Exploiting Statistical Dependencies in Sparse Representations*

Michael Elad

The Computer Science Department

The Technion – Israel Institute of technology

Haifa 32000, Israel

* Joint work with

Tomer Faktor & Yonina Eldar

Sparse Models, Algorithms and Learning for Large-scale data SMALL

This research was supported by the European Community's FP7-FET program SMALL under grant agreement no. 225913

Agenda

Part I – Motivation for Modeling Dependencies

Part II – Basics of the Boltzmann Machine

Part III – Sparse Recovery Revisited

Part IV – Construct Practical Pursuit Algorithms

Part V – First Steps Towards Adaptive Recovery

Today we will show that

Using BM, sparse and redundant representation modeling can be extended naturally and comfortably to cope with dependencies within the representation vector, while keeping most of the simplicity of the original model

Part I

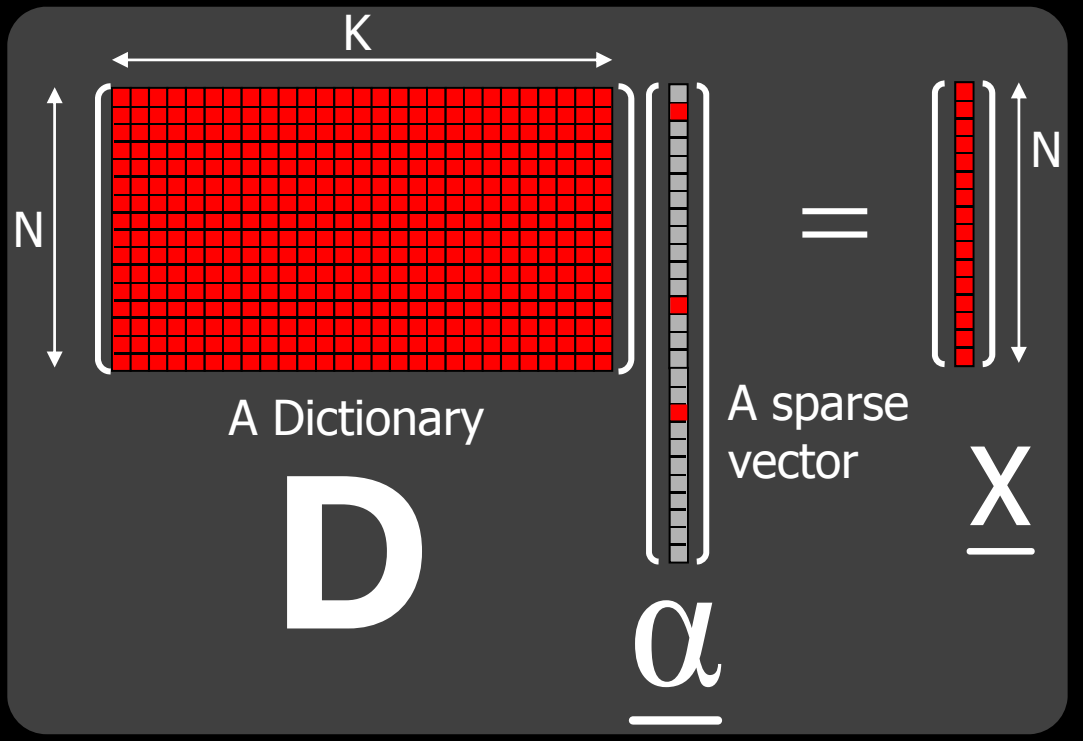
Motivation for Modeling Dependencies

Sparse & Redundant Representations

- We model a signal $\underline{x}$ as a multiplication of a (given) dictionary **D** by a sparse vector $\underline{\alpha}$.
- The classical (Bayesian) approach for modeling $\underline{\alpha}$ is as a Bernoulli-Gaussian RV, implying that the entries of $\underline{\alpha}$ are iid.
- Typical recovery: find $\underline{\alpha}$ from

$$\underline{y} = \underline{x} + \underline{e} = \mathbf{D}\underline{\alpha} + \underline{e} \ .$$

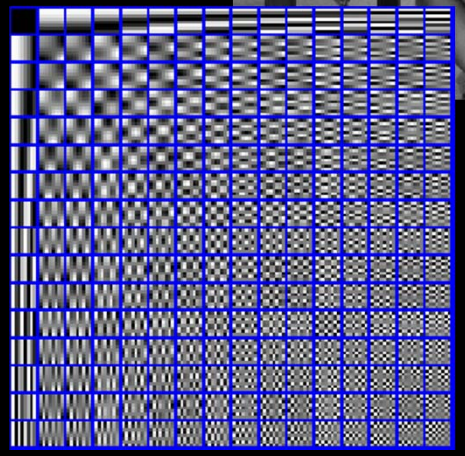
  This is done with pursuit algorithms (OMP, BP, …) that assume this iid structure.
- Is this fair? Is it correct?



Lets Apply a Simple Experiment

- We gather a set of 100,000 randomly chosen image-patches of size 8×8, extracted from the image 'Barbara'. These are our signals $\{\underline{y}_k\}_k$.
- We apply sparse-coding for these patches using the OMP algorithm (error-mode with σ=2) with the redundant DCT dictionary (256 atoms). Thus, we approximate the solution of

$$\underline{\hat{\alpha}}_k = \min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \quad \text{s.t.} \quad \|\underline{y}_k - \mathbf{D}\underline{\alpha}\|_2^2 \le N\sigma^2$$



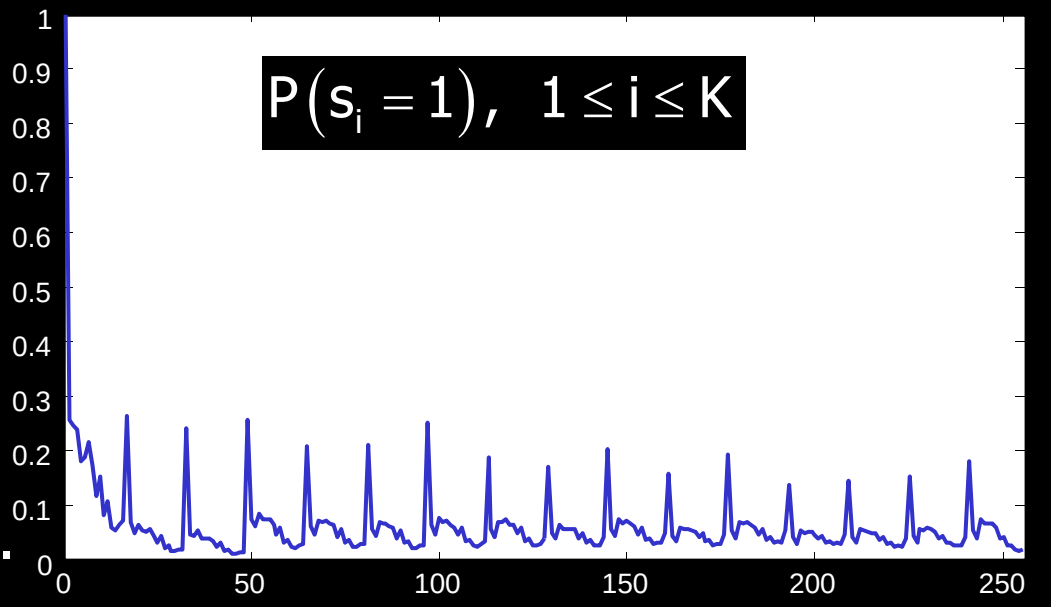
- Lets observe the obtained representations' supports. Specifically, lets check whether the iid assumption is true.

The Experiment's Results (1)

- The supports are denoted by $\underline{s}$ of length K (256), with {+1} for on-support element and {-1} for off-support ones.

$$\left\{ \boxed{s_1} \; \boxed{s_2} \; \boxed{s_3} \; \bullet\bullet\bullet \; \boxed{s_{K-1}} \; \boxed{s_K} \right\} \in \{-1,1\}^K$$

- First, we check the probability of each atom participate in the support.



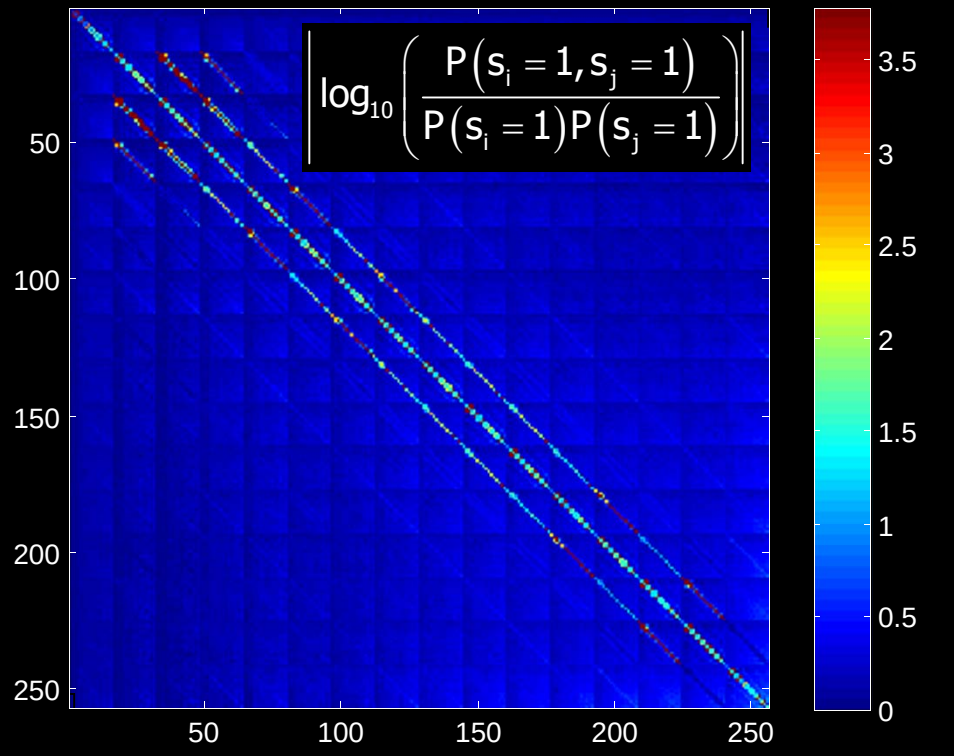
As can be seen, these probabilities are far from being uniform, even though OMP gave all atoms the same weight.

The Experiment's Results (2)

- Next, we check whether there are dependencies between the atoms, by evaluating

$$\frac{P\left(s_i = 1, s_j = 1\right)}{P\left(s_i = 1\right)P\left(s_j = 1\right)}, \quad 1 \le i, j \le K$$

- This ratio is 1 if s_i and s_j are independent, and away from 1 when there is a dependency. After abs-log, a value away from zero indicates a dependency.



