Efficient Data-Driven Learning of Sparse Signal Models and Its Applications

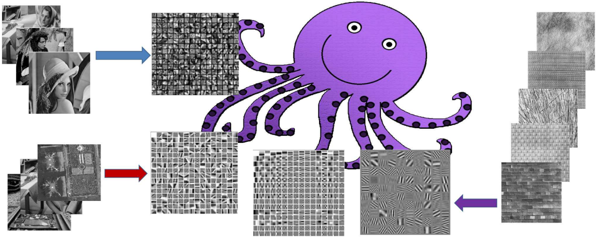
Saiprasad Ravishankar

Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor

Dec 10, 2015

Outline of Talk

- Synthesis & Transform models.
- **Transform learning: Efficient, Scalable, Effective, Guarantees.**
- **Transform learning methods:**
  - Union of transforms (OCTOBOS) learning
  - Online transform learning for big data

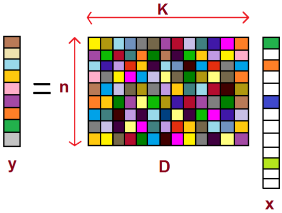


- Applications: Compression, Denoising, Compressed sensing, Classification.
- Conclusions

Fig. from B. Wen, UIUC.

Synthesis Model 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 $\|x\|_0 \ll K$.



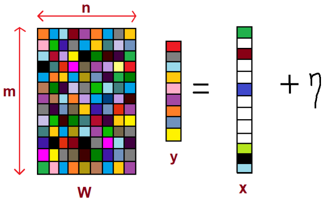
- Real world signals modeled as $y = Dx + e$, e is deviation term.
- Given D, sparsity level s, the *synthesis sparse coding* problem is

$$\hat{x} = \arg\min_{x} \|y - Dx\|_2^2 \;\; s.t. \;\; \|x\|_0 \leq s$$

- This problem is NP-hard.
- Dictionary-based approaches are often computationally expensive.

Transform Model 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 η - error term.



- Natural signals are approximately sparse in Wavelets, DCT.
- Given W, and sparsity s, *transform sparse coding* is

$$\hat{x} = \arg\min_{x} \|Wy - x\|_2^2 \;\; s.t. \;\; \|x\|_0 \leq s$$

- $\hat{x} = H_s(Wy)$ computed by thresholding Wy to the s largest magnitude elements. **Sparse coding is cheap!** Signal recovered as $W^{\dagger}\hat{x}$.
- Sparsifying transforms exploited for compression (JPEG2000), etc.

Key Topic of Talk: Sparsifying Transform Learning

- **Square Transform Models**
  - Unstructured transform learning [IEEE TSP, 2013 & 2015]
  - Doubly sparse transform learning [IEEE TIP, 2013]
  - Online learning for Big Data [IEEE JSTSP, 2015]
  - Convex formulations for transform learning [ICASSP, 2014]
- **Overcomplete Transform Models**
  - Unstructured overcomplete transform learning [ICASSP, 2013]
  - Learning structured overcomplete transforms with block cosparsity (OCTOBOS) [IJCV, 2015]
- **Applications:** Image & Video denoising, Classification, Magnetic resonance imaging (MRI) [SPIE 2015, ICIP 2015].

Square Transform Learning Formulation[1]

$$(\text{P0}) \quad \min_{W,X} \; \overbrace{\|WY - X\|_F^2}^{\text{Sparsification Error}} + \lambda \overbrace{\left(\|W\|_F^2 - \log|\det W| \right)}^{\text{Regularizer} \triangleq v(W)}$$

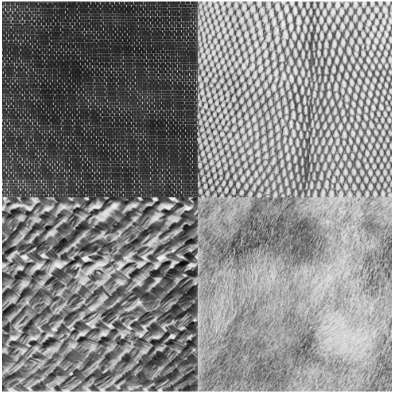
$$s.t. \;\; \|X_i\|_0 \leq s \;\; \forall \, i$$

- $Y = [\, Y_1 \,|\, Y_2 \,|\, ... \,|\, Y_N \,] \in \mathbb{R}^{n \times N}$: matrix of training signals.
- $X = [\, X_1 \,|\, X_2 \,|\, ... \,|\, X_N \,] \in \mathbb{R}^{n \times N}$: matrix of sparse codes of Y_i.
- Sparsification error - measures deviation of data in transform domain from perfect sparsity.
- $\lambda > 0$. Regularizer $v(W)$ prevents trivial solutions and fully controls condition number of W.
- (P0) is limited due to a single W for all the data.

