Adaptive Sparse Representations and their Applications

Saiprasad Ravishankar

Department of Electrical and Computer Engineering and Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

November 11, 2014
Key Topics of Thesis

- **Intro to prior sparse models & advantages of Transform model**

- **Square Transform Models**
  - Unstructured square transform learning
  - Doubly sparse transform learning
  - Online learning for Big Data
  - Convex formulations for transform learning

- **Overcomplete Transform Models**
  - Unstructured overcomplete transform learning
  - Learning structured overcomplete transforms with block cosparse (OCTOBOS)

- **Applications**: Sparse Representation, Denoising, Classification, Blind compressed sensing (BCS), magnetic resonance imaging (MRI)

- Other Topics: Adaptive sampling for BCS-based MRI
Introduction
Synthesis Model (SM) for Sparse Representation

Given a signal \( y \in \mathbb{R}^n \), and dictionary \( D \in \mathbb{R}^{n \times K} \), we assume 
\[ y = Dx \] 
with \( \|x\|_0 \ll K \).

Real world signals modeled as 
\[ y = Dx + e \] 
e is deviation term.

Given \( D \), sparsity level \( s \), the synthesis sparse coding problem is 
\[ \hat{x} = \arg \min_x \|y - Dx\|_2^2 \quad s.t. \quad \|x\|_0 \leq s \]

This problem is NP-hard.

Greedy and \( \ell_1 \)-relaxation algorithms can be computationally expensive.
Analysis Model (AM) for Sparse Representation

- (Strict) AM: Given a signal $y \in \mathbb{R}^n$, and analysis dictionary $\Omega \in \mathbb{R}^{m \times n}$, $\|\Omega y\|_0 \ll m$.

Noisy Signal Analysis Model (NSAM): $y = q + e$, $\Omega q = z$ sparse.

- Given $\Omega$, co-sparsity level $t$, the analysis sparse coding problem is

$$\hat{q} = \arg \min_q \|y - q\|_2^2 \text{ s.t. } \|\Omega q\|_0 \leq m - t$$

This problem is NP-hard.

- Greedy$^1$ and $\ell_1$-relaxation$^2$ algorithms are computationally expensive.

$^1$ [Rubinstein et al. '12], $^2$ [Yaghoobi et al. '12].
Transform Model (TM) for Sparse Representation

Given a signal \( y \in \mathbb{R}^n \), and transform \( W \in \mathbb{R}^{m \times n} \), we model \( Wy = x + \eta \) with \( \|x\|_0 \ll m \) and \( \eta \) - error term.

Natural signals are approximately sparse in Wavelets, DCT, Curvelets.

Given \( W \), and sparsity \( s \), \textit{transform sparse coding} is

\[
\hat{x} = \arg \min_x \|Wy - x\|_2^2 \quad \text{s.t.} \quad \|x\|_0 \leq s
\]

\( \hat{x} = H_s(Wy) \) computed by thresholding \( Wy \) to the \( s \) largest magnitude elements. \textit{Sparse coding is cheap!} Signal recovered as \( W^\dagger \hat{x} \).

Sparsifying transforms exploited for compression (JPEG2000), etc.
Generality of Transform Model

- TM more general than AM.
- TM more general than the notion of compressibility of $Wy$.
- TM more general than NSAM:
  - $y = q + e$ with $Wq = z$ sparse $\Rightarrow Wy = Wq + We = z + We$.
  - However, $Wy = x + \eta$ with $x$ sparse $\not\Rightarrow y = q' + e'$ with $Wq' = x$.
  - TM does not enforce $x \in R(W)$!
- Synthesis is the most general.
Learning formulations - typically non-convex and NP-hard.

Approximate algorithms for Synthesis Learning: MOD$^3$, K-SVD$^4$, online dictionary learning$^5$, etc.

Heuristics for Analysis Learning:
- (Strict) Analysis: Sequential Minimal Eigenvalues$^6$, AOL$^7$.
- Noisy Analysis: Analysis K-SVD$^8$, NAAOL$^9$, GOAL$^{10}$.

Algorithms typically computationally expensive.

Algorithms may not converge.

$^3$ [Engan et al. '99], $^4$ [Aharon et al. '06], $^5$ [Mairal et al. '09], $^6$ [Ophir et al. '11], $^7$ [Yaghoobi et al. '11], $^8$ [Rubinstein et al. '12], $^9$ [Yaghoobi et al. '12], $^{10}$ [Hawe et al. '13].
Unstructured Square Transform Learning
Square Transform Learning Formulation

(P1) \[ \min_{W,X} \left\{ \|WY - X\|^2_F + \lambda \left( \xi \|W\|^2_F - \log |\det W| \right) \right\} \]

s.t. \( \|X_i\|_0 \leq s \ \forall \ i \)

- \( Y = [Y_1 | Y_2 | \ldots | Y_N] \in \mathbb{R}^{n \times N} \): matrix of training signals.
- \( X = [X_1 | X_2 | \ldots | X_N] \in \mathbb{R}^{n \times N} \): matrix of sparse codes of \( Y_i \).

- **Sparsification error** - measures deviation of data in transform domain from perfect sparsity.

- \( \lambda, \xi > 0 \). The \( \log |\det W| \) restricts solution to full rank transforms, and avoids repeated rows.

- \( \|W\|^2_F \) keeps objective function bounded from below.

- (P1) is non-convex.
Properties of Formulation

\[
(P1) \quad \min_{W,X} \| WY - X \|_F^2 + \lambda \left( \xi \| W \|_F^2 - \log |\det W| \right)
\]

s.t. \( \| X_i \|_0 \leq s \ \forall \ i \)

- (P1) attains lower bound of objective if and only if \( \exists (\hat{W}, \hat{X}) \) with \( \hat{X} \) sparse such that \( \hat{W}Y = \hat{X} \), and the condition number \( \kappa(\hat{W}) = 1 \).
- (P1) favors both a low sparsification error and good conditioning.
- Minimizing the \( \lambda \left( \xi \| W \|_F^2 - \log |\det W| \right) \) penalty encourages reduction of condition number.
- \( \lambda \) enables complete control over \( \kappa \). The solution to (P1) is perfectly conditioned (\( \kappa = 1 \)) as \( \lambda \to \infty \).
- If \( w_i \) is the \( i^{th} \) row of \( W \), then \( \max_{i \neq j} \left| \frac{\| w_i \| - \| w_j \|}{\| w_i \|} \right| \leq \kappa(W) - 1 \).
Algorithm with Iterative Transform Update

- (P1) solved by alternating between updating $X$ and $W$.

- **Sparse Coding Step** solves for $X$ with fixed $W$.
  \[
  \min_{\hat{X}} \| WY - \hat{X} \|_F^2 \quad \text{s.t.} \quad \| X_i \|_0 \leq s \quad \forall \ i
  \] (1)

  - **Easy** problem: Solution $\hat{X}$ computed exactly by zeroing out all but the $s$ largest magnitude coefficients in each column of $WY$.

- **Transform Update Step** solves for $W$ with fixed $X$.
  \[
  \min_{W} \| WY - X \|_F^2 + \lambda \left( \xi \| W \|_F^2 - \log |\det W| \right)
  \] (2)

  - Solved using Non-linear Conjugate Gradients (NLCG)$^{11}$.

$^{11}$ [Ravishankar & Bresler, IEEE TSP, 2013].
Exact Transform Update

- **Transform Update Step:**

  $$\min_W \| W Y - X \|_F^2 + \lambda \left( \xi \| W \|_F^2 - \log |\det W| \right) \tag{3}$$

- **Closed-form solution:**

  $$\hat{W} = 0.5 U \left( \Sigma + (\Sigma^2 + 2\lambda I_n)^{\frac{1}{2}} \right) Q^T L^{-1} \tag{4}$$

  where $YY^T + \lambda \xi I_n = LL^T$, and $L^{-1} Y X^T$ has a full singular value decomposition (SVD) of $Q \Sigma U^T$.