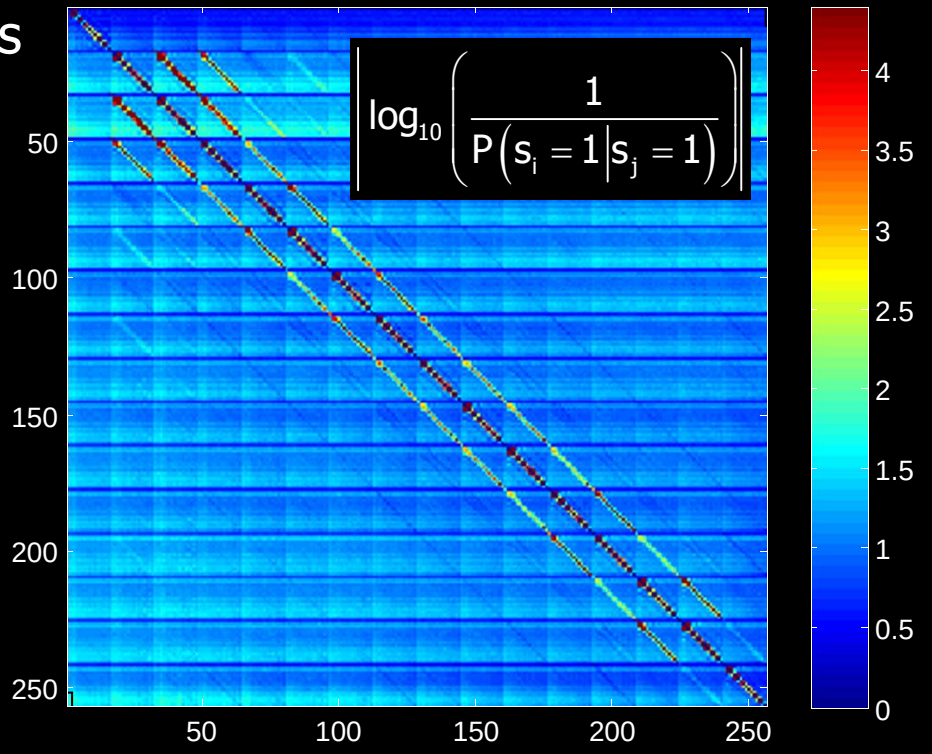
As can be seen, there are some dependencies between the elements, even though OMP did not promote them.

The Experiment's Results (3)

- We saw that there are dependencies within the support vector. Could we claim that these form a block-sparsity pattern?
- If s_i and s_j belong to the same block, we should get that

$$P\left(s_i = 1 \,|\, s_j = 1\right) = 1$$

- Thus, after abs-log, any value away from 0 means a non-block structure.



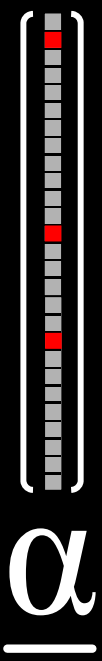
We can see that the supports do not form a block-sparsity structure, but rather a more complex one.

Our Goals

- We would like to (statistically) model these dependencies somehow.
- We will adopt a generative model for $\underline{\alpha}$ of the following form:

Generate the support vector s (of length K) by drawing it from the distribution P(s)

Generate the non-zero values for the on-support entries, drawn from the distributions $N(0, \sigma_i^2)$.



- What P(s) is? We shall follow recent work and choose the Boltzmann Machine [Garrigues & Olshausen '08] [Cevher, Duarte, Hedge & Baraniuk '09].
- Our goals in this work are:
  - Develop general-purpose pursuit algorithms for this model.
  - Estimate this model's parameters for a given bulk of data.

Part II
Basics of the Boltzmann Machine

BM – Definition

- Recall that the support vector, $\underline{s}$, is a binary vector of length K, indicating the on- and the off-support elements:

$$\left\{ \boxed{s_1} \; \boxed{s_2} \; \boxed{s_3} \; \cdots \; \boxed{s_{K-1}} \; \boxed{s_K} \right\} \in \{-1,1\}^K$$

- The Boltzmann Machine assigns the following probability to this vector:

$$P(\underline{s}) = \frac{1}{Z(\mathbf{W},\underline{b})} \exp\left\{ \underline{b}^T \underline{s} + \frac{1}{2} \underline{s}^T \mathbf{W} \underline{s} \right\}$$

- Boltzmann Machine (BM) is a special case of the more general Markov-Random-Field (MRF) model.

The Vector $\underline{b}$

- If we assume that $\mathbf{W}=\mathbf{0}$, the BM becomes

$$P(\underline{s}) = \frac{1}{Z}\exp\left\{\underline{b}^T\underline{s} + \frac{1}{2}\underline{s}^T\mathbf{W}\underline{s}\right\}$$

$$= \frac{1}{Z}\exp\left\{\underline{b}^T\underline{s}\right\} = \frac{1}{Z}\prod_{k=1}^{K}\exp(b_k s_k)$$

- In this case, the entries of $\underline{s}$ are independent.
- In this case, the probability of $s_k=+1$ is given by

$$p_k = P(s_k = 1) = \frac{\exp(b_k)}{\exp(b_k) + \exp(-b_k)}$$

Thus, p_k becomes very small (promoting sparsity) if $b_k<<0$.

The Matrix **W**

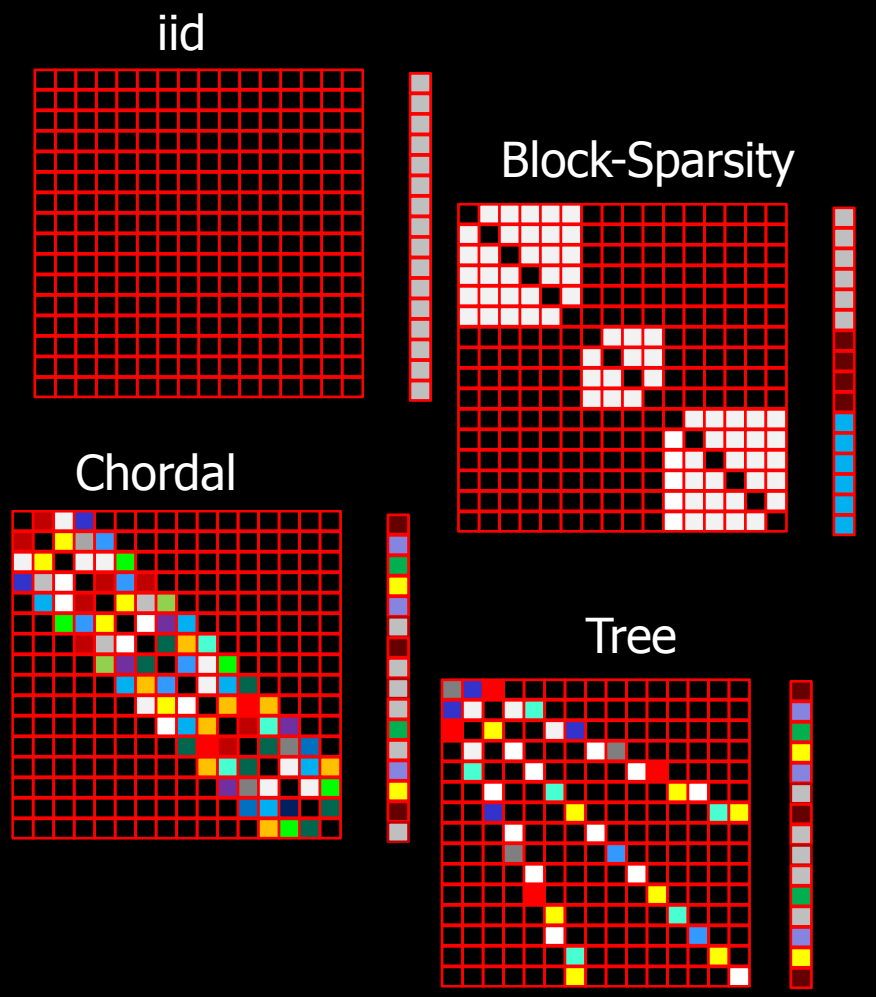
- **W** introduces the inter-dependencies between the entries of $\underline{s}$:
  - Zero value W_{ij} implies an 'independence' between s_i and s_j.
  - A positive value W_{ij} implies an 'excitatory' interaction between s_i and s_j.
  - A negative value W_{ij} implies an 'inhibitory' interaction between s_i and s_j.
  - The larger the magnitude of W_{ij}, the stronger are these forces.
- **W** is symmetric, since the interaction between i-j is the same as the one between j-i.
- We assume that diag(**W**)=$\underline{0}$, since

$$P(\underline{s}) = \frac{1}{Z}\exp\left\{\underline{b}^T\underline{s} + \frac{1}{2}\underline{s}^T\mathbf{W}\underline{s}\right\}$$

$$= \frac{1}{Z}\exp\left\{\underline{b}^T\underline{s} + \frac{1}{2}\underline{s}^T\left(\mathbf{W} - \text{diag}(\mathbf{W})\right)\underline{s} + \overbrace{\frac{1}{2}\underline{s}^T\text{diag}(\mathbf{W})\underline{s}}^{\text{trace}\{\mathbf{W}\}}\right\}$$

Special Cases of Interest

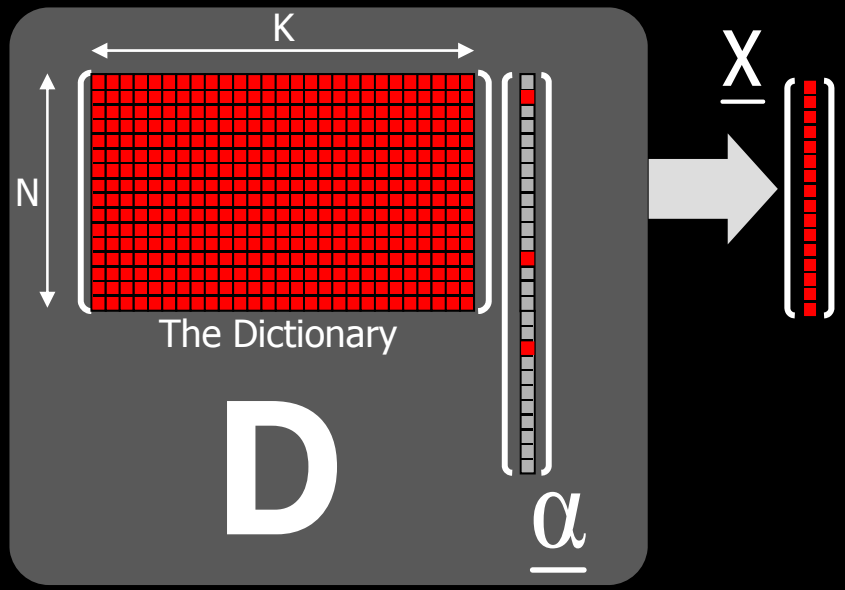
- When $\mathbf{W}=\mathbf{0}$ and $\underline{b}=c\cdot\underline{1}$, we return to the iid case we started with. If $c<<0$, this model promotes sparsity.
- When $\mathbf{W}$ is block-diagonal (up to permutation) with very high and positive entries, this corresponds to a block-sparsity structure.
- A banded $\mathbf{W}$ (up to permutation) with 2L+1 main diagonals corresponds to a chordal graph that leads to a decomposable model.
- A tree-structure can also fit this model easily.



