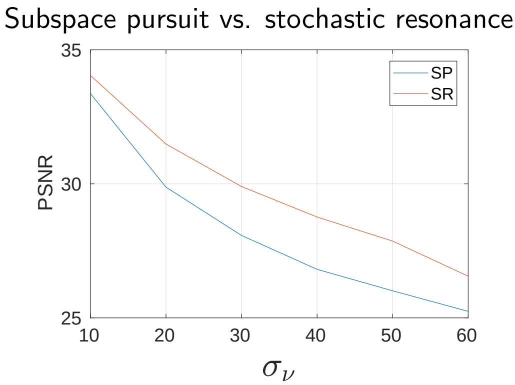
Subspace Pursuit.
PSNR=26.88 dB.

Stochastic Resonance.
PSNR=28.76 dB.

Clean Image.

$\sim$ 2dB better.

Image Denoising – Results



Outline

Conclusions

Conclusions

- MMSE estimator is desired, but involves an exhaustive computation.

Conclusions

- MMSE estimator is desired, but involves an exhaustive computation.
- MAP is hard as well, but many approximation methods are available.

Conclusions

- MMSE estimator is desired, but involves an exhaustive computation.
- MAP is hard as well, but many approximation methods are available.
- We can use synthetic noise and any MAP estimator approximation to achieve an MMSE estimator approximation.

Conclusions

- MMSE estimator is desired, but involves an exhaustive computation.
- MAP is hard as well, but many approximation methods are available.
- We can use synthetic noise and any MAP estimator approximation to achieve an MMSE estimator approximation.
- MMSE estimator approximation is attainable, even for large dimensions.

Thank You