- The solution is invariant to the specific choice of square root $L$.

- It is unique if and only if $L^{-1} Y X^T$ is non-singular.
Proposition 1

For $\xi = 0.5$, as $\lambda \to \infty$, the sparse coding and transform update solutions in (P1) coincide with the solutions obtained by employing alternating minimization on

$$\min_{W,X} \|WY - X\|_F^2 \quad s.t. \quad W^T W = I, \|X_i\|_0 \leq s \quad \forall \ i.$$  \hspace{1cm} (5)

Specifically, the sparse coding step for Problem (5) involves

$$\min_{X} \|WY - X\|_F^2 \quad s.t. \quad \|X_i\|_0 \leq s \quad \forall \ i.$$  \hspace{1cm} (6)

and the solution is $\hat{X}_i = H_s(WY_i) \quad \forall \ i$. Transform update involves

$$\max_{W} \text{tr} (WYX^T) \quad s.t. \quad W^T W = I.$$  \hspace{1cm} (7)

Let $YX^T = U\Sigma V^T$ be a full SVD. Then, an optimal $\hat{W}$ in (7) is $VU^T$. 
Define the barrier function

\[ \psi(X) = \begin{cases} 
0, & \|X_i\|_0 \leq s, \ \forall \ i \\
+\infty, & \text{else}
\end{cases} \]

(P1) is equivalent to the problem of minimizing \( g(W, X) \).

\[ g(W, X) \triangleq \|WY - X\|_F^2 + \lambda \xi \|W\|_F^2 - \lambda \log |\det W| + \psi(X) \] (8)

For \( h \in \mathbb{R}^p \), \( \phi_j(h) \) is the magnitude of the \( j^{th} \) largest element (magnitude-wise) of \( h \).

For \( B \in \mathbb{C}^{p \times q} \), \( \|B\|_\infty \triangleq \max_{i,j} |B_{ij}| \).
Convergence Guarantees

Theorem 1

For the sequence \( \{W^k, X^k\} \) generated by Algorithm A1 with initial \((W^0, X^0)\), we have

- \( \{g(W^k, X^k)\} \) converges to a finite value \( g^* = g^*(W^0, X^0) \).
- \( \{W^k, X^k\} \) is bounded, and any specific accumulation point \((W, X)\) is a fixed point of Algorithm A1 satisfying

\[
g(W + dW, X + \Delta X) \geq g(W, X) = g^* \tag{9}
\]

The condition holds for all sufficiently small \( dW \in \mathbb{R}^{n \times n} \) satisfying \( \|dW\|_F \leq \epsilon \) for some \( \epsilon = \epsilon(W) > 0 \), and all \( \Delta X \in R1 \cup R2 \).

R1. The half-space \( \text{tr} \{(WY - X)\Delta X^T\} \leq 0 \).

R2. The local region defined by

\[
\|\Delta X\|_\infty < \min_i \{\phi_s(WY_i) : \|WY_i\|_0 > s\}.
\]

Furthermore, if we have \( \|WY_i\|_0 \leq s \forall i \), then \( \Delta X \) can be arbitrary.
Global Convergence Guarantees

Corollary 1

For each initialization of Algorithm A1, the objective converges to a local minimum, and the iterates converge to an equivalence class of local minimizers.

Corollary 2

Algorithm A1 is globally convergent (i.e., from any Initialization) to the set of local minimizers of the non-convex objective $g(W, X)$. 
Cost per iteration of proposed algorithms: $O(Nn^2)$ for $N$ training signals and $W \in \mathbb{R}^{n \times n}$.

Synthesis/Analysis K-SVD cost per iteration: $O(Nn^3)$. Cost dominated by sparse coding.

For images, this is a reduction of computations in the order by $n$, corresponding to $\sqrt{n} \times \sqrt{n}$ patches.

Closed-form solution for transform update also provides speedup of about $J$ over NLCG, where $J$ is the number of NLCG steps.
Convergence for (P1) with Various Initializations

Barbara - $8 \times 8$ patches

Objective Function

Sparsification Error ($s = 11$)

Learnt $W$ - DCT Init

2D DCT

S. Ravishankar    Adaptive Sparse Models
Learnt transforms are better than analytical transforms

- Normalized Sparsification Error (NSE) measures the fraction of energy lost in sparse fitting with sparse code $X$.

$$\text{NSE} = \frac{\|WY - X\|^2_F}{\|WY\|^2_F}$$

$\text{NSE}(W) \approx 4.4\%$, $\text{NSE}(\text{DCT}) = 6.8\%$.

- Recovery PSNR (rPSNR) measures the error in recovering image as $\hat{Y} = W^{-1}X$.

$$\text{rPSNR} = \frac{255\sqrt{P}}{\|Y - W^{-1}X\|_F}$$

$P$ is $\#$ of image pixels.

- rPSNRs for the learnt $W$ about 1.7 dB better than for DCT.

- Varying $\lambda$ allows trade-off between NSE and $\kappa(W)$. rPSNR best at intermediate $\kappa$. 

S. Ravishankar
Adaptive Sparse Models
Transform learning (TL) provides better sparsification & recovery than DCT.

Adapted well-conditioned transforms perform better (upto 0.3 dB better recovery) than adapted orthonormal transforms.

Adapted transforms outperform Independent Component Analysis (ICA).
Estimate image $x \in \mathbb{R}^P$ from its noisy measurement $y = x + h$.

$R_j \in \mathbb{R}^{n \times P}$ extracts patches. $R_j y \approx$ noiseless $x_j$.

$u(x_1, x_2, \ldots, x_n)$ is a regularizer $\Rightarrow$ regularized inverse problem.

$\tau \propto \frac{1}{\sigma}$ with $\sigma$ being the noise level.

Denoised $x$ obtained by averaging $x_j$'s at their 2D locations.
Image Denoising with Transform Learning Regularizer

\[
\text{(P2)} \quad \min_{W, \{x_j\}, \{\alpha_j\}} \quad \sum_{j=1}^{M} \| Wx_j - \alpha_j \|_2^2 + \lambda \nu(W) + \tau \sum_{j=1}^{M} \| R_j y - x_j \|_2^2
\]

\[
s.t. \quad \| \alpha_j \|_0 \leq s_j \quad \forall j
\]

- \( R_j \in \mathbb{R}^{n \times P} \) extracts patches. \( R_j y \approx \text{noiseless} \ x_j, \ Wx_j \approx \alpha_j \).
- \( \alpha_j \in \mathbb{R}^n \) is transform sparse code of \( x_j \).
- \( \text{(P2)} \) is solved by an efficient alternating scheme that uses \textbf{closed-form updates}, and \( s_j \) are found adaptively.
- Denoised \( x \) obtained by averaging \( x_j \)'s at their 2D locations.
Image Denoising Example

Noisy Image
PSNR = 24.60 dB

64 × 64 \( W (\kappa = 1.3) \)
PSNR = 31.66 dB

64 × 256 Synthesis \( D \)
PSNR = 31.50 dB

- Closed-form updates-based denoising is better and 17× faster than overcomplete K-SVD denoising.
- Square K-SVD (PSNR = 31.14 dB) denoises worse, and is slower.
- Our denoising PSNR increases with patch size \( n \), while still providing speedups over K-SVD of lower \( n \).
Summary

- We proposed formulations for learning square sparsifying transforms.
- Proposed alternating algorithms
  - involve efficient optimal updates
  - converge globally to the set of local minimizers of objective
  - low computational cost
  - encourage well-conditioning
- Adapted transforms provide better representations than analytical ones.
- Adaptive transforms denoise comparably or better than learnt overcomplete synthesis dictionaries, but are faster.
- Future Work: 3D applications.
Transform Blind Compressed Sensing
Compressed Sensing (CS)

- CS enables accurate recovery of images from fewer measurements than # of unknowns or Nyquist sampling
  - Sparsity in transform domain or dictionary
  - Acquisition incoherent with transform
  - Reconstruction problem is hard

Reconstruction problem (NP-hard) -

$$\min_x \|Ax - y\|_2^2 + \lambda \|\Psi x\|_0$$  \hspace{1cm} (10)

- $x \in \mathbb{C}^P$ : signal/image as vector, $y \in \mathbb{C}^m$ : measurements.
- $A \in \mathbb{C}^{m \times P}$ : sensing matrix ($m < P$), $\Psi \in \mathbb{C}^{T \times P}$ : given transform.