Part III

MAP/MMSE Sparse Recovery Revisited

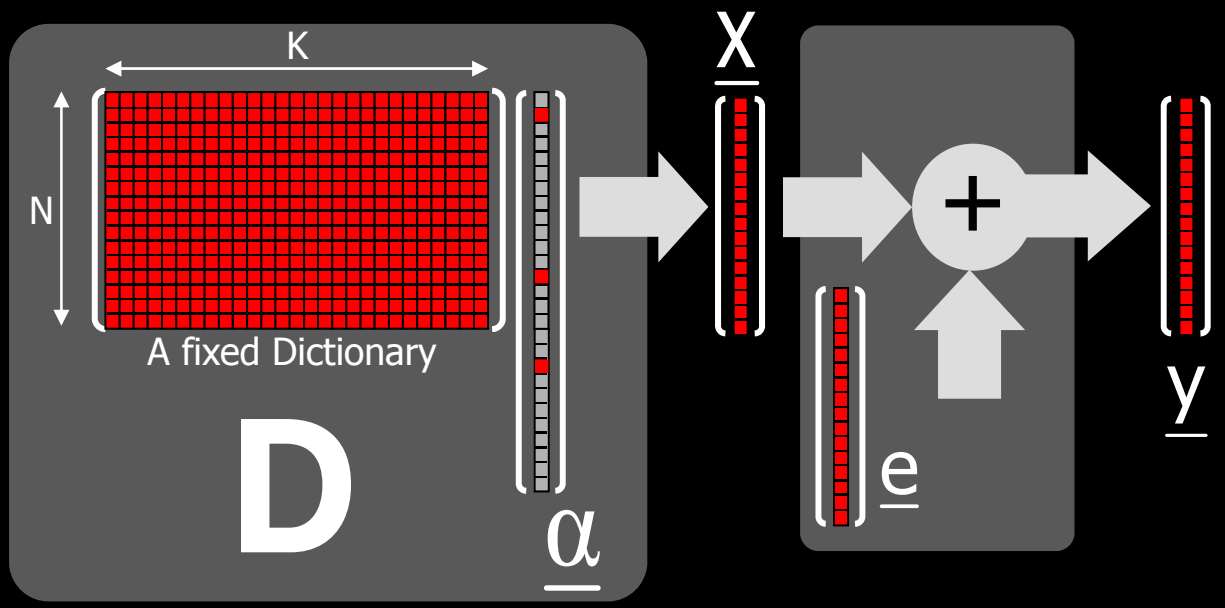
The Signal Model



- **D** is fixed and known.
- We assume that $\underline{\alpha}$ is built by:
  - Choosing the support $\underline{s}$ with BM probability $P(\underline{s})$ from all the 2^K possibilities Ω.
  - Choosing the $\underline{\alpha}_s$ coefficients using independent Gaussian entries $N\left(0,\sigma_i^2\right)$ → we shall assume that σ_i are the same ($=\sigma_\alpha$).
- The ideal signal is $\underline{x}=\mathbf{D}\underline{\alpha}=\mathbf{D}_s\underline{\alpha}_s$.

The p.d.f. $P(\underline{\alpha})$ and $P(\underline{x})$ are clear and known

Adding Noise



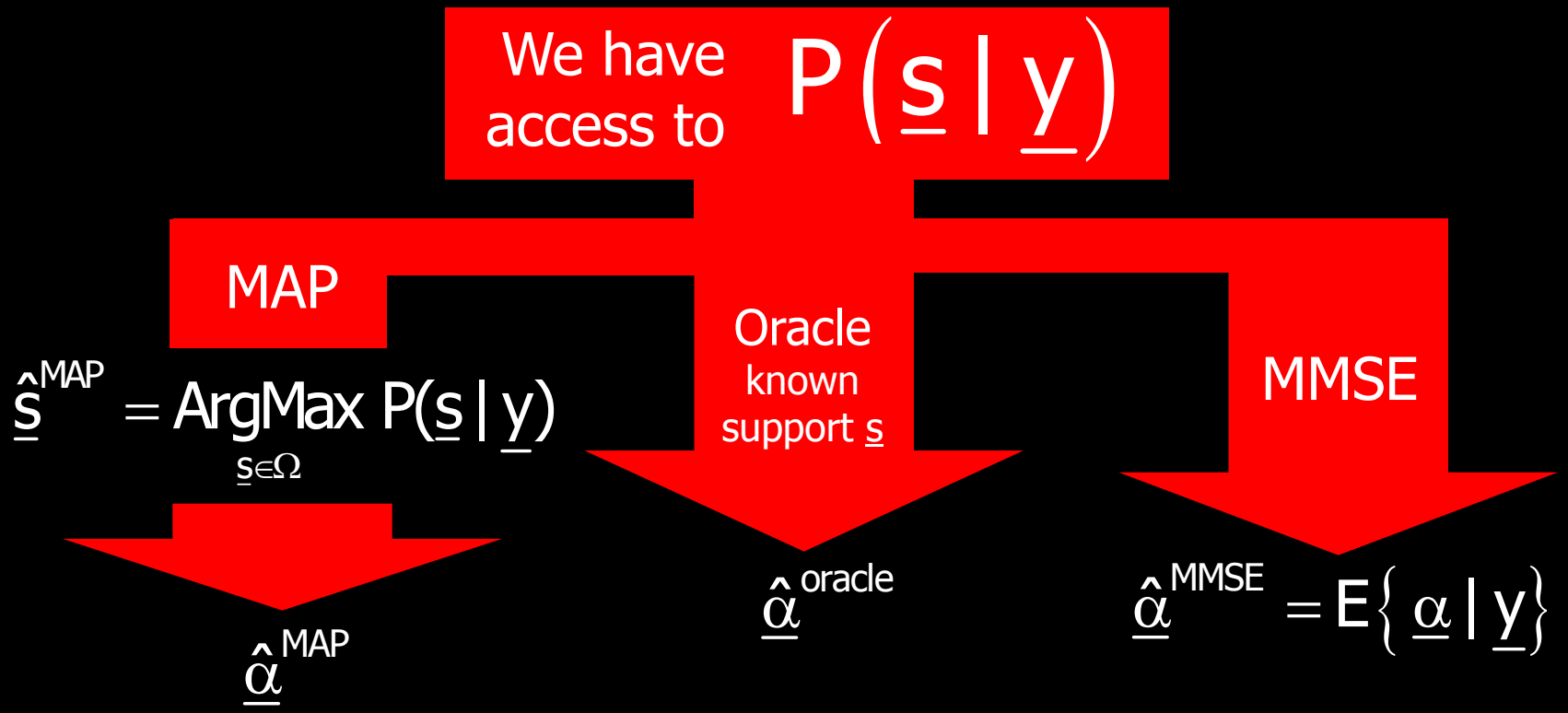
Noise Assumed:

The noise $\underline{e}$ is additive white Gaussian vector with probability $P_E(\underline{e})$

$$P(\underline{y}|\underline{x}) = C \cdot \exp\left\{-\frac{\|\underline{x}-\underline{y}\|^2}{2\sigma_e^2}\right\}$$

The conditional p.d.f.'s $P(\underline{y}|\underline{\alpha})$, $P(\underline{\alpha}|\underline{y})$, and even $P(\underline{y}|\underline{s})$, $P(\underline{s}|\underline{y})$, are all clear and well-defined (although they may appear nasty).

Pursuit via the Posterior



- There is a delicate problem due to the mixture of continuous and discrete PDF's. This is why we estimate the support using MAP first.
- MAP estimation of the representation is obtained using the oracle's formula.

The Oracle

$$P\left(\underline{\alpha} \mid \underline{y}, \underline{s}\right) = P\left(\underline{\alpha}_s \mid \underline{y}\right)$$

$$P\left(\underline{y} \mid \underline{\alpha}_s\right) \propto \exp\left\{-\frac{\left\|\underline{y} - \mathbf{D}_s \underline{\alpha}_s\right\|^2}{2\sigma_e^2}\right\}$$

$$P\left(\underline{\alpha}_s\right) \propto \exp\left\{-\frac{\left\|\underline{\alpha}_s\right\|_2^2}{2\sigma_\alpha^2}\right\}$$

$$P\left(\underline{\alpha}_s \mid \underline{y}\right) \propto \exp\left\{-\frac{\left\|\underline{y} - \mathbf{D}_s \underline{\alpha}_s\right\|^2}{2\sigma_e^2} - \frac{\left\|\underline{\alpha}_s\right\|_2^2}{2\sigma_\alpha^2}\right\}$$

$$\hat{\underline{\alpha}}_s = \left[\mathbf{D}_s^T \mathbf{D}_s + \frac{\sigma_e^2}{\sigma_\alpha^2}\mathbf{I}\right]^{-1} \mathbf{D}_s^T \underline{y}$$

Comments:

- This estimate is both the MAP and MMSE.
- The oracle estimate of $\underline{x}$ is obtained by multiplication by $\mathbf{D}_s$.

MAP Estimation

$$\hat{\underline{s}}^{MAP} = \underset{\underline{s}\in\Omega}{\text{ArgMax}}\, P(\underline{s}\,|\,\underline{y}) = \underset{\underline{s}\in\Omega}{\text{ArgMax}} \frac{P(\underline{y}\,|\,\underline{s})P(\underline{s})}{P(\underline{y})}$$

$$P(\underline{y}\,|\,\underline{s}) = \int_{\underline{\alpha}_s} P(\underline{y}\,|\,\underline{s},\underline{\alpha}_s)P(\underline{\alpha}_s)d\underline{\alpha}_s$$

$$= \ldots.$$

$$\propto \left(\frac{\sigma_e^2}{\sigma_\alpha^2}\right)^{\frac{|s|}{2}} \cdot \exp\left\{\frac{\underline{h}_s^T \mathbf{Q}_s^{-1}\underline{h}_s}{2\sigma_e^2} - \frac{\log(\det(\mathbf{Q}_s))}{2}\right\}$$

Based on our prior for generating the support

$$P(\underline{s}) = C \cdot \exp\left(\underline{b}^T\underline{s} + 0.5\underline{s}^T\mathbf{W}\underline{s}\right)$$

$$P(\underline{s}|\underline{y}) \propto \left(\frac{\sigma_e^2}{\sigma_\alpha^2}\right)^{\frac{|s|}{2}} \cdot \exp\left\{\frac{\underline{h}_s^T \mathbf{Q}_s^{-1}\underline{h}_s}{2\sigma_e^2} - \frac{\log(\det(\mathbf{Q}_s))}{2} + \underline{b}^T\underline{s} + \frac{1}{2}\underline{s}^T\mathbf{W}\underline{s}\right\}$$

The MAP

$$\hat{\underline{s}}^{MAP} = \underset{s\in\Omega}{\text{ArgMax}} \exp\left\{ \frac{\underline{h}_s^T \mathbf{Q}_s^{-1} \underline{h}_s}{2\sigma_e^2} - \frac{\log(\det(\mathbf{Q}_s))}{2} + \left(\underline{b} - \frac{1}{4}\log\frac{\sigma_\alpha^2}{\sigma_e^2}\right)^T \underline{s} + \frac{1}{2}\underline{s}^T \mathbf{W} \underline{s} \right\}$$

Implications:

- The MAP estimator requires to test all the possible supports for the maximization. For the found support, the oracle formula is used.
- In typical problems, this process is impossible as there is a combinatorial set of possibilities.
- In order to use the MAP approach, we shall turn to approximations.