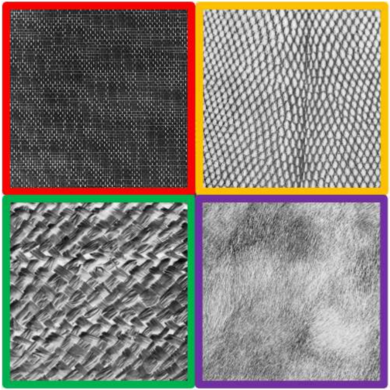
[1] [Ravishankar & Bresler '12]

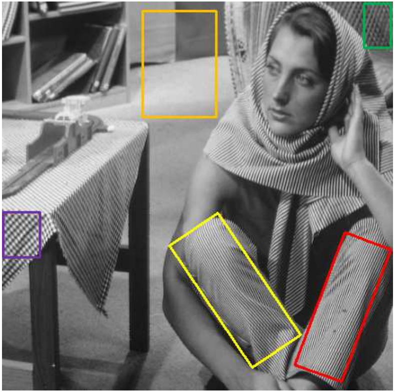
Why Union of Transforms?

- Natural images typically have diverse features or textures.



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





OCTOBOS Learning Formulation

$$(\text{P1}) \quad \min_{\{W_k, X_i, C_k\}} \overbrace{\sum_{k=1}^{K} \sum_{i \in C_k} \|W_k Y_i - X_i\|_2^2}^{\text{Sparsification Error}} + \overbrace{\sum_{k=1}^{K} \lambda_k \left(\|W_k\|_F^2 - \log |\det W_k| \right)}^{\text{Regularizer}}$$

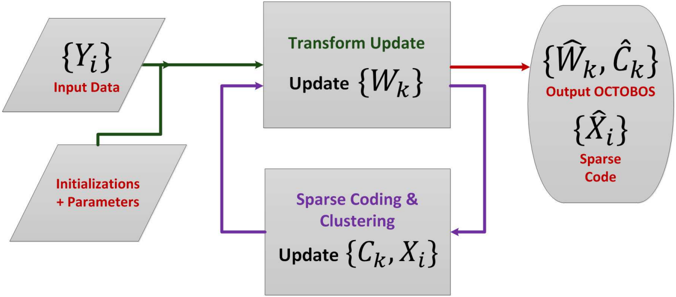
$$s.t. \quad \|X_i\|_0 \leq s \;\; \forall\, i, \quad \{C_k\}_{k=1}^{K} \in G$$

- C_k is the set of indices of signals in class k.
- G is the set of all possible partitions of $[1 : N]$ into K disjoint subsets.
- (P1) jointly learns the union-of-transforms $\{W_k\}$ and clusters the data Y.
- Regularizer necessary to control scaling and conditioning (κ) 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 (P1).
  - As $\lambda_0 \to \infty$, $\kappa(W_k) \to 1$, $\|W_k\|_2 \to 1/\sqrt{2} \; \forall\, k$ for solutions in (P1).

Alternating Minimization Algorithm for (P1)

$$\text{(P1)} \quad \min_{\{W_k, X_i, C_k\}} \overbrace{\sum_{k=1}^{K} \sum_{i \in C_k} \|W_k Y_i - X_i\|_2^2}^{\text{Sparsification Error}} + \overbrace{\sum_{k=1}^{K} \lambda_k \left(\|W_k\|_F^2 - \log |\det W_k| \right)}^{\text{Regularizer}}$$

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



Alternating OCTOBOS Learning Algorithm: Step 1

- **Transform Update:** Solves for only the $\{W_k\}$ in (P1).

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

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

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

  - I is the identity matrix. $\lambda_k = \lambda_0 \left\| Y_{C_k} \right\|_F^2$.
  - $Y_{C_k} Y_{C_k}^T + \lambda_k I = L_k L_k^T$. L_k is a matrix square root.
  - SVD: $L_k^{-1} Y_{C_k} X_{C_k}^T = V_k \Sigma_k U_k^T$.

Alternating OCTOBOS Learning Algorithm: Step 2

- **Sparse Coding & Clustering:** Solves for only the $\{C_k, X_i\}$ in (P1).

$$\min_{\{C_k\},\{X_i\}} \sum_{k=1}^{K} \sum_{i \in C_k} \left\{ \|W_k Y_i - X_i\|_2^2 + \lambda_0 \|Y_i\|_2^2 \, v(W_k) \right\} \tag{3}$$
$$s.t. \;\; \|X_i\|_0 \leq s \;\; \forall \, i, \quad \{C_k\} \in G$$

- **Exact Clustering:** finds the global optimum $\{\hat{C}_k\}$ in (3) as

$$\left\{\hat{C}_k\right\} = \arg\min_{\{C_k\}} \sum_{k=1}^{K} \sum_{i \in C_k} \overbrace{\left\{ \|W_k Y_i - H_s(W_k Y_i)\|_2^2 + \lambda_0 \|Y_i\|_2^2 \, v(W_k) \right\}}^{\text{Clustering Measure} \triangleq M_{k,i}} \tag{4}$$

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

- **Exact and Cheap Sparse Coding:** $\hat{X}_i = H_s(W_k Y_i) \;\; \forall i \in \hat{C}_k, \forall k.$

Computational Advantages of OCTOBOS

- **Cost per-iteration for learning OCTOBOS $\mathbf{W} \in \mathbb{R}^{Kn \times n}$:**
  - Assume number of training signals $N \gg m = Kn$.
  - Cost of Clustering & Sparse coding Step: $O(mnN)$.
  - Cost of Transform Update Step: $O(n^2 N)$.
  - Cost dominated by clustering.