- $\ell_2$ penalty may be replaced with alternative choices, depending on the imaging process and the statistics of measurement noise.
Synthesis-based Blind Compressed Sensing (BCS)

(P3) \[
\min_{x,D,B} \sum_{j=1}^{N} \| R_j x - D b_j \|_2^2 + \nu \| A x - y \|_2^2
\]

\[ s.t. \quad \| d_k \|_2 = 1 \forall k, \quad \| b_j \|_0 \leq s \forall j. \]

- \( B \in \mathbb{C}^{n \times N} \): matrix that has the sparse codes \( b_j \) as its columns.

- (P3) learns \( D \in \mathbb{C}^{n \times K} \), and reconstructs \( x \), from only undersampled \( y \) ⇒

  *dictionary adaptive to underlying image.*

- (P3) is NP-hard, non-convex even if \( \ell_0 \)-quasinorm relaxed to \( \ell_1 \).

- DLMRI\(^{12}\) solves (P3) for MRI and works better than non-adaptive CS.

- Synthesis BCS algorithms have no guarantees and are expensive.

---

\(^{12}\) [Ravishankar & Bresler ’11]
Transform-based Blind Compressed Sensing (BCS)

\[
(P4) \quad \min_{x,W,B} \sum_{j=1}^{N} \|WR_jx - b_j\|_2^2 + \nu \|Ax - y\|_2^2 + \lambda v(W)
\]

s.t. \(\sum_{j=1}^{N} \|b_j\|_0 \leq s, \quad \|x\|_2 \leq C.\)

- \((P4)\) learns \(W \in \mathbb{C}^{n \times n}\), and reconstructs \(x\), from only undersampled \(y \Rightarrow \text{transform adaptive to underlying image.}\)

- \(v(W) \triangleq -\log |\det W| + 0.5 \|W\|_F^2\) controls scaling and \(\kappa\) of \(W\).

- We set \(\lambda = \lambda_0 N\), with \(\lambda_0 > 0\) a constant.

- \(\|x\|_2 \leq C\) is an energy/range constraint. \(C > 0\).
Proposition 2

Let \( x \in \mathbb{C}^p \), and let \( y = Ax \) be CS measurements obtained with a sensing matrix \( A \in \mathbb{C}^{m \times p} \). Suppose

1. \( \|x\|_2 \leq C \)
2. \( W \in \mathbb{C}^{n \times n} \) is a unitary transform
3. \( \sum_{j=1}^{N} \|WR_jx\|_0 \leq s \)

Further, let \( B \) denote the matrix that has \( WR_jx \) as its columns. Then, \((x, W, B)\) is a global minimizer of Problem (P4), i.e., it is identifiable by solving (P4).

- Conditions for uniqueness of solution to (P4) an open question.

- Given minimizer \((x, W, B)\) of (P4), \((x, \Theta W, \Theta B)\) is another equivalent minimizer \( \forall \Theta \) s.t. \( \Theta^H\Theta = I \), \( \sum_j \|\Theta b_j\|_0 \leq s \).
\[ \text{(P5)} \quad \min_{x,W,B} \sum_{j=1}^{N} \| WR_j x - b_j \|_2^2 + \nu \| Ax - y \|_2^2 \]

\[ \text{s.t.} \quad W^H W = I, \quad \sum_{j=1}^{N} \| b_j \|_0 \leq s, \quad \| x \|_2 \leq C. \]

\[ \text{(P5)} \] is also a unitary synthesis dictionary-based BCS problem, with \( W^H \) the synthesis dictionary.

\[ \text{(P6)} \quad \min_{x,W,B} \sum_{j=1}^{N} \| WR_j x - b_j \|_2^2 + \nu \| Ax - y \|_2^2 + \lambda \nu(W) + \eta^2 \sum_{j=1}^{N} \| b_j \|_0 \]

\[ \text{s.t.} \quad \| x \|_2 \leq C. \]
(P4) solved by alternating between updating $W$, $B$, and $x$.

Alternate a few times between the $W$ and $B$ updates, before performing an image update.

**Sparse Coding Step** solves (P4) for $B$ with fixed $x$, $W$.

$$
\min_B \sum_{j=1}^{N} \|WR_jx - b_j\|_2^2 \quad s.t. \quad \sum_{j=1}^{N} \|b_j\|_0 \leq s. \quad (11)
$$

**Cheap Solution:** Let $Z \in \mathbb{C}^{n \times N}$ be the matrix with $WR_jx$ as its columns. Solution $\hat{B} = H_s(Z)$ computed exactly by zeroing out all but the $s$ largest magnitude coefficients in $Z$. 
Transform Update Step solves (P4) for $W$ with fixed $x, B$.

$$
\min_W \sum_{j=1}^{N} \|WR_jx - b_j\|_2^2 + 0.5\lambda \|W\|_F^2 - \lambda \log |\det W| \tag{12}
$$

Let $X \in \mathbb{C}^{n \times N}$ be the matrix with $R_jx$ as its columns.

Closed-form solution:

$$
\hat{W} = 0.5R \left( \Sigma + \left( \Sigma^2 + 2\lambda I \right)^{\frac{1}{2}} \right) V^H L^{-1} \tag{13}
$$

where $XX^H + 0.5\lambda I = LL^H$, and $L^{-1}XB^H$ has a full SVD of $V\Sigma R^H$.

Solution is unique if and only if $L^{-1}XB^H$ is non-singular.
**Image Update Step** solves (P4) for $x$ with fixed $W$, $B$.

\[
\min_x \sum_{j=1}^{N} \|WR_jx - b_j\|^2_2 + \nu \|Ax - y\|^2_2 \quad s.t. \quad \|x\|^2_2 \leq C .
\]  

(14)

Least squares problem with $\ell_2$ norm constraint.

Solution is unique as long as the set of overlapping patches cover all image pixels.

**Solve Least squares Lagrangian formulation:**

\[
\min_x \sum_{j=1}^{N} \|WR_jx - b_j\|^2_2 + \nu \|Ax - y\|^2_2 + \mu \left(\|x\|^2_2 - C \right) 
\]

(15)

The optimal multiplier $\hat{\mu} \in \mathbb{R}^+$ is the smallest real such that $\|\hat{x}\|^2_2 \leq C$. $\hat{\mu}$ can be found cheaply.
Define the barrier function $\psi(B)$ as

$$
\psi(B) = \begin{cases} 
0, & \sum_{j=1}^{N} \|b_j\|_0 \leq s \\
+\infty, & \text{else}
\end{cases}
$$

$\chi(x)$ is the barrier function corresponding to $\|x\|_2 \leq C$.