The MMSE

$$\underline{\hat{\alpha}}^{MMSE} = E\{\underline{\alpha} \mid \underline{y}\} = \sum_{s\in\Omega} P(s \mid \underline{y}) \cdot E\{\underline{\alpha} \mid \underline{y}, s\}$$

This is the oracle for s, as we have seen before

$$E\{\underline{\alpha} \mid \underline{y}, s\} = \underline{\hat{\alpha}}_s = \mathbf{Q}_s^{-1}\underline{h}_s$$

$$P(\underline{s}|\underline{y}) \propto \left(\frac{\sigma_e^2}{\sigma_\alpha^2}\right)^{\frac{|s|}{2}} \cdot \exp\left\{\frac{\underline{h}_s^T \mathbf{Q}_s^{-1}\underline{h}_s}{2\sigma_e^2} - \frac{\log(\det(\mathbf{Q}_s))}{2} + \underline{b}^T\underline{s} + \frac{1}{2}\underline{s}^T\mathbf{W}\underline{s}\right\}$$

$$\underline{\hat{\alpha}}^{MMSE} = \sum_{s\in\Omega} P(s \mid \underline{y}) \cdot \underline{\hat{\alpha}}_s$$

The MMSE

$$\underline{\hat{\alpha}}^{MMSE} = E\left\{\underline{\alpha} \mid \underline{y}\right\} = \sum_{s\in\Omega} P(s \mid \underline{y}) \cdot E\left\{\underline{\alpha} \mid \underline{y}, s\right\}$$

Implications:

$$\boxed{\underline{\hat{\alpha}}^{MMSE} = \sum_{s\in\Omega} P(s \mid \underline{y}) \cdot \underline{\hat{\alpha}}_s}$$

- The best estimator (in terms of L_2 error) is a weighted average of many sparse representations!!!
- As in the MAP case, in typical problems one cannot compute this expression, as the summation is over a combinatorial set of possibilities. We should propose approximations here as well.

Part IV
Construct Practical Pursuit Algorithms

An OMP-Like Algorithm

$$\hat{\underline{s}}^{MAP} = \underset{s\in\Omega}{\text{ArgMax}} \exp\left\{ \frac{\underline{h}_s^T \mathbf{Q}_s^{-1} \underline{h}_s}{2\sigma_e^2} - \frac{\log(\det(\mathbf{Q}_s))}{2} + \left(\underline{b} - \frac{1}{4}\log\frac{\sigma_\alpha^2}{\sigma_e^2} \right)^T \underline{s} + \frac{1}{2}\underline{s}^T \mathbf{W} \underline{s} \right\}$$

The Core Idea – Gather $\underline{s}$ one element at a time greedily:

- Initialize all entries with {-1}.
- Sweep through all entries and check which gives the maximal growth of the expression – set it to {+1}.
- Proceed this way till the overall expression decreases for any additional {+1}.

A THR-Like Algorithm

$$\hat{\underline{s}}^{MAP} = \underset{s\in\Omega}{\text{ArgMax}} \ \exp\left\{ \frac{\underline{h}_s^T \mathbf{Q}_s^{-1} \underline{h}_s}{2\sigma_e^2} - \frac{\log(\det(\mathbf{Q}_s))}{2} + \left(\underline{b} - \frac{1}{4}\log\frac{\sigma_\alpha^2}{\sigma_e^2} \right)^T \underline{s} + \frac{1}{2}\underline{s}^T \mathbf{W} \underline{s} \right\}$$

The Core Idea – Find the on-support in one-shut:

- Initialize all entries with {-1}.
- Sweep through all entries and check for each the obtained growth of the expression – sort (down) the elements by this growth.
- Choose the first $|\underline{s}|$ elements as {+1}. Their number should be chosen so as to maximize the MAP expression (by checking 1,2,3, …).

An Subspace-Pursuit-Like Algorithm

$$\underline{\hat{s}}^{MAP} = \underset{s\in\Omega}{\text{ArgMax}} \exp\left\{ \frac{\underline{h}_s^T \mathbf{Q}_s^{-1} \underline{h}_s}{2\sigma_e^2} - \frac{\log(\det(\mathbf{Q}_s))}{2} + \left(\underline{b} - \frac{1}{4}\log\frac{\sigma_\alpha^2}{\sigma_e^2}\right)^T \underline{s} + \frac{1}{2}\underline{s}^T \mathbf{W} \underline{s} \right\}$$

The Core Idea – Add and remove entries from $\underline{s}$ iteratively:

- Operate as in the THR algorithm, and choose the first $|\underline{s}|+\Delta$ entries as $\{+1\}$, where $|s|$ is maximizes the posterior.
- Consider removing each of the $|\underline{s}|+\Delta$ elements, and see the obtained decrease in the posterior – remove the weakest Δ of them.
- Add/remove Δ entries … similar to [Needell & Tropp `09],[Dai & Milenkovic `09].

An MMSE Approximation Algorithm

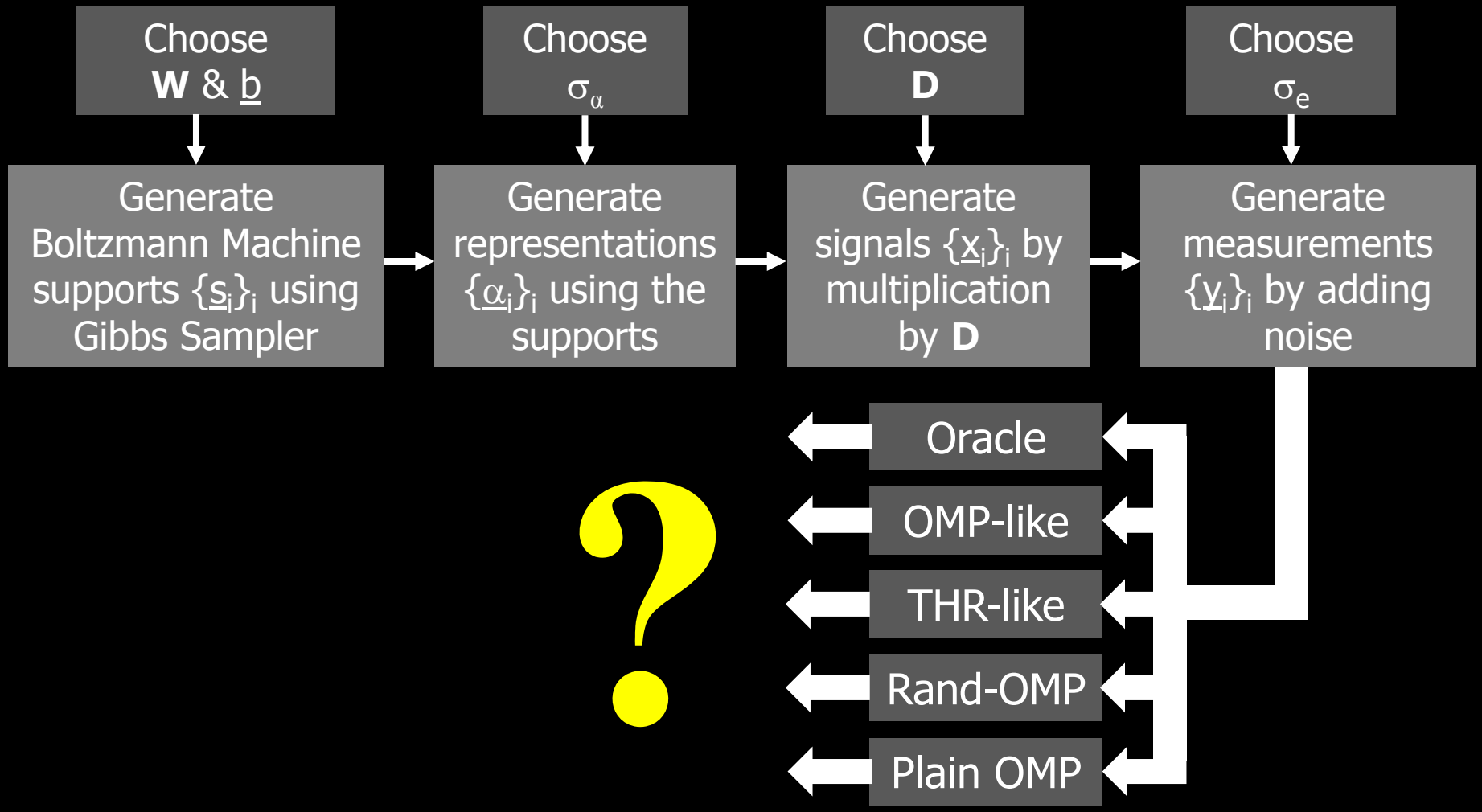
$$P\left(\underline{s}\middle|\underline{y}\right) \propto \left(\frac{\sigma_e^2}{\sigma_\alpha^2}\right)^{\frac{|s|}{2}} \cdot \exp\left\{\frac{\underline{h}_s^T \mathbf{Q}_s^{-1} \underline{h}_s}{2\sigma_e^2} - \frac{\log(\det(\mathbf{Q}_s))}{2} + \underline{b}^T \underline{s} + \frac{1}{2}\underline{s}^T \mathbf{W} \underline{s}\right\}$$

$$\Rightarrow \quad \underline{\hat{\alpha}}^{MMSE} = \sum_{s\in\Omega} P(s \mid \underline{y}) \cdot \mathbf{Q}_s^{-1} \underline{h}_s$$

The Core Idea – Draw random supports from the posterior and average their oracle estimates:

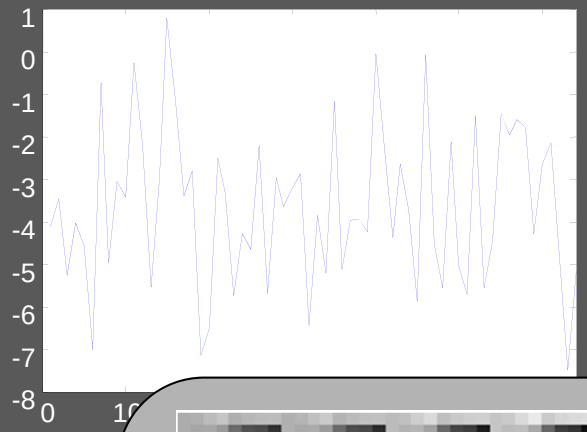
- Operate like the OMP-like algorithm, but choose the next +1 by randomly drawing it based on the marginal posteriors.
- Repeat this process several times, and average their oracle's solutions.
- This is an extension of the Random-OMP algorithm [Elad & Yavneh, `09].