Model $(m, s \propto n)$	Square $W \in \mathbb{R}^{n \times n}$	OCTOBOS $\mathbf{W} \in \mathbb{R}^{m \times n}$	KSVD $D \in \mathbb{R}^{n \times m}$
Per-iter. Cost	$O(n^2 N)$	$O(n^2 N)$	$O(n^3 N)$

- In practice, OCTOBOS learning converges in few iterations.
- OCTOBOS learning is cheaper than dictionary learning by K-SVD[2].

[2] [Aharon et al. '06]

Global Convergence Guarantees for OCTOBOS

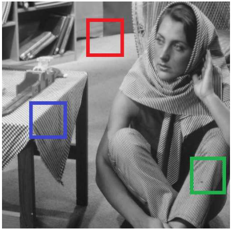
$$\text{(P1)} \quad \min_{\{W_k, X_i, C_k\}} \overbrace{\sum_{k=1}^{k} \sum_{i \in C_k} \|W_k Y_i - X_i\|_2^2}^{\text{Sparsification Error}} + \overbrace{\sum_{k=1}^{k} \lambda_k \left(\|W_k\|_F^2 - \log|\det W_k| \right)}^{\text{Regularizer}}$$

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

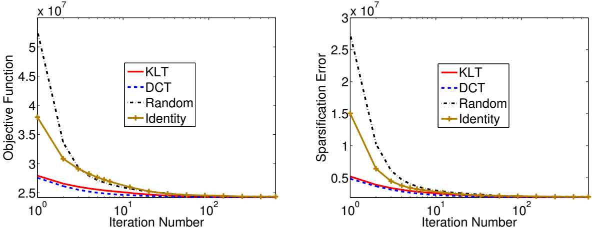
- **The alternating OCTOBOS learning algorithm is globally convergent to the set of partial minimizers of the objective in (P1).**
- These partial minimizers are global minimizers w.r.t. $\{W_k\}$ and $\{X_i, C_k\}$, respectively, and local minimizers w.r.t. $\{W_k, X_i\}$.
- **Under certain (mild) conditions, the algorithm converges to the set of stationary points of the equivalent objective $f(\mathbf{W})$.**

$$f(\mathbf{W}) \triangleq \sum_{i=1}^{N} \min_{k} \left\{ \|W_k Y_i - H_s(W_k Y_i)\|_2^2 + \lambda_0\, v(W_k)\, \|Y_i\|_2^2 \right\}$$

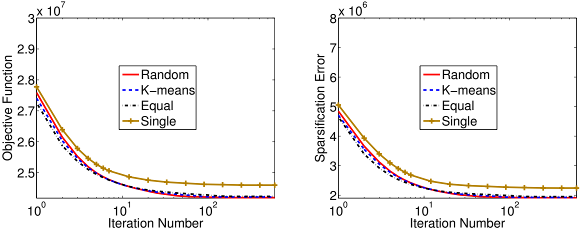
Algorithm Insensitive to Initializations



8×8 patches, $K = 2$



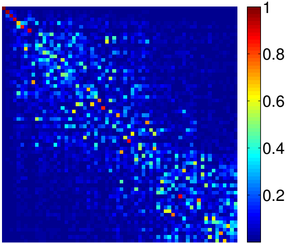
Various initializations for $\{W_k\}$



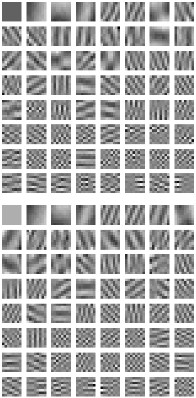
Various initializations for $\{C_k\}$ and $K = 1$ (single) case

Visualization of Learned OCTOBOS

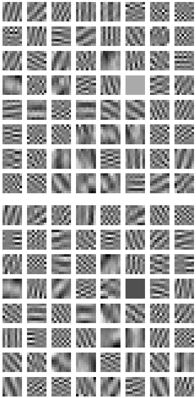
- The square blocks of a learnt OCTOBOS are **NOT** similar $\Rightarrow$ cluster-specific W_k.
- OCTOBOS **W** learned with different initializations appear different.
- **The W learned with different initializations sparsify equally well.**