(P4) is equivalent to the problem of minimizing $g(W, B, x) = \sum_{j=1}^{N} \|WR_jx - b_j\|_2^2 + \nu \|Ax - y\|_2^2 + \lambda \nu(W) + \psi(B) + \chi(x)$.

For $H \in \mathbb{C}^{p \times q}$, $\rho_j(H)$ is the magnitude of the $j^{\text{th}}$ largest element (magnitude-wise) of $H$.

$X \in \mathbb{C}^{n \times N}$ denotes a matrix with $R_jx$, $1 \leq j \leq N$, as its columns.
Theorem 2

For the sequence \( \{W^t, B^t, x^t\} \) generated by the BCD Algorithm with initial \((W^0, B^0, x^0)\), we have

- \( \{g(W^t, B^t, x^t)\} \) converges to a finite \( g^* = g^*(W^0, B^0, x^0) \).
- \( \{W^t, B^t, x^t\} \) is bounded, and all its accumulation points are equivalent as they achieve the same value \( g^* \) of the objective.

Every accumulation point \((W, B, x)\) is a critical or stationary point of \( g \) satisfying the following partial global optimality conditions

\[
\begin{align*}
    x &\in \arg\min_{\tilde{x}} \ g(W, B, \tilde{x}) \\
    W &\in \arg\min_{\tilde{W}} \ g(\tilde{W}, B, x) \\
    B &\in \arg\min_{\tilde{B}} \ g(W, \tilde{B}, x)
\end{align*}
\]
Theorem 3

Each accumulation point \((W, B, x)\) of \(\{W^t, B^t, x^t\}\) also satisfies the following partial local optimality conditions

\[
g(W + dW, B + \Delta B, x) \geq g(W, B, x) = g^* \tag{19}
\]

\[
g(W, B + \Delta B, x + \tilde{\Delta}x) \geq g(W, B, x) = g^* \tag{20}
\]

The conditions each hold for all \(\tilde{\Delta}x \in \mathbb{C}^p\), and all \(dW \in \mathbb{C}^{n \times n}\) satisfying \(\|dW\|_F \leq \epsilon\) for some \(\epsilon = \epsilon(W) > 0\), and all \(\Delta B \in \mathbb{C}^{n \times N}\) in \(R1 \cup R2\).

R1. The half-space \(\text{Re} \left( \text{tr} \left\{ (WX - B)\Delta B^H \right\} \right) \leq 0\).

R2. The local region defined by \(\|\Delta B\|_\infty < \rho_s(WX)\).

Furthermore, if \(\|WX\|_0 \leq s\), then \(\Delta B\) can be arbitrary.
Corollary 3

For each initialization, the iterate sequence in the BCD algorithm converges to an equivalence class of critical points, that are also partial global/local minimizers.

Corollary 4

The BCD algorithm is globally convergent (i.e., from any Initialization) to a subset of the set of critical points of the non-convex BCS objective $g(W, B, x)$. The subset includes all $(W, B, x)$, that are at least partial global minimizers of $g$ with respect to each of $W$, $B$, and $x$, and partial local minimizers of $g$ with respect to $(W, B)$, and $(B, x)$. 
Cost per iteration of transform BCS: $O(n^2 NL)$
- $N$ overlapping patches of size $\sqrt{n} \times \sqrt{n}$, $W \in \mathbb{C}^{n \times n}$.
- $L$ : # inner alternations between transform update & sparse coding.

Cost per iteration of Synthesis BCS method DLMRI$^{13}$: $O(n^3 NJ)$.
- $D \in \mathbb{C}^{n \times K}$, $K \propto n$, sparsity $s \propto n$.
- $J$ : # of inner iterations of dictionary learning using K-SVD.

Transform BCS much cheaper as $n$ increases $\Rightarrow$ 3D or 4D imaging.

---

$^{13}$ [Ravishankar & Bresler ’11]
Application: Compressed Sensing MRI

- Data - samples in k-space of spatial Fourier transform of object, acquired sequentially in time.

- Acquisition rate limited by MR physics, physiological constraints on RF energy deposition.

- CSMRI enables accurate recovery of images from far fewer measurements than \# unknowns or Nyquist sampling.

- Two directions to improve CSMRI -
  - better sparse modeling - TLMRI
  - better choice of sampling pattern ($F_u$)

Fig. from Lustig et al. '07
TLMRI Convergence - 4x Undersampling \((s = 3.4\%)\)

Reference \(^{14}\)

Sampling mask

Objective

\[
\|x^t - x^{t-1}\|_2 \text{ vs. } t
\]

\(^{14}\)Data from Miki Lustig.
Convergence & Learning - 4x Undersampling ($s = 3.4\%$)

Zero-filling (28.94 dB)  
TLMRI (32.66 dB)

PSNR and HFEN $^{15}$

real (top), imaginary (bottom) parts of learnt $36 \times 36$ $W$

$^{15}$ [Ravishankar & Bresler ’11]
Comparison (PSNR & Runtime) to Recent Methods

<table>
<thead>
<tr>
<th>Sampling Scheme</th>
<th>Undersampling</th>
<th>Zero-filling</th>
<th>Sparse MRI</th>
<th>PBDWS</th>
<th>DLMRI</th>
<th>TLMRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D Random</td>
<td>4x</td>
<td>25.30</td>
<td>30.32</td>
<td>32.64</td>
<td>32.91</td>
<td>33.04</td>
</tr>
<tr>
<td></td>
<td>7x</td>
<td>25.33</td>
<td>27.34</td>
<td>31.31</td>
<td>31.46</td>
<td>31.81</td>
</tr>
<tr>
<td>Cartesian</td>
<td>4x</td>
<td>28.94</td>
<td>30.20</td>
<td>32.02</td>
<td>32.46</td>
<td>32.64</td>
</tr>
<tr>
<td></td>
<td>7x</td>
<td>27.87</td>
<td>25.53</td>
<td>30.09</td>
<td>30.72</td>
<td>31.04</td>
</tr>
<tr>
<td>Avg. Runtime (s)</td>
<td></td>
<td>251</td>
<td>794</td>
<td>2051</td>
<td></td>
<td>211</td>
</tr>
</tbody>
</table>

- TLMRI is up to 5.5 dB better than Sparse MRI\(^\text{16}\), that uses Wavelets + TV.
- TLMRI provides up to 1 dB improvement in PSNR over the PBDWS\(^\text{17}\) method, that uses redundant Wavelets and trained patch-based geometric directions.
- It is up to 0.35 dB better than DLMRI\(^\text{18}\), that learns 4x overcomplete dictionary.
- TLMRI is 10x than DLMRI, and 4x faster than the PBDWS method.

\(^\text{16}\) Lustig et al. '07  \(^\text{17}\) Ning et al. '13  \(^\text{18}\) Ravishankar & Bresler '11
Reconstruction Errors - Cartesian 7x Undersampling

Sparse MRI

PBDWS

DLMRI

TLMRI
We introduced a transform-based BCS framework.

Proposed BCS algorithms have a low computational cost.

We provided novel convergence guarantees for the algorithms.

For CS MRI, the proposed approach is better than leading image reconstruction methods, while being much faster.

Future work: uniqueness of solution in BCS; Convergence to global minimizer.
Online Learning for Big Data†

† This is a joint work with B. Wen (equal contributor).
**Prior work:** batch transform learning, where learning is done using all the training data simultaneously.

Big data ⇒ training set is very large ⇒ batch learning computationally expensive in time and memory.

In real-time applications, data arrives sequentially, and must be processed sequentially to limit latency.