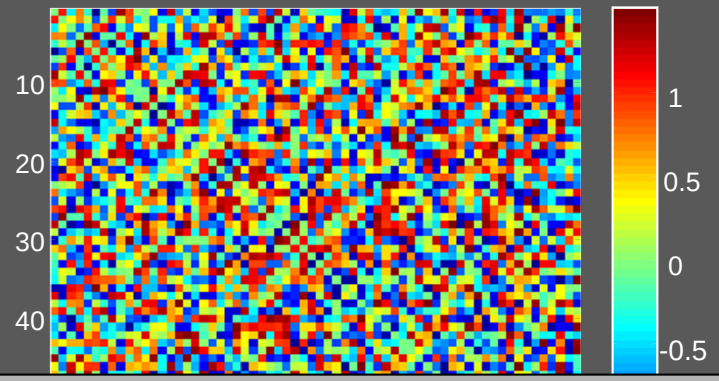
Lets Experiment



Chosen Parameters

The Boltzmann parameters $\underline{b}$ and $\mathbf{W}$

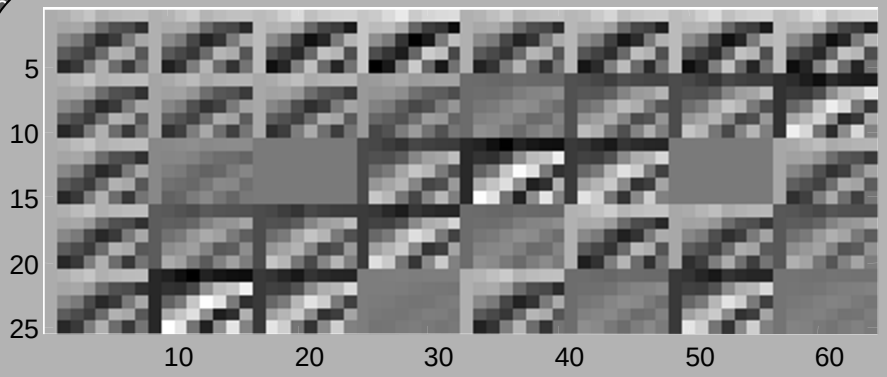


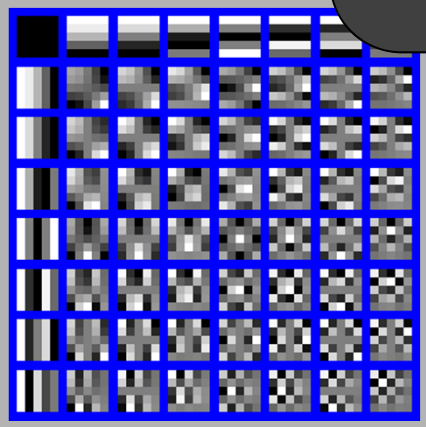


The signal and noise STD's

$\sigma_\alpha = 60$

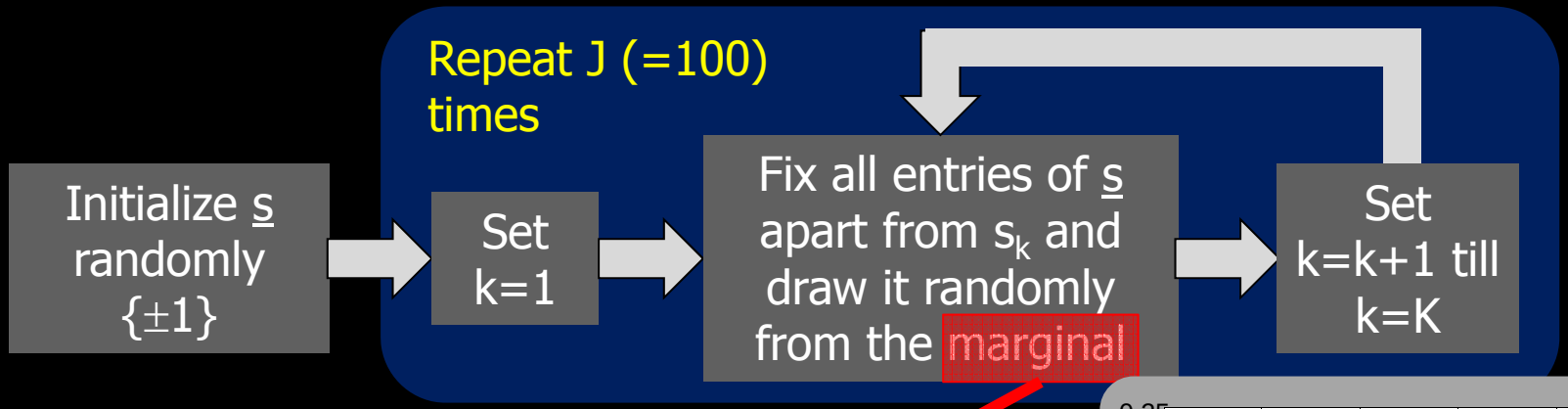
$\sigma_e = 4 \div 80$





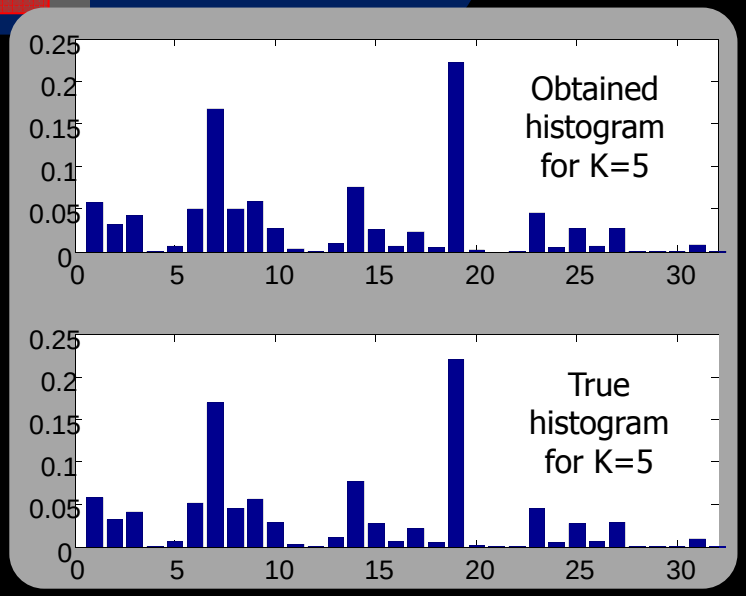
The dictionary $\underline{\mathbf{D}}$

Gibbs Sampler for Generating $\underline{s}$



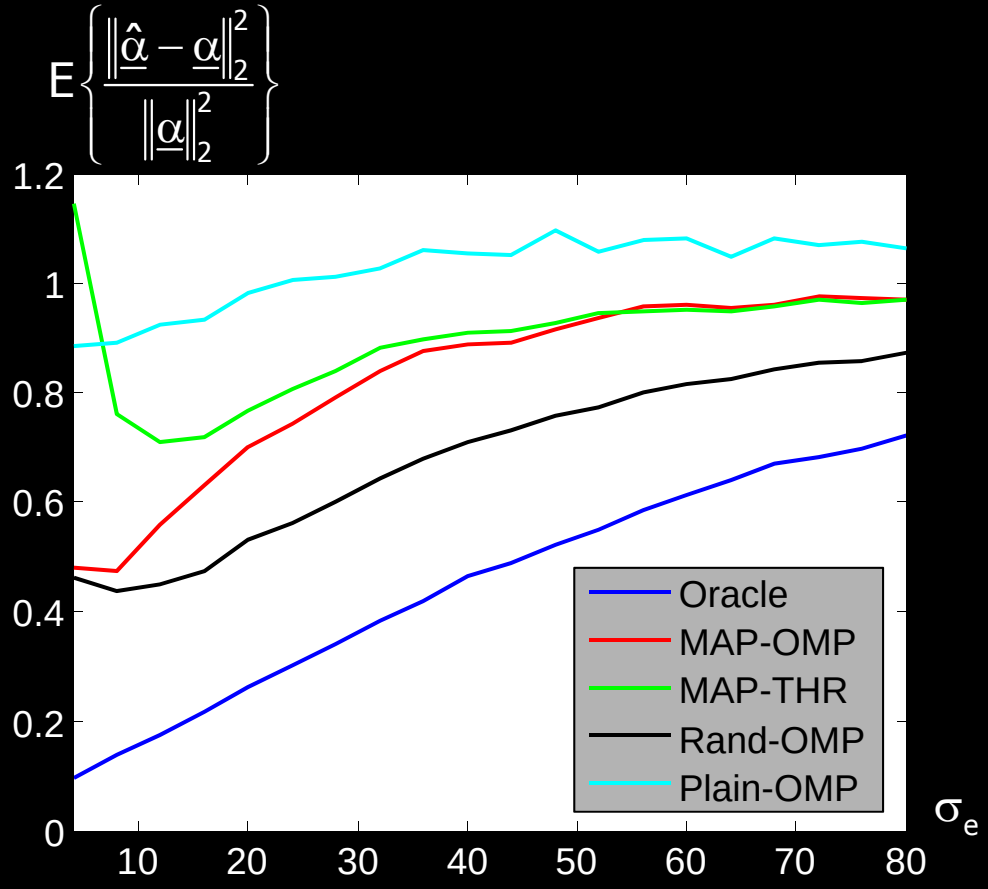
$$P\left(s_k \middle| \underline{s}_k^C\right) = \frac{P\left(s_k, \underline{s}_k^C\right)}{P\left(\underline{s}_k^C\right)}$$

$$= \frac{\exp\left(s_k b_k + s_k \mathbf{W}_k^T \underline{s}_k^C\right)}{2\cosh\left(b_k + \mathbf{W}_k^T \underline{s}_k^C\right)}$$