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



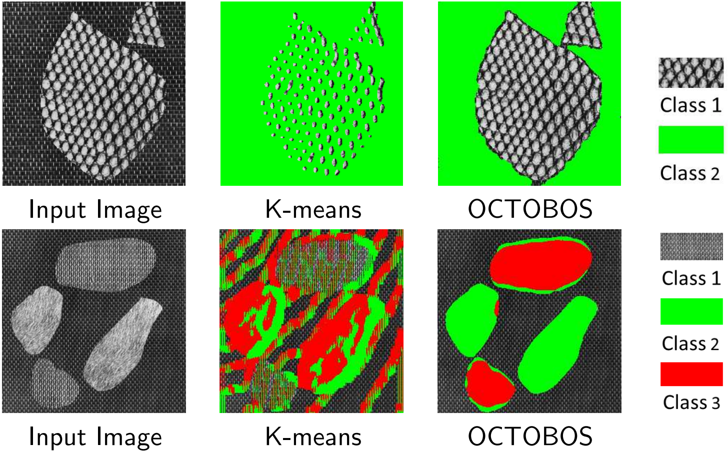
Random matrix Init.



KLT Init.

Application: Unsupervised Classification

- The overlapping image patches are first clustered by OCTOBOS learning.
- Each image pixel is then classified by a majority vote among the patches that cover that pixel.



Application: Compressed Sensing (CS)

- CS enables accurate recovery of images from far fewer measurements than the number of unknowns
  - Sparsity of image in transform domain or dictionary
  - Measurement procedure incoherent with transform
  - **Reconstruction non-linear**
- Reconstruction problem (**NP-hard**) -

$$\min_{x} \overbrace{\|Ax - y\|_2^2}^{\text{Data Fidelity}} + \lambda \overbrace{\|\Psi x\|_0}^{\text{Regularizer}} \tag{5}$$

- $x \in \mathbb{C}^P$: vectorized image, $y \in \mathbb{C}^m$: measurements $(m < P)$.
- A : **fat** sensing matrix, Ψ : transform. ℓ_0 "norm" counts non-zeros.
- CS with non-adaptive regularizer limited to low undersampling in MRI.

UNITE-BCS: Union of Transforms Blind CS

$$(\mathrm{P2}) \quad \min_{x,B,\{W_k,C_k\}} \nu \overbrace{\|Ax-y\|_2^2}^{\text{Data Fidelity}} + \overbrace{\sum_{k=1}^{K} \sum_{j \in C_k} \|W_k R_j x - b_j\|_2^2}^{\text{Sparsification Error}} + \eta^2 \overbrace{\sum_{k=1}^{K} \sum_{j \in C_k} \|b_j\|_0}^{\text{Sparsity Penalty}}$$

$$s.t. \;\; W_k^H W_k = I \;\forall\, k, \;\; \{C_k\} \in G, \;\; \|x\|_2 \leq C.$$

- $R_j \in \mathbb{R}^{n \times P}$ extracts patches. $W_k \in \mathbb{C}^{n \times n}$ is cluster-specific transform.
- $W_k R_j x \approx b_j, \forall j \in C_k, \forall k$ with $b_j \in \mathbb{C}^n$ sparse. $B \triangleq [b_1 \mid b_2 \mid \ldots \mid b_N]$.
- (P2) learns a union of unitary transforms, reconstructs x, and clusters the patches of x, using only the undersampled y.
  - $\Rightarrow$ **model adaptive to underlying image.**
- $\|x\|_2 \leq C$ is an energy or range constraint. $C > 0$.

Block Coordinate Descent (BCD) Algorithm for (P2)

- **Sparse Coding & Clustering:** Solves for only $\{C_k\}$ & B in (P2).

$$\min_{\{C_k\},\{b_j\}} \sum_{k=1}^{K} \sum_{j \in C_k} \left\{ \|W_k R_j x - b_j\|_2^2 + \eta^2 \|b_j\|_0 \right\} \qquad (6)$$
$$s.t. \;\; \{C_k\} \in G$$

- **Exact Clustering:** finds the global optimum $\{\hat{C}_k\}$ in (6) as

$$\left\{\hat{C}_k\right\} = \arg\min_{\{C_k\}} \sum_{k=1}^{K} \sum_{j \in C_k} \overbrace{\left\|W_k R_j x - H_\eta(W_k R_j x)\right\|_2^2 + \eta^2 \left\|H_\eta(W_k R_j x)\right\|_0}^{\text{Clustering Measure} \triangleq M_{k,j}} \qquad (7)$$

  - For patch $P_j x$, the optimal cluster index $\hat{k}_j = \arg\min_k M_{k,j}$.

- **Exact Sparse Coding by Hard-thresholding:** $\hat{b}_j = H_\eta(W_k R_j x) \;\; \forall j \in \hat{C}_k, \forall k$.

BCD Algorithm: Transform Update Step

- **Transform Update Step** solves (P2) for $\{W_k\}$. For each k, solve

$$\min_{W_k} \|W_k X_{C_k} - B_{C_k}\|_F^2 \quad s.t. \;\; W_k^H W_k = I. \tag{8}$$

  - X_{C_k} is matrix with columns $R_j x$ for $j \in C_k$. B_{C_k} is matrix of corresponding sparse codes.
  - **Closed-form solution:**

$$\hat{W}_k = VU^H \tag{9}$$

    - SVD: $X_{C_k} B_{C_k}^H = U \Sigma V^H$.