Online learning enables sequential adaptation of the transform (and sparse codes or signal estimates)

- amenable to big data, and real-time applications.
- involves cheap computations and modest memory requirements.
Online Transform Learning

$z_t : \text{Learnt Transform/Sparse Codes/Signal Estimates}$
(P7) \( \left\{ \hat{W}_t, \hat{x}_t \right\} = \arg \min_{W, x_t} \frac{1}{t} \sum_{j=1}^{t} \left\{ \| W y_j - x_j \|_2^2 + \lambda_j v(W) \right\} \)

s.t. \( \| x_t \|_0 \leq s \), \( x_j = \hat{x}_j \), \( 1 \leq j \leq t - 1 \).

- \( \lambda_j = \lambda_0 \| y_j \|_2^2 \) \( \forall j \), with \( \lambda_0 > 0 \). \( v(W) \triangleq \| W \|_F^2 - \log |\det W| \).
- \( \lambda_0 \) controls the condition number and scaling of learnt \( \hat{W}_t \).
- At time \( t \), estimate of \( \{ y_j \} \) obtained as \( \left\{ \hat{W}_t^{-1} \hat{x}_j \right\} \) (decompression).
- For non-stationary data, use forgetting factor \( \rho \in [0, 1] \), to diminish the influence of old data.

\[ \frac{1}{t} \sum_{j=1}^{t} \rho^{t-j} \left\{ \| W y_j - x_j \|_2^2 + \lambda_j v(W) \right\} \] (21)
Mini-Batch Transform Learning

\[
\{ \hat{W}_J, \hat{X}_J \} = \arg \min_{W, X_J} \sum_{j=1}^{J} \left\{ \| WY_j - X_j \|_F^2 + \Lambda_j v(W) \right\}
\]

s.t. \[ \|X_{JM-M+i}\|_0 \leq s \ \forall \ i \in \{1, \ldots, M\} \ (P8) \]

- \( Y_J = [y_{JM-M+1} \ y_{JM-M+2} \ \ldots \ y_{JM}] \), with \( M \) : mini-batch size.

- \( X_J = [x_{JM-M+1} \ x_{JM-M+2} \ \ldots \ x_{JM}] \). \( \Lambda_j = \lambda_0 \|Y_j\|_F^2 \).

- **Mini-batch learning**
  - can provide reductions in operation count over online learning.
  - increased latency and memory requirements.
Online Adaptive Transform Denoising

\[
(P9) \min_{W, x_t} \frac{1}{t} \sum_{j=1}^{t} \left\{ \|Wy_j - x_j\|^2_2 + \lambda_j \nu(W) + \tau_j^2 \|x_j\|_0 \right\}
\]

- **Goal**: Given \( \{y_t\} \), with \( y_t = z_t + h_t \), and \( h_t \in \mathbb{R}^n \) the noise, find \( z_t \ \forall \ t \).

- \( \tau_j \propto \sigma \), with \( \sigma \) - noise level.

- Denoised signal is \( \hat{z}_t = \hat{W}_t^{-1} \hat{x}_t \) – computed efficiently in our algorithm.

- \( (P9) \) can be used for denoising images, or image sequences:
  - overlapping patches of the noisy image(s) denoised sequentially.
  - image estimated by averaging denoised patches at 2D locations.
Online Transform Learning Algorithm

- **Sparse Coding:** solve for $x_t$ in (P7) with fixed $W = \hat{W}_{t-1}$.

  \[
  \min_{x_t} \| Wy_t - x_t \|_2^2 \quad s.t. \quad \|x_t\|_0 \leq s
  \]  
  \[ (22) \]

- **Cheap Solution:** $\hat{x}_t = H_s(Wy_t)$.

- **Transform Update:** solves for $W$ in (P7) with $x_t = \hat{x}_t$.

  \[
  \min_W \frac{1}{t} \sum_{j=1}^{t} \left\{ \| Wy_j - x_j \|_2^2 + \lambda_j \left( \| W \|_F^2 - \log |\det W| \right) \right\}
  \]  
  \[ (23) \]

  \[
  \hat{W}_t = 0.5 R_t \left( \Sigma_t + \left( \Sigma_t^2 + 2 \beta_t I \right)^{\frac{1}{2}} \right) Q_t^T L_t^{-1}
  \]  
  \[ (24) \]

- $t^{-1} \sum_{j=1}^{t} \left( y_j y_j^T + \lambda_0 \| y_j \|_2^2 I \right) = L_t L_t^T$. **Perform rank-1 update.**

- $\beta_t = \lambda_0 t^{-1} \sum_{j=1}^{t} \| y_j \|_2^2$. $Q_t \Sigma_t R_t^T$ is full SVD of $L_t^{-1} \Theta_t = t^{-1} \sum_{j=1}^{t} L_t^{-1} y_j x_j^T$.

- $L_t^{-1} \Theta_t \approx (1 - t^{-1}) L_{t-1}^{-1} \Theta_{t-1} + t^{-1} L_t^{-1} y_t x_t^T \Rightarrow$ **rank-1 SVD update.**

- **No matrix-matrix products.** Approx. error bounded, and cheaply monitored.
Mini-Batch Transform Learning Algorithm

- **Sparse Coding:** solve for $X_J$ in (P8) with fixed $W = \hat{W}_{J-1}$.

$$\min_{X_J} \| W Y_J - X_J \|^2_F \quad s.t. \quad \| x_{JM-M+i} \|_0 \leq s \ \forall \ i \tag{25}$$

- **Cheap Solution:** $\hat{x}_{JM-M+i} = H_s(Wy_{JM-M+i}) \ \forall i \in \{1, .., M\}$.

- **Transform Update:** solves for $W$ in (P8) with fixed $\{X_j\}_{j=1}^J$.

$$\min_W \frac{1}{JM} \sum_{j=1}^J \left\{ \| W Y_j - X_j \|^2_F + \Lambda_j \left( \| W \|^2_F - \log |\det W| \right) \right\} \tag{26}$$

$$\hat{W}_J = 0.5 \tilde{R}_J \left( \tilde{\Sigma}_J + \left( \tilde{\Sigma}_J^2 + 2 \tilde{\beta}_J I \right)^{\frac{1}{2}} \right) \tilde{Q}_J^T \tilde{L}_J^{-1} \tag{27}$$

- $G_J \triangleq \frac{1}{JM} \sum_{j=1}^J \left( Y_J Y_J^T + \lambda_0 \| Y_J \|^2_F I \right) = \tilde{L}_J \tilde{L}_J^T$. $\tilde{\beta}_J = \frac{\lambda_0}{JM} \sum_{j=1}^J \| Y_j \|^2_F$.

- $\tilde{Q}_J \tilde{\Sigma}_J \tilde{R}_J^T$ is full SVD of $\frac{1}{JM} \tilde{L}_J^{-1} \Theta_J$ with $\Theta_J = \sum_{j=1}^J Y_J X_J^T$.

- $G_J$, $\tilde{\beta}_J$, $\Theta_J$ updated sequentially over time.

- For $M \ll n$, use rank-M updates. For $M \geq O(n)$, direct SVDs.
## Comparison of Computations, Memory, and Latency

<table>
<thead>
<tr>
<th>Properties</th>
<th>Online</th>
<th>Mini-batch</th>
<th>Batch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Small $M \ll n$</td>
<td>Large $M$</td>
</tr>
<tr>
<td>Computations per sample</td>
<td>$O(n^2 \log^2 n)$</td>
<td>$O(n^2 \log^2 n)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$O(n^2)$</td>
<td>$O(Pn^2)$</td>
</tr>
<tr>
<td>Memory</td>
<td>$O(n^2)$</td>
<td>$O(n^2)$</td>
<td>$O(nM)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$O(nM)$</td>
<td>$O(nN)$</td>
</tr>
<tr>
<td>Latency</td>
<td>0</td>
<td>$M - 1$</td>
<td>$N - 1$</td>
</tr>
</tbody>
</table>

- **Latency** is measured as the max. time between arrival of a signal, and generation of corresponding output.