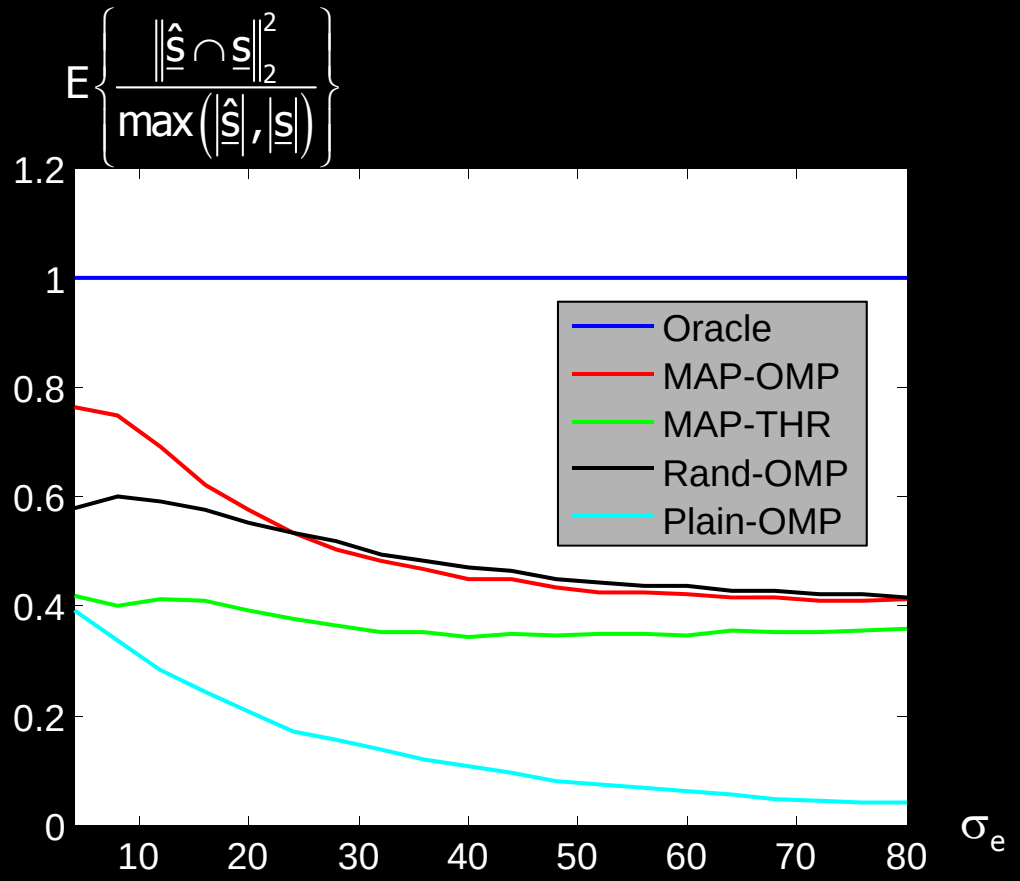
Results – Representation Error

- As expected, the oracle performs the best.
- The Rand-OMP is the second best, being an MMSE approximation.
- MAP-OMP is better than MAP-THR.
- There is a strange behavior in MAP-THR for weak noise – should be studied.
- The plain-OMP is the worst, with practically useless results.



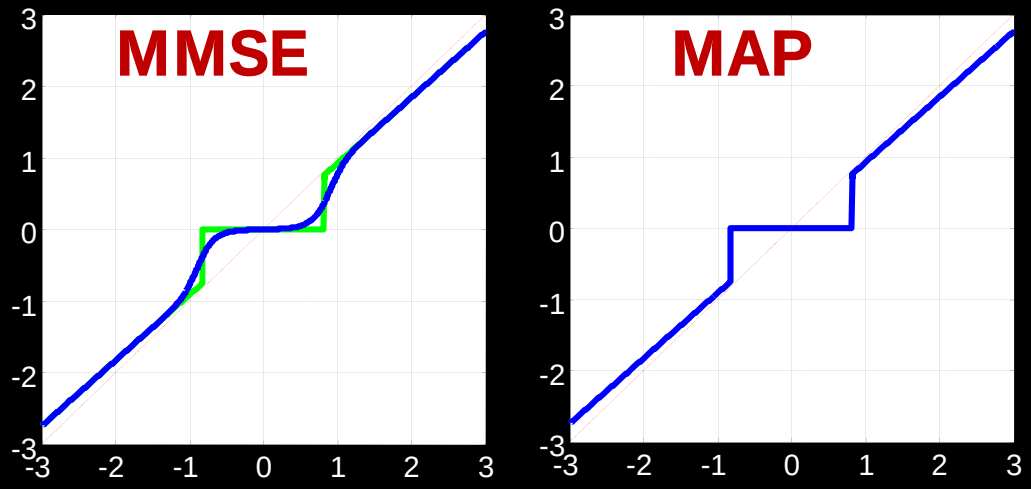
Results – Support Error

- Generally the same story.
- Rand-OMP result is not sparse! It is compared by choosing the first |s| dominant elements, where |s| is the number of elements chosen by the MAP-OMP. As can be seen, it is not better than the MAP-OMP.



The Unitary Case

- What happens when **D** is square and unitary?
- We know that in such a case, if **W**=0 (i.e., no interactions), then MAP and MMSE have closed-form solutions, in the form of a shrinkage [Turek, Yavneh, Protter, Elad, `10].



$$\underline{\hat{\alpha}} = \rho\left\{\mathbf{D}^{T}\underline{y}\right\}$$

- Can we hope for something similar for the more general **W**? The answer is (partly) positive!

The Posterior Again

$$P\left(\underline{s}\middle|\underline{y}\right) \propto \left(\frac{\sigma_e^2}{\sigma_\alpha^2}\right)^{\frac{|s|}{2}} \cdot \exp\left\{\frac{\underline{h}_s^T \mathbf{Q}_s^{-1} \underline{h}_s}{2\sigma_e^2} - \frac{\log(\det(\mathbf{Q}_s))}{2} + \underline{b}^T \underline{s} + \frac{1}{2}\underline{s}^T \mathbf{W} \underline{s}\right\}$$

$$\mathbf{Q}_s = \mathbf{D}_s^T \mathbf{D}_s + \frac{\sigma_e^2}{\sigma_\alpha^2}\mathbf{I} = \frac{\sigma_\alpha^2 + \sigma_e^2}{\sigma_\alpha^2}\mathbf{I}$$

$$\underline{h}_s = \mathbf{D}_s^T \underline{y}$$

$$P\left(\underline{s}\middle|\underline{y}\right) \propto \exp\left\{\underline{q}^T \underline{s} + \frac{1}{2}\underline{s}^T \mathbf{W} \underline{s}\right\} \quad \text{where} \quad \underline{q} = \underline{b} + \frac{1}{4}\left(\frac{\sigma_\alpha^2}{\sigma_e^2(\sigma_\alpha^2+\sigma_e^2)}\mathbf{D}^T \underline{y} - \log\left(1+\frac{\sigma_\alpha^2}{\sigma_e^2}\right)\right)$$

When **D** is unitary, the BM prior and the resulting posterior are conjugate distributions

Implications

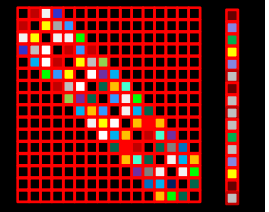
The MAP estimation becomes the following Boolean Quadratic Problem (NP-Hard!)

$$\hat{\underline{s}} = \underset{\underline{s}\,|\,s_k^2=1}{\text{ArgMax}}\, P\left(\underline{s}\middle|\underline{y}\right) = \underset{\underline{s}\,|\,s_k^2=1}{\text{ArgMax}}\, \exp\left\{\underline{q}^T\underline{s} + \frac{1}{2}\underline{s}^T\mathbf{W}\underline{s}\right\}$$

Approximate

Assume $\mathbf{W} \geq 0$

Assume banded $\mathbf{W}$



Multi-user detection in communication [Verdu, 1998]

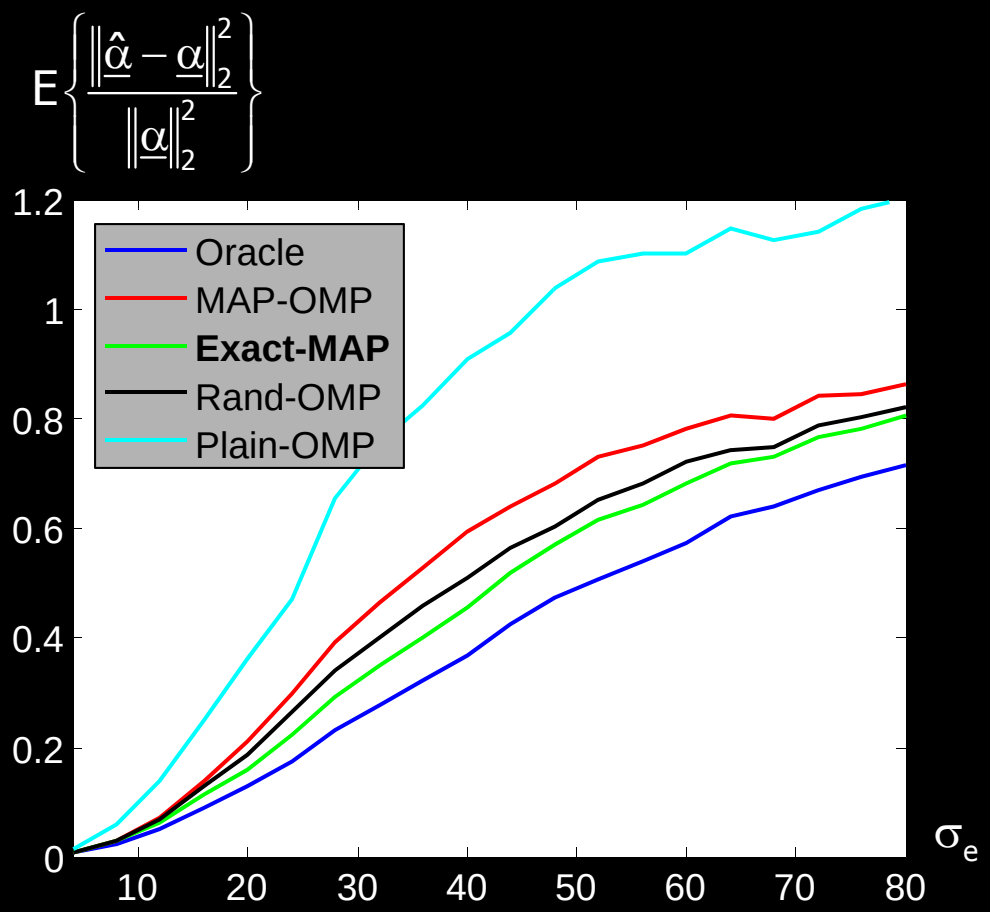
Graph-cut can be used for exact solution [Cevher, Duarte, Hedge, & Baraniuk 2009]

Our work: Belief-Propagation for exact solution with $O(K \cdot 2^L)$

What about exact MMSE? It is possible too! We are working on it NOW.