BCD Algorithm: Image Update Step

- **Image Update Step** solves (P2) for x with other variables fixed.

$$\min_{x} \; \nu \left\| Ax - y \right\|_2^2 + \sum_{k=1}^{K} \sum_{j \in C_k} \left\| W_k R_j x - b_j \right\|_2^2 \quad s.t. \quad \left\| x \right\|_2 \leq C. \quad (10)$$

  - Least squares problem with ℓ_2 norm constraint.
  - **Solve Least squares Lagrangian formulation with Normal Equation:**

$$\left(\sum_{j=1}^{N} R_j^T R_j \; + \; \nu \, A^H A + \hat{\mu} I \right) x = \sum_{k=1}^{K} \sum_{j \in C_k} R_j^T W_k^H b_j \; + \; \nu \, A^H y \quad (11)$$

    - The optimal multiplier $\hat{\mu} \in \mathbb{R}^+$ is the smallest real such that $\left\| \hat{x} \right\|_2 \leq C$. $\hat{\mu}$ and $\hat{x}$ can be found cheaply in applications such as MRI.

BCS Convergence Guarantees - Notations

- Define the barrier function $\varphi(W)$ as

$$\varphi(W) = \begin{cases} 0, & W^H W = I \\ +\infty, & \text{else} \end{cases}$$

- $\chi(x)$ is the barrier function corresponding to $\|x\|_2 \leq C$.
- (P2) can be written in unconstrained form:

$$h(W, B, \Gamma, x) = \nu \|Ax - y\|_2^2 + \sum_{k=1}^{K} \sum_{j \in C_k} \left\{ \|W_k R_j x - b_j\|_2^2 + \eta^2 \|b_j\|_0 \right\} + \sum_{k=1}^{K} \varphi(W_k) + \chi(x)$$

  - OCTOBOS W obtained by stacking the W_k's.
    Γ : row vector whose entries are the cluster indices of patches.

Union of Transforms BCS Convergence Guarantees

Theorem 1

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

- $\{h(W^t, B^t, \Gamma^t, x^t)\} \to h^* = h^*(W^0, B^0, \Gamma^0, x^0)$.
- $\{W^t, B^t, \Gamma^t, x^t\}$ *is bounded, and all its accumulation points are equivalent, i.e., they achieve the same value* h^* *of the objective.*
- $\left\|x^t - x^{t-1}\right\|_2 \to 0$ *as* $t \to \infty$.
- *Every accumulation point* (W, B, Γ, x) *satisfies the following partial global optimality conditions*

$$x \in \arg\min_{\tilde{x}} \; h(W, B, \Gamma, \tilde{x}) \tag{12}$$

$$W \in \arg\min_{\tilde{W}} \; h\left(\tilde{W}, B, \Gamma, x\right), \; (B, \Gamma) \in \arg\min_{\tilde{B}, \tilde{\Gamma}} \; h\left(W, \tilde{B}, \tilde{\Gamma}, x\right) \tag{13}$$

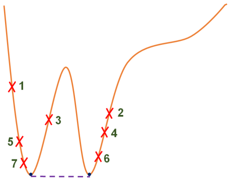
UNITE-BCS Convergence Guarantees

Theorem 2

Each accumulation point (W, B, Γ, x) *of* $\{W^t, B^t, \Gamma^t, x^t\}$ *also satisfies the following partial local optimality condition for all* $\Delta x \in \mathbb{C}^P$, *and all* $\Delta B \in \mathbb{C}^{n \times N}$ *satisfying* $\|\Delta B\|_\infty < \eta/2$.

$$h(W, B + \Delta B, \Gamma, x + \Delta x) \geq h(W, B, \Gamma, x) = h^* \tag{14}$$

UNITE-BCS Global Convergence Guarantees



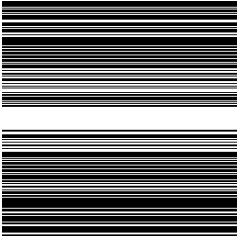
Corollary 1

For each initialization, the iterate sequence in the BCD algorithm converges to an equivalence class (same objective values) of accumulation points of the objective that are also partial global and partial local minimizers.