- $P$ is # of batch iterations. $N$ is the total number of samples.

- $\log^2 n < P \Rightarrow$ online scheme is computationally cheaper than batch.

- For big data, online & mini-batch schemes have low memory & latency costs.

- **Online synthesis learning**\(^\text{19}\) has high computational cost per sample: $O(n^3)$.

---

\(^{19}\) [Mairal et al. '10]
Convergence Behavior

- \{y_t\} generated as \{W^{-1}x_t\} with random unitary 20 × 20 \textit{W}, and random \textit{x}_t with \|\textit{x}_t\|_0 = 3.

- The exact and approximate online schemes behave identically.

- Sparsification error converges to zero, and \(\kappa(\textit{W}) \in [1.02, 1.04]\) for the schemes ⇒ learnt a good model.
### Image Denoising – PSNR (dB) and runtime (sec)

<table>
<thead>
<tr>
<th>Images</th>
<th>$\sigma$</th>
<th>Noisy PSNR</th>
<th>Batch K-SVD</th>
<th>Batch TL</th>
<th>Mini-batch TL ($M = 64$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Couple (512 × 512)</td>
<td>5</td>
<td>34.16</td>
<td>37.28</td>
<td>37.33</td>
<td>37.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>time 1250</td>
<td>92</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>28.11</td>
<td>33.51</td>
<td>33.62</td>
<td>33.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>time 671</td>
<td>68</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>22.11</td>
<td>30.02</td>
<td>30.02</td>
<td>30.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>time 190</td>
<td>61</td>
<td>20</td>
</tr>
<tr>
<td>Man (768 × 768)</td>
<td>5</td>
<td>34.15</td>
<td>36.47</td>
<td>36.66</td>
<td>36.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>time 1279</td>
<td>205</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>28.13</td>
<td>32.71</td>
<td>32.96</td>
<td>33.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>time 701</td>
<td>130</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>22.11</td>
<td>29.40</td>
<td>29.57</td>
<td>29.52</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>time 189</td>
<td>80</td>
<td>41</td>
</tr>
</tbody>
</table>

- Overlapping 8 × 8 patches are denoised sequentially with a forgetting factor. We observed better denoising with a forgetting factor.

- Mini-batch denoising is better and provides average speedup of 26.0× and 3.4× over the batch K-SVD and batch transform denoising schemes.
Big Image Denoising - Data

Airport (1024 × 1024)

Man (1024 × 1024)

Railway (2048 × 2048)

Campus (3264 × 3264 × 3)
## Big Image Denoising – PSNR (dB) and runtime (sec)

<table>
<thead>
<tr>
<th>Images</th>
<th>Methods</th>
<th>(\sigma = 20) (22.11)</th>
<th>(\sigma = 50) (14.15)</th>
<th>(\sigma = 100) (8.13)</th>
<th>Run Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport</td>
<td>DCT</td>
<td>28.79</td>
<td>24.65</td>
<td>21.00</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>TL</td>
<td><strong>28.83</strong></td>
<td><strong>25.07</strong></td>
<td><strong>22.53</strong></td>
<td>28</td>
</tr>
<tr>
<td>Man</td>
<td>DCT</td>
<td>30.44</td>
<td>25.80</td>
<td>21.87</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>TL</td>
<td><strong>30.64</strong></td>
<td><strong>26.62</strong></td>
<td><strong>23.88</strong></td>
<td>27</td>
</tr>
<tr>
<td>Railway</td>
<td>DCT</td>
<td>31.90</td>
<td>26.44</td>
<td>22.04</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>TL</td>
<td><strong>32.42</strong></td>
<td><strong>27.58</strong></td>
<td><strong>24.35</strong></td>
<td>111</td>
</tr>
<tr>
<td>Campus</td>
<td>DCT</td>
<td>30.89</td>
<td>25.88</td>
<td>21.99</td>
<td>323</td>
</tr>
<tr>
<td></td>
<td>TL</td>
<td><strong>33.10</strong></td>
<td><strong>27.47</strong></td>
<td><strong>23.24</strong></td>
<td>451</td>
</tr>
</tbody>
</table>

- Adaptive mini-batch denoising (TL) with \(M = 256\) performs better than the DCT, without much loss in runtime.

- Results demonstrate the potential of our schemes for real-time denoising of large-scale data.
Online Transform Learning: Convergence Analysis
Mairal et al. (2010) analyzed the convergence of online synthesis dictionary learning (OSDL).

The OSDL formulation is biconvex.

The convergence result for OSDL involves various restrictive assumptions.

Our problem formulations are not biconvex
- due to $\ell_0$ quasi norm for sparsity.
- due to log-determinant penalty for $W$.

Our convergence analysis relies on simpler (easy to verify) assumptions.
For (P7), we perform sparse coding as $\hat{x}_t = H_s \left( \hat{W}_{t-1} y_t \right)$.

The objective of the transform update step is

$$\hat{g}_t(W) = \frac{1}{t} \sum_{j=1}^{t} \left\{ \| W y_j - \hat{x}_j \|_2^2 + \lambda_0 \| y_j \|_2^2 v(W) \right\}$$  \hspace{1cm} (28)

The empirical objective function is

$$g_t(W) = \frac{1}{t} \sum_{j=1}^{t} \left\{ \| W y_j - H_s(W y_j) \|_2^2 + \lambda_0 \| y_j \|_2^2 v(W) \right\}$$  \hspace{1cm} (29)

This is the objective that is minimized in batch transform learning.

In the online setting, the sparse codes of past signals cannot be optimally set at future times $t$. 
(A1) We assume that the signals $y_t$ satisfy $\|y_t\|_2 = 1 \ \forall \ t$.

(A2) The transform update step of our algorithm(s) is assumed to be performed exactly.

- Although in practice the SVD is computed using iterative methods, these are guaranteed to quickly provide machine precision accuracy.

(A3) Assume $L_t^{-1} \Theta_t = t^{-1} \sum_{j=1}^{t} L_t^{-1} y_j x_j^T$ has non-degenerate (distinct, non zero) singular values $\forall \ t$.

- This is not required to establish the convergence of the objective $\hat{g}_t$.

(A4) $y_t$ are i.i.d. random samples from the sphere $S^n = \{y \in \mathbb{R}^n : \|y\|_2 = 1\}$, assuming absolutely continuous probability measure $p$. 

Convergence Analysis: Assumptions
We follow the standard approach in the analysis of online algorithms and consider the minimization of

\[ g(W) = \mathbb{E}_y \left[ \| Wy - H_s(Wy) \|_2^2 + \lambda_0 \| y \|_2^2 \nu(W) \right] \]  \hspace{1cm} (30)

It follows from Assumption A4 that \( \lim_{t \to \infty} g_t(W) = g(W) \) a.s. (almost sure convergence).

Given a specific training set, it is unnecessary to minimize the batch objective \( g_t(W) \) to high precision, since \( g_t(W) \) only approximates \( g(W) \).

Even an inaccurate minimizer of \( g_t(W) \) could provide the same, or better value of \( g(W) \) than a fully optimized one.
Main Convergence Results for (P1)

Theorem 4

For the sequence \( \{ \hat{W}_t \} \) generated by our online scheme, we have

(i) As \( t \to \infty \), \( \hat{g}_t(\hat{W}_t) \), \( g_t(\hat{W}_t) \), and \( g(\hat{W}_t) \) converge almost surely to a common limit, say \( g^* \).