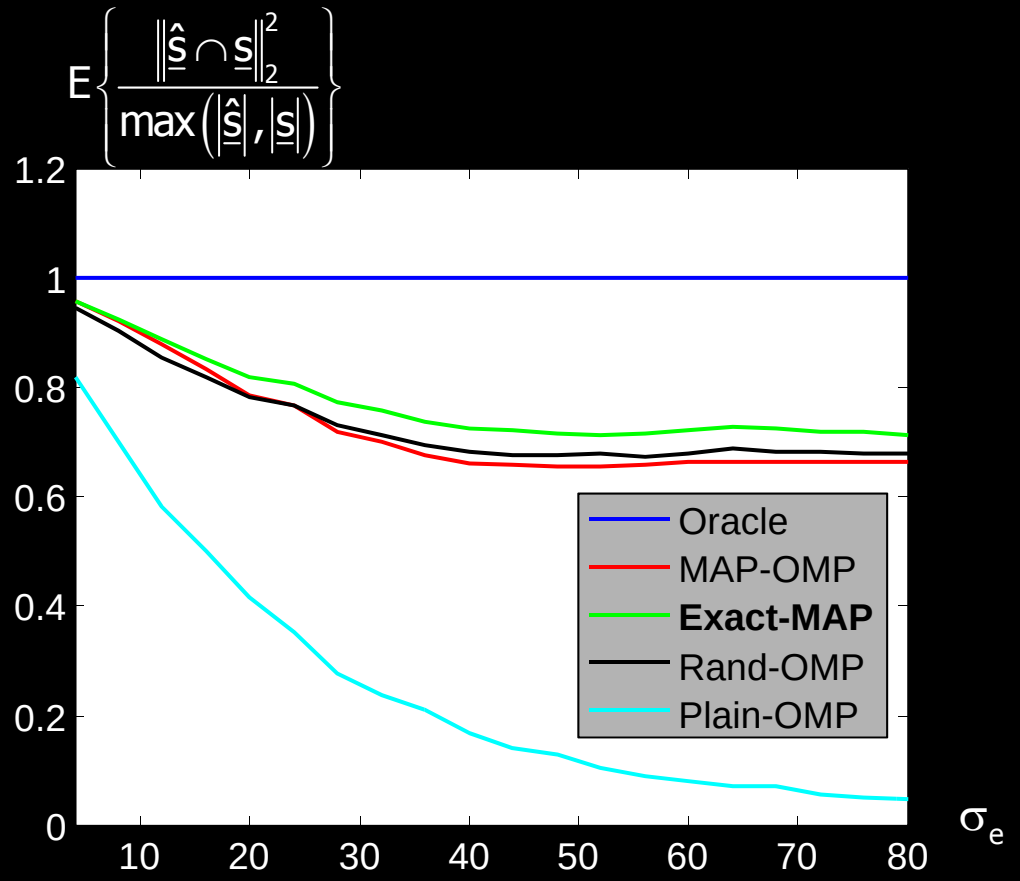
Results

- **W** (of size 64×64) in this experiment has 11 diagonals.
- All algorithms behave as expected.
- Interestingly, the exact-MAP outperforms the approximated MMSE algorithm.
- The plain-OMP (which would have been ideal MAP for the iid case) is performing VERY poorly.



Results

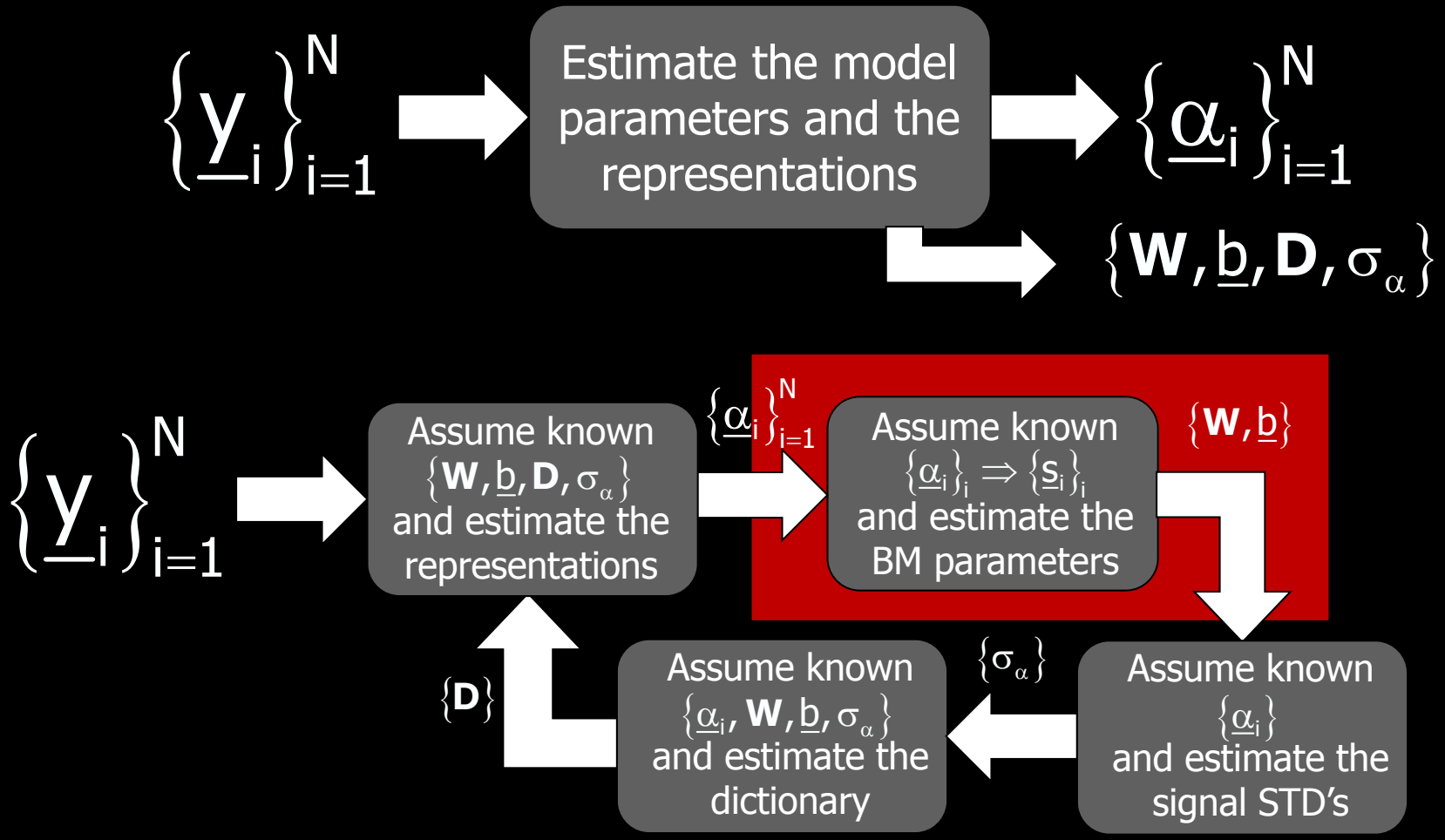
- As before …



Part V

First Steps Towards Adaptive Recovery

The Grand-Challenge



Maximum-Likelihood?

$$\left\{\underline{s}_i\right\}_{i=1}^{N} \Rightarrow \boxed{\text{ML-Estimate of the BM parameters}} \Rightarrow \left\{\mathbf{W},\underline{b}\right\}$$

$$\left\{\hat{\mathbf{W}},\hat{\underline{b}}\right\} = \underset{\mathbf{W},\underline{b}}{\text{ArgMax}}\ P\left(\left\{\underline{s}_i\right\}_{i=1}^{N}\middle|\mathbf{W},\underline{b}\right) = \underset{\mathbf{W},\underline{b}}{\text{ArgMax}}\prod_{i=1}^{N} P\left(\underline{s}_i\middle|\mathbf{W},\underline{b}\right)$$

$$= \underset{\mathbf{W},\underline{b}}{\text{ArgMax}}\frac{1}{Z\left(\mathbf{W},\underline{b}\right)^N}\prod_{i=1}^{N}\exp\left\{\underline{b}^T\underline{s}_i + \frac{1}{2}\underline{s}_i^T\mathbf{W}\underline{s}_i\right\}$$

Problem !!! We do not have access to the partition function $Z(\mathbf{W},\underline{b})$.

Solution: Maximum-Pseudo-Likelihood

$$\left\{\underline{s}_i\right\}_{i=1}^{N} \Rightarrow \boxed{\textbf{MPL}\text{-Estimate of the BM parameters}} \Rightarrow \left\{\mathbf{W}, \underline{b}\right\}$$

Likelihood function $P\left(\underline{s}\middle|\mathbf{W},\underline{b}\right)$ $\Rightarrow$ $\prod_{k=1}^{K} P\left(s_k\middle|\mathbf{W},\underline{b},\underline{s}_k^C\right)$ Pseudo-Likelihood function

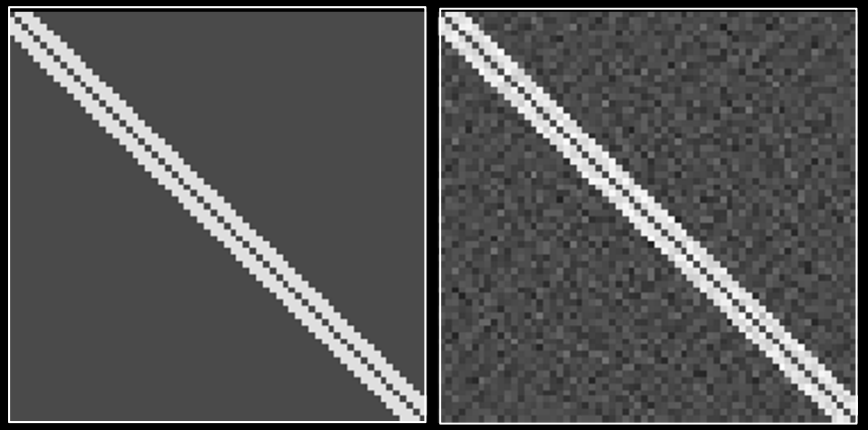
$$P\left(s_k\middle|\underline{s}_k^C\right)=\frac{\exp\left(s_k b_k + s_k \mathbf{W}_k^T \underline{s}_k^C\right)}{2\cosh\left(b_k + \mathbf{W}_k^T \underline{s}_k^C\right)} \Rightarrow \left\{\mathbf{W},\hat{\underline{b}}\right\} = \underset{\mathbf{W},\underline{b}}{\text{ArgMax}} \left[\begin{array}{r} \text{tr}\left\{\mathbf{S}^T\mathbf{W}\mathbf{S}\right\} + \underline{1}^T\mathbf{S}^T\underline{b} \\ -\underline{1}^T\rho\left(\mathbf{W}\mathbf{S}+\underline{b}\right)\underline{1}\end{array}\right]$$