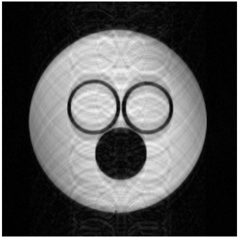
Corollary 2

The BCD algorithm is **globally convergent** *to (a subset of) the set of partial minimizers of the objective.*

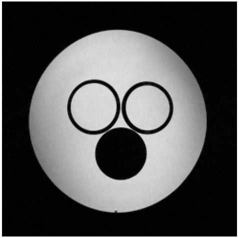
CS MRI Example - 2.5x Undersampling ($K = 3$)



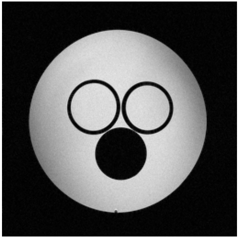
Sampling mask



Zero-filling (24.9 dB)

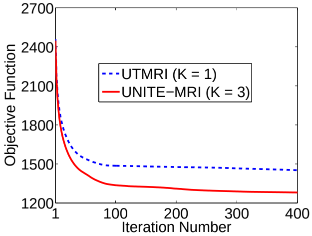


UNITE-MRI recon (37.3 dB)

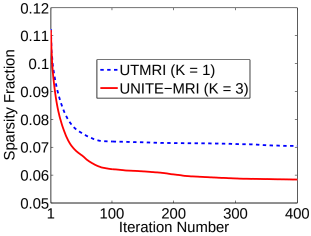


Reference

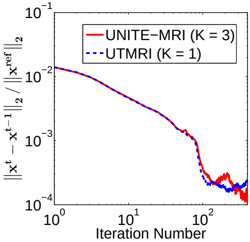
Convergence Behavior: UTMRI ($K = 1$) & UNITE-MRI



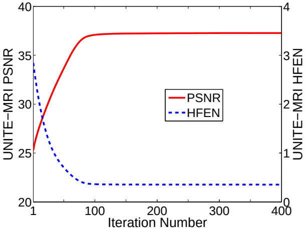
Objectives



Sparsity fractions (B)

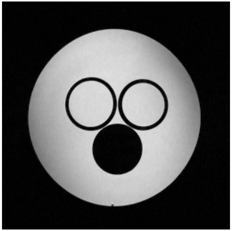


$\left\|x^t - x^{t-1}\right\|_2$



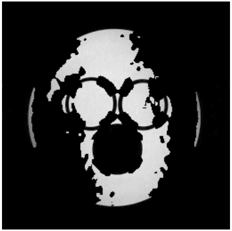
UNITE-MRI PSNR, HFEN

UNITE-MRI Clustering with $K = 3$



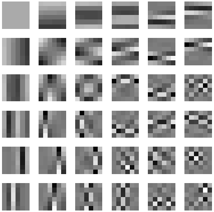
UNITE-MRI recon

Cluster 1

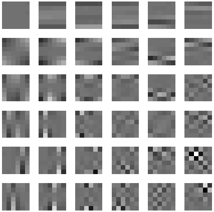


Cluster 2

Cluster 3

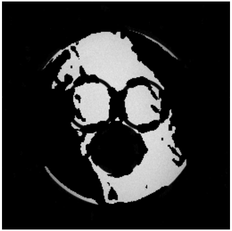


Real part of learnt W for cluster 2

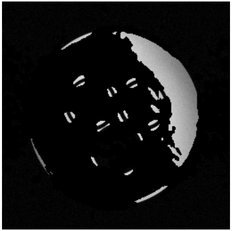


Imaginary part of learnt W for cluster 2

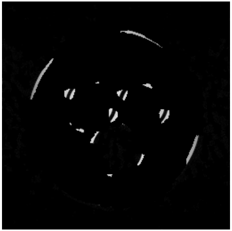
UNITE-MRI Clustering with $K = 4$



Cluster 1

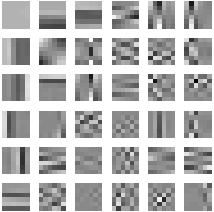


Cluster 2

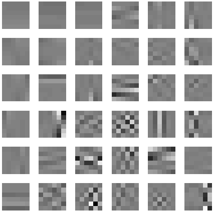


Cluster 3

Cluster 4

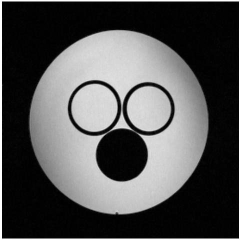


Real part of learnt W for cluster 4

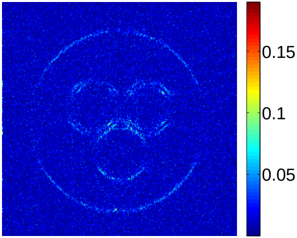


Imaginary part of learnt W for cluster 4

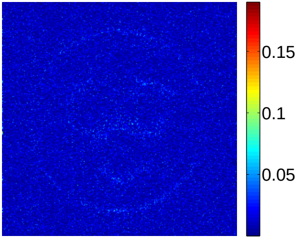
Reconstructions - Cartesian 2.5x Undersampling ($K = 16$)



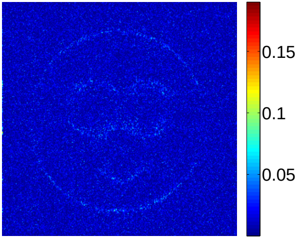
UNITE-MRI recon (37.4 dB, 631s)



DLMRI[3] error (36.6 dB, 1797s)