(ii) The sequence \( \{ \hat{W}_t \} \) is bounded. Every accumulation \( \hat{W}_\infty \) of \( \{ \hat{W}_t \} \) is a stationary point of the expected cost \( g(W) \) satisfying \( \nabla g(\hat{W}_\infty) = 0 \).

(iii) All accumulation points of \( \{ \hat{W}_t \} \) achieve the same value \( g^* \) of the objective \( g(W) \) with probability 1.

Corollary 5

The sequence \( \{ \hat{W}_t \} \) converges to the set of stationary points of \( g(W) \).
Theorem 5

For the sequence \( \{ \hat{W}_t \} \) generated by our online scheme, we have

(i) \( \hat{g}_{t+1}(\hat{W}_{t+1}) - \hat{g}_t(\hat{W}_t) \) decays as \( O(1/t) \).

(ii) \( \hat{g}_{t+1}(\hat{W}_{t+1}) - \hat{g}_{t+1}(\hat{W}_t) \) decays as \( O(1/t^2) \).

(iii) \( \hat{W}_{t+1} - \hat{W}_t \) decays (in norm) as \( O(1/t) \).
We introduced an online sparsifying transform learning framework.

Proposed methods are particularly useful for big data & real-time applications.

Iterates converge to the set of stationary points of the expected transform learning cost.

The online schemes perform well and are highly efficient for sparse representation & denoising.

Future work: video denoising, online blind compressed sensing.
Union of Transforms or OCTOBOS†

† This is a joint work with B. Wen (equal contributor).
Why Union of Transforms (UOT)?

- Natural images typically have diverse textures.
Why Union of Transforms (UOT)?

- Union of transforms: one for each class of textures or features.

[Image of textures and a person with highlighted areas]
Block Cosparisity

For $x \in \mathbb{R}^{Kn}$, $\|x\|_{0,s} \triangleq \sum_{k=1}^{K} I \left( \|x^k\|_0 \leq s \right)$.

- $x^k \in \mathbb{R}^n$ is $k^{th}$ block of $x$, $1 \leq k \leq K$.
- $I(\cdot)$ is the indicator function.

- $\|x\|_{0,s}$ counts the number of blocks of $x$ with at least $n - s$ zeros.

$n = 4$, $\|x\|_{0,2} = 1$

2 Zeros in Second block.
UOT $\equiv$ OCTOBOS

**UOT Model:**

Signal $y$ is matched to the best transform in the set $\{W_k\}_{k=1}^K$.

\[(P10) \quad \min_{1 \leq k \leq K} \min_{z^k} \|W_k y - z^k\|_2^2 \quad s.t. \quad \|z^k\|_0 \leq s \quad \forall \ k\]

**OCTOBOS Model:**

- Overcomplete $W \triangleq [W_1^T | W_1^T | \ldots | W_K^T]^T \in \mathbb{R}^{m \times n}$, $m = Kn$.
- $Wy = x + e$, with $x$ that is block cosparse.

\[(P11) \quad \min_x \|Wy - x\|_2^2 \quad s.t. \quad \|x\|_{0,s} \geq 1\]

**OCTOBOS and UOT models are equivalent!**
\[(P12) \quad \min_{\{W_k, X_i, C_k\}} \sum_{k=1}^{K} \sum_{i \in C_k} \|W_k Y_i - X_i\|_2^2 + \sum_{k=1}^{K} \lambda_k \left(\|W_k\|_F^2 - \log |\det W_k|\right) \]

\[\text{s.t.} \quad \|X_i\|_0 \leq s \quad \forall \ i, \quad \{C_k\}_{k=1}^{K} \in G\]

- $G$ is the set of all partitions of $[1 : N]$ into $K$ disjoint subsets $\{C_k\}_{k=1}^{K}$.
- $(P12)$ jointly learns the union-of-transforms $\{W_k\}$ and clusters the data $Y$.
- Regularizer necessary to control scaling and conditioning ($\kappa$) of transforms.
  - $\lambda_k = \lambda_0 \|Y_{C_k}\|_F^2$, with $Y_{C_k}$ the matrix of all $Y_i \in C_k$, achieves **scale invariance** of the solution in $(P12)$.
  - As $\lambda_0 \to \infty$, $\kappa(W_k) \to 1$ and $\|W_k\|_2 \to \frac{1}{\sqrt{2}} \forall k$. 

---

S. Ravishankar  
Adaptive Sparse Models
**Transform Update:** Solves for only the \( \{W_k\} \) in (P12).

\[
\min_{\{W_k\}} \sum_{k=1}^{K} \left\{ \sum_{i \in C_k} \| W_k Y_i - X_i \|_2^2 + \lambda_k v(W_k) \right\} \tag{31}
\]

**Closed-form solution** using Singular Value Decomposition (SVD):

\[
\hat{W}_k = 0.5 R_k (\Sigma_k + (\Sigma_k^2 + 2\lambda_k I)^{1/2}) V_k^T L_k^{-1}, \quad \forall k
\tag{32}
\]

- \( I \) is the identity matrix. \( \lambda_k = \lambda_0 \| Y_{C_k} \|_F^2 \).
- \( Y_{C_k} Y_{C_k}^T + \lambda_k I = L_k L_k^T \). \( L_k \) is a matrix square root.
- \( L_k^{-1} Y_{C_k} X_{C_k}^T = V_k \Sigma_k U_k^T \) by Full SVD.
Alternating OCTOBOS Learning Algorithm: Step 2

- **Sparse Coding & Clustering:** Solves for only the \( \{C_k, X_i\} \) in (P12).

\[
\begin{align*}
\min_{\{C_k\}, \{X_i\}} & \quad \sum_{k=1}^{K} \sum_{i \in C_k} \left\{ \|W_k Y_i - X_i\|_2^2 + \lambda_0 \|Y_i\|_2^2 v(W_k) \right\} \\
\text{s.t.} & \quad \|X_i\|_0 \leq s \quad \forall \ i, \quad \{C_k\} \in G
\end{align*}
\] (33)

- **Exact Clustering:** finds the global optimum \( \{\hat{C}_k\} \) in (33) as

\[
\{\hat{C}_k\} = \arg \min_{\{C_k\}} \sum_{k=1}^{K} \sum_{i \in C_k} \left\{ \|W_k Y_i - H_s(W_k Y_i)\|_2^2 + \lambda_0 \|Y_i\|_2^2 v(W_k) \right\}
\] (34)

- For each \( Y_i \), the optimal cluster index \( \hat{k}_i = \arg \min_k M_{k,i} \).

- **Exact and Cheap Sparse Coding:** \( \hat{X}_i = H_s(W_k Y_i) \) \( \forall i \in \hat{C}_k, \forall k \).
Cost per-iteration for learning $W \in \mathbb{R}^{Kn \times n}$ OCTOBOS:

- Assume the number of training signals $N \gg m = Kn$.
- Cost of Clustering & Sparse coding Step: $O(mnN)$.
- Cost of Transform Update Step: $O(n^2N)$.
- Total computational cost per-iteration dominated by clustering.

<table>
<thead>
<tr>
<th>Models ($s \propto n$)</th>
<th>Square $W \in \mathbb{R}^{n \times n}$</th>
<th>OCTOBOS $W \in \mathbb{R}^{m \times n}$</th>
<th>KSVD $D \in \mathbb{R}^{n \times m}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per-iter. Cost</td>
<td>$O(n^2N)$</td>
<td>$O(mnN)$</td>
<td>$O(mn^2N)$</td>
</tr>
</tbody>
</table>

In practice, OCTOBOS learning algorithm converges in few iterations.

OCTOBOS learning is cheaper than dictionary learning by K-SVD\textsuperscript{20}.