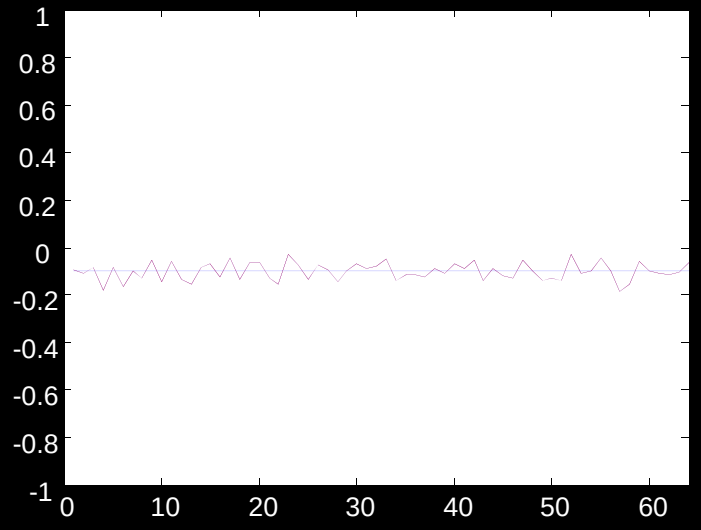
- The function $\rho(\cdot)$ is $\log(\cosh(\cdot))$. This is a convex programming task.
- MPL is known to be a consistent estimator.
- We solve this problem using SESOP+SD.

Simple Experiment & Results

We perform the following experiment:

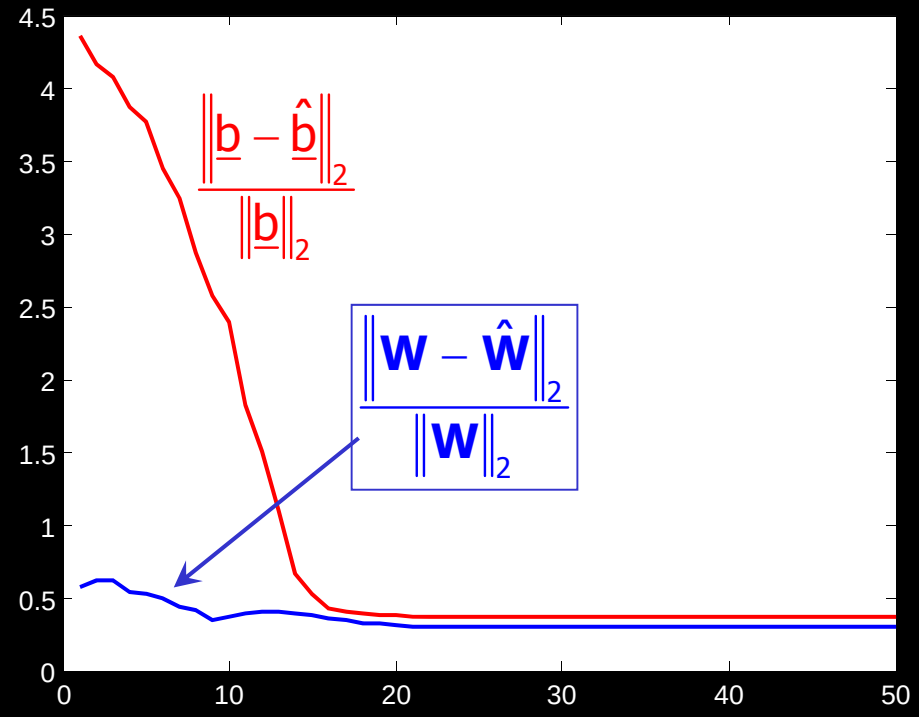
- **W** is 64×64 banded, with L=3, and all values fixed (0.2).
- b is a constant vector (-0.1).
- 10,000 supports are created from this BM distribution – the average cardinality is 16.5 (out of 64).
- We employ 50 iterations of the MPL-SESOP to estimate the BM parameters.
- Initialization: **W**=0 and $\underline{b}$ is set by the scalar probabilities for each entry.





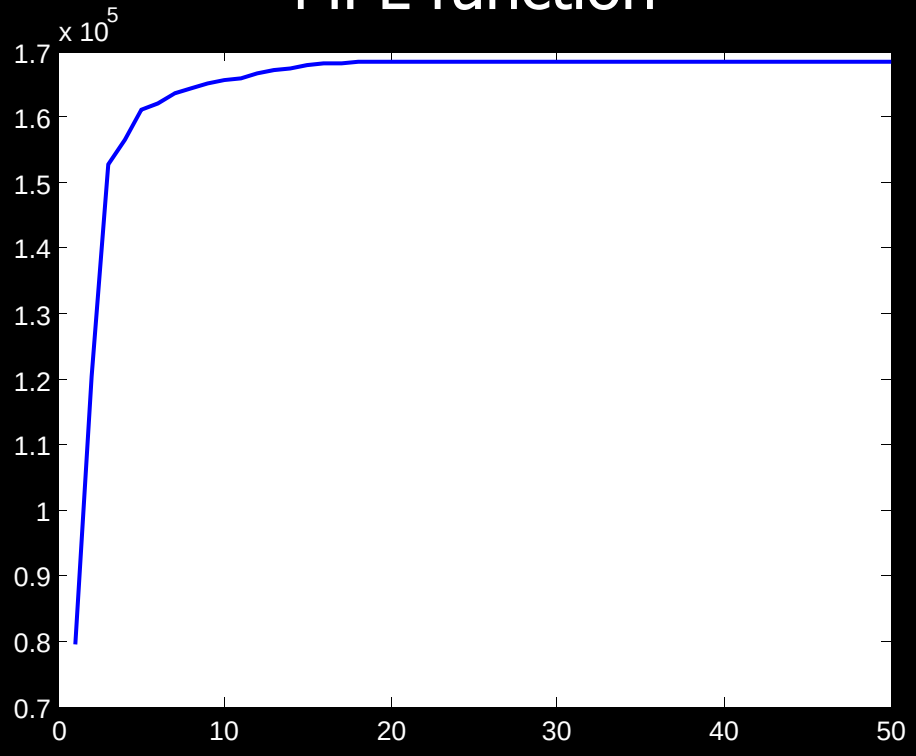
Simple Experiment & Results

Estimation Error



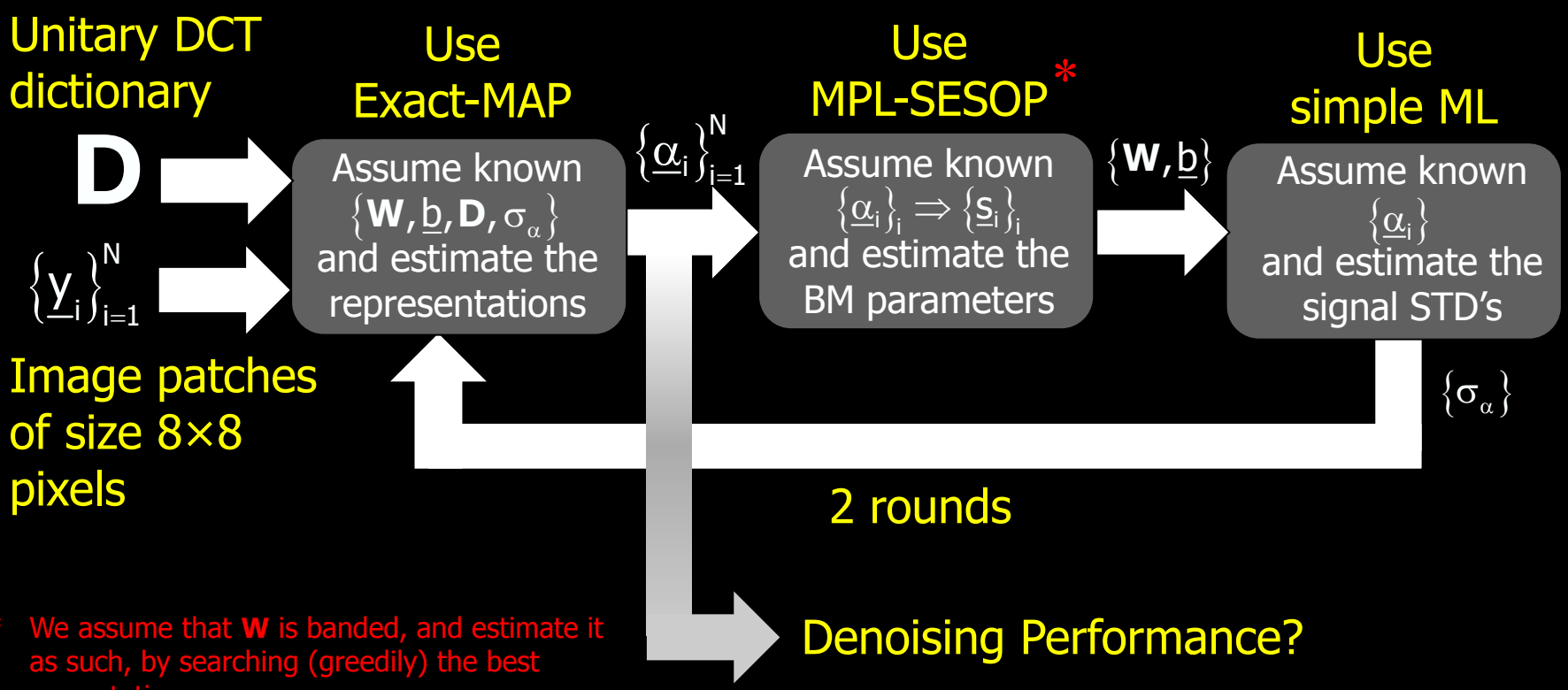
Iterations

MPL function



Iterations

Testing on Image Patches



* We assume that **W** is banded, and estimate it as such, by searching (greedily) the best permutation.

- This experiment was done for patches from 5 images (Lenna, Barbara, House, Boat, Peppers), with varying σ_e in the range [2-25].
- The results show an improvement of ~0.9dB over plain-OMP.

Part VI

Summary and Conclusion

Today We Have Seen that ...

Sparsity and Redundancy are important ideas that can be used in designing better tools in signal/image processing

Is it enough?

No! We know that the found representations exhibit dependencies. Taking these into account could further improve the model and its behavior

In this work

We use the Boltzmann Machine and propose:

- A general set of pursuit methods
- Exact MAP pursuit for the unitary case
- Estimation of the BM parameters

What next?

We intend to explore:

- Other pursuit methods
- Theoretical study of these algorithms
- K-SVD + this model
- Applications ...