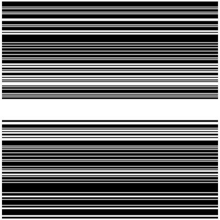
UNITE-MRI error



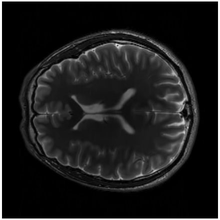
UTMRI error (37.2 dB, 125s)

[3] [Ravishankar & Bresler '11]

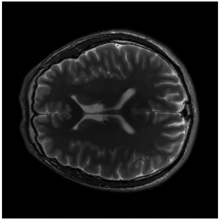
Example - Cartesian 2.5x Undersampling ($K = 16$)



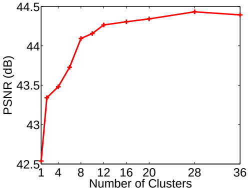
Sampling mask



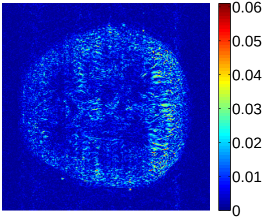
UTMRI (42.5 dB)



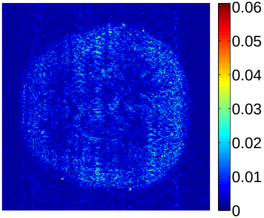
UNITE-MRI (44.3 dB)



PSNR vs. K



UTMRI Error



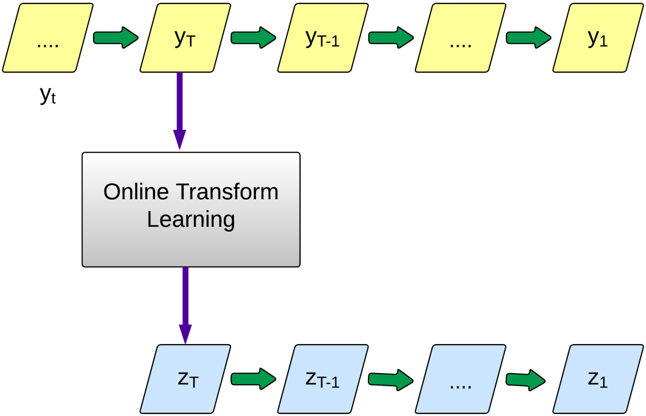
UNITE-MRI Error

Online Transform Learning

Why Online Transform Learning?

- **Batch learning:** learning using all the training data simultaneously.
- Big data $\Rightarrow$ large training sets $\Rightarrow$ batch learning computationally expensive in time and memory.
- Real-time or streaming data applications $\Rightarrow$ data arrives sequentially, and must be processed sequentially to limit latency.
- Online learning uses sequential model adaptation and signal reconstruction.
  - cheap computations and modest memory requirements.

Online Transform Learning



z_t : Learnt Transform/Sparse Codes/Signal Estimates

Online Transform Learning Formulation

- For $t = 1, 2, 3, ...,$ solve

$$\text{(P3)} \;\; \left\{\hat{W}_t, \hat{x}_t\right\} = \underset{W,\, x_t}{\arg\min}\; \frac{1}{t}\sum_{j=1}^{t}\left\{\|Wy_j - x_j\|_2^2 + \lambda_j v(W)\right\}$$

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

- $\lambda_j = \lambda_0 \|y_j\|_2^2$, with $\lambda_0 > 0$. $v(W) \triangleq \|W\|_F^2 - \log|\det W|$.
- λ_0 controls the condition number and scaling of learned $\hat{W}_t$.
- $\hat{W}_t^{-1}\hat{x}_t$ is an (e.g., denoised) estimate of y_t computed efficiently.
- For non-stationary data, use forgetting factor $\rho \in [0, 1]$, to diminish the influence of old data.

$$\frac{1}{t}\sum_{j=1}^{t}\rho^{t-j}\left\{\|Wy_j - x_j\|_2^2 + \lambda_j v(W)\right\} \tag{15}$$

Mini-Batch Transform Learning

- For $J = 1, 2, 3, ...,$ solve

$$\left\{\hat{W}_J, \hat{X}_J\right\} = \underset{W,\, X_J}{\arg\min} \frac{1}{JM} \sum_{j=1}^{J} \left\{ \| WY_j - X_j \|_F^2 + \Lambda_j v(W) \right\}$$

$$s.t. \;\; \|x_{JM-M+i}\|_0 \leq s, \; 1 \leq i \leq M. \;\; \mathrm{(P4)}$$

- $Y_J = [\, y_{JM-M+1} \,|\, y_{JM-M+2} \,|\, ... \,|\, y_{JM} \,]$, with M : mini-batch size.
- $X_J = [\, x_{JM-M+1} \,|\, x_{JM-M+2} \,|\, ... \,|\, x_{JM} \,]$. $\Lambda_j = \lambda_0 \, \| Y_j \|_F^2$.
- Mini-batch learning
  - **can provide reductions in operation count over online learning.**
  - **increased latency and memory requirements.**
- Alternative: Sparsity constraints can be replaced with ℓ_0 penalties.