\textsuperscript{20} [Aharon et al. ’06]
Global Convergence Guarantees for OCTOBOS

\[
\begin{align*}
\text{(P12)} \quad & \min_{\{W_k, X_i, C_k\}} \sum_{k=1}^{K} \sum_{i \in C_k} \|W_k Y_i - X_i\|_2^2 + \sum_{k=1}^{K} \lambda_k \left(\|W_k\|_F^2 - \log |\det W_k|\right) \\
& \text{s.t. } \|X_i\|_0 \leq s \ \forall \ i, \ \{C_k\}_{k=1}^{K} \in G
\end{align*}
\]

- The alternating OCTOBOS learning algorithm is globally convergent to the set of partial minimizers of the objective in (P12).
- These partial minimizers are global minimizers w.r.t. \(\{W_k\}\) and \(\{X_i, C_k\}\), respectively, and local minimizers w.r.t. \(\{W_k, X_i\}\).
- Under certain (mild) conditions, the algorithm converges to the set of stationary points of the equivalent objective \(f(W)\).

\[
f(W) \triangleq \sum_{i=1}^{N} \min_k \left\{\|W_k Y_i - H_s(W_k Y_i)\|_2^2 + \lambda_0 \nu(W_k) \|Y_i\|_2^2\right\}
\]
Convergence for Various Initializations ($s = 11$)

8 × 8 patches, $K = 2$

Various initializations for $\{W_k\}$

Various initializations for $\{C_k\}$ and $K = 1$ (single) case
The square blocks of a learnt OCTOBOS are dissimilar $\Rightarrow$ cluster-specific $W_k$.

- OCTOBOS $W$ learnt with different initializations can appear different.
- The $W$ learnt with different initializations sparsify equally well.

Cross-gram matrix between $W_1$ and $W_2$ for KLT Init.

Random matrix Init.  KLT Init.
Unsupervised Classification by OCTOBOS

- Overlapping image patches are clustered by learnt OCTOBOS.
- Each pixel is then classified by a vote among the patches that cover it.

Original Image  
K-means  
OCTOBOS

Original Image  
K-means  
OCTOBOS
Goal: Estimate image $x \in \mathbb{R}^P$ from its noisy measurement $y = x + h$.

- $R_j \in \mathbb{R}^{n \times P}$ extracts patches. $R_j y \approx$ noiseless $x_j$, $W_k x_j \approx \alpha_j$, $\forall j \in C_k$, $\forall k$.

- $\alpha_j \in \mathbb{R}^n$ is sparse code of $x_j$; $\tau \propto \frac{1}{\sigma}$, where $\sigma$ is the noise level.

- (P13) is solved by a simple iterative scheme that uses closed-form updates, and the $s_j$'s are found adaptively.

- Denoised image $x$ obtained by averaging the $x_j$'s at their 2D locations.

- (P13) is re-solved by replacing $y$ with successively denoised images for further denoising.
OCTOBOS denoises 0.36 dB better than K-SVD\(^{21}\) on avg., and is faster.

OCTOBOS also denoises 0.43 dB better than GMM\(^{22}\) on average here.

Its performance is comparable to BM3D\(^{23}\) in some cases.

\[\text{\cite{Elad2006, Zoran2011, Dabov2007}}\]

\[\begin{array}{|c|c|c|c|c|c|}
\hline
\text{Image} & \sigma & \text{Noisy PSNR} & \text{BM3D} & \text{K-SVD} & \text{GMM} & \text{OCTOBOS} \\
\hline
\text{Cameraman} & 5 & 34.12 & 38.21 & 37.81 & 38.06 & 38.19 \\
 & 10 & 28.14 & 34.15 & 33.72 & 34.00 & 34.15 \\
 & 15 & 24.61 & 31.91 & 31.50 & 31.85 & 31.94 \\
 & 20 & 22.10 & 30.37 & 29.82 & 30.21 & 30.24 \\
 & 100 & 8.14 & 23.15 & 21.76 & 22.89 & 22.24 \\
\hline
\text{Barbara} & 5 & 34.15 & 38.30 & 38.08 & 37.59 & 38.31 \\
 & 10 & 28.14 & 34.97 & 34.41 & 33.61 & 34.64 \\
 & 15 & 24.59 & 33.05 & 32.33 & 31.28 & 32.53 \\
 & 20 & 22.13 & 31.74 & 30.83 & 29.74 & 31.05 \\
 & 100 & 8.11 & 23.61 & 21.87 & 22.13 & 22.41 \\
\hline
\end{array}\]
**Image Denoising - Effect of Overcompleteness (K)**

![Graph](image)

**PSNR for Barbara at $\sigma = 10$**

**PSNR for Barbara at $\sigma = 20$**

- **OCTOBOS** denoises up to 0.4 dB better than the square transform here.
- Best choice of $K$ (number of clusters) lower at higher $\sigma$. 

---

---
Summary

- We proposed learning Union-of-Transforms or OCTOBOS.
- Proposed algorithms have global convergence guarantees.
- Algorithms are cheap and perform well in applications.

Future Work

- Combination of OCTOBOS and non-local methods in denoising.
- Further exploration of OCTOBOS classification.
- Comparison to unstructured overcomplete transform learning.
Overall Conclusions and Future Directions

- We proposed several methods for learning square or overcomplete, structured or unstructured sparsifying transforms.

- Proposed algorithms typically
  - encourage well-conditioning
  - have low computational cost
  - have convergence guarantees

- Adaptive transforms
  - provide better sparse representations than analytical ones.
  - denoise better than learnt overcomplete synthesis dictionaries.
  - are useful for compressed sensing, classification, Big Data.

- Future Work: Analyze blind denoising or compressed sensing further.
Acknowledgments

- Advisor: Yoram Bresler
- Committee: Jeff Fessler, Olgica Milenkovic, Brad Sutton, Minh Do
- Collaborators: Bihan Wen
- Staff: Peggy Wells
- Funding: National Science Foundation (NSF)


S. Ravishankar and Y. Bresler, “\( \ell_0 \) sparsifying transform learning with efficient optimal updates and convergence guarantees,” IEEE TSP, 2014, accepted with minor revision.


S. Ravishankar and Y. Bresler, “Fast doubly sparse transform learning with convergence guarantees,” 2014, manuscript to be submitted.
List of Publications - Conferences


Thank you! Questions??
Convergence Guarantees - Definitions

**Definition 1**

Let $\phi : \mathbb{R}^q \mapsto (-\infty, +\infty]$ be a proper function and let $z \in \text{dom}\phi$. The Fréchet sub-differential of the function $\phi$ at $z$ is the following set:

$$
\hat{\partial}\phi(z) \triangleq \left\{ h \in \mathbb{R}^q : \liminf_{b \to z, b \neq z} \frac{1}{\|b-z\|} (\phi(b) - \phi(z) - \langle b - z, h \rangle) \geq 0 \right\} \quad (35)
$$

If $z \notin \text{dom}\phi$, then $\hat{\partial}\phi(z) = \emptyset$. The sub-differential of $\phi$ at $z$ is defined as

$$
\partial\phi(z) \triangleq \left\{ \tilde{h} \in \mathbb{R}^q : \exists z_k \to z, \phi(z_k) \to \phi(z), h_k \in \hat{\partial}\phi(z_k) \to \tilde{h} \right\} . \quad (36)
$$

**Lemma 1**

A necessary condition for $z \in \mathbb{R}^q$ to be a minimizer of the function $\phi : \mathbb{R}^q \mapsto (-\infty, +\infty]$ is that $z$ is a critical point of $\phi$, i.e., $0 \in \partial\phi(z)$. If $\phi$ is a convex function, this condition is also sufficient.