Online Transform Learning Algorithm

- **Sparse Coding:** solve for x_t in (P3) with fixed $W = \hat{W}_{t-1}$.

$$\min_{x_t} \|Wy_t - x_t\|_2^2 \;\; s.t. \;\; \|x_t\|_0 \leq s \tag{16}$$

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

- **Transform Update:** solves for W in (P3) with $x_t = \hat{x}_t$.

$$\min_W \frac{1}{t}\sum_{j=1}^{t}\left\{\|Wy_j - x_j\|_2^2 + \lambda_j\left(\|W\|_F^2 - \log|\det W|\right)\right\} \tag{17}$$

$$\hat{W}_t = 0.5R_t\left(\Sigma_t + \left(\Sigma_t^2 + 2\beta_t I\right)^{\frac{1}{2}}\right)Q_t^T L_t^{-1} \tag{18}$$

  - $t^{-1}\sum_{j=1}^{t}\left(y_j y_j^T + \lambda_0 \|y_j\|_2^2 I\right) = L_t L_t^T$. **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 (P4) with fixed $W = \hat{W}_{J-1}$.

$$\min_{X_J} \|WY_J - X_J\|_F^2 \quad s.t. \quad \|x_{JM-M+i}\|_0 \leq s \;\forall\, i. \tag{19}$$

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

- **Transform Update:** solves for W in (P4) with fixed $\{X_j\}_{j=1}^{J}$.

$$\min_{W} \frac{1}{JM} \sum_{j=1}^{J} \left\{ \|WY_j - X_j\|_F^2 + \Lambda_j \left(\|W\|_F^2 - \log|\det W| \right) \right\} \tag{20}$$

  - Closed-form solution involving SVDs.
  - For $M \ll n$, use rank-M updates. For $M \geq O(n)$, direct SVDs.

Comparison of Computations, Memory, and Latency

Properties	Online	Mini-batch		Batch
		Small $M \ll n$	Large M	
Computations per sample	$O(n^2 \log^2 n)$	$O(n^2 \log^2 n)$	$O(n^2)$	$O(Pn^2)$
Memory	$O(n^2)$	$O(n^2)$	$O(nM)$	$O(nN)$
Latency	0	$M-1$	$M-1$	$N-1$

- Latency: max. time between arrival of a signal and generation of the output.
- P: # batch iterations, N: total samples, M: mini-batch size, n: signal size.
- $\log^2 n < P \Rightarrow$ online scheme is computationally cheaper than batch.
- For big data, online & mini-batch schemes have low memory & latency costs.
- **Online synthesis learning[4] has high computational cost per sample: $O(n^3)$.**

[4] [Mairal et al. '10]

Online Learning Convergence Analysis: Notations

- **The objective in the transform update step of (P3) is**

$$\hat{g}_t(W) = \frac{1}{t}\sum_{j=1}^{t}\left\{\|Wy_j - \hat{x}_j\|_2^2 + \lambda_0 \|y_j\|_2^2\, v(W)\right\} \qquad (21)$$

- **The empirical objective function is**

$$g_t(W) = \frac{1}{t}\sum_{j=1}^{t}\{\|Wy_j - H_s(Wy_j)\|_2^2 + \lambda_0 \|y_j\|_2^2\, v(W)\} \qquad (22)$$

  - 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.

Expected Transform Learning Cost

- **Assumption:** 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.

- We consider the minimization of the expected learning cost:

$$g(W) = \mathbb{E}_y \left[\|Wy - H_s(Wy)\|_2^2 + \lambda_0 \|y\|_2^2 \, v(W) \right] \tag{23}$$

- It follows from the Assumption that $\lim_{t\to\infty} g_t(W) = g(W)$ a.s.

- 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

Theorem 3

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

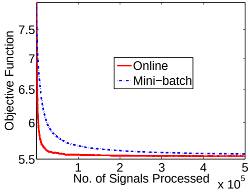
(i) *As $t \to \infty$, $\hat{g}_t(\hat{W}_t)$, $g_t(\hat{W}_t)$, and $g(\hat{W}_t)$ all converge a.s. to a common limit, say g^*.*

(ii) *The sequence $\{\hat{W}_t\}$ is bounded. Every accumulation point $\hat{W}_\infty$ of $\{\hat{W}_t\}$ satisfies $\nabla g(\hat{W}_\infty) = 0$ and $g(\hat{W}_\infty) = g^*$ with probability 1.*

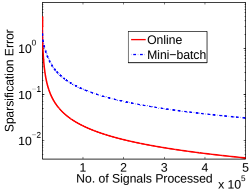
(iii) *The distance between $\hat{W}_t$ and the set of stationary points of $g(W)$ converges to 0 a.s.*

(iv) *$\hat{g}_{t+1}(\hat{W}_{t+1}) - \hat{g}_t(\hat{W}_t)$ and $\hat{W}_{t+1} - \hat{W}_t$ both decay as $O(1/t)$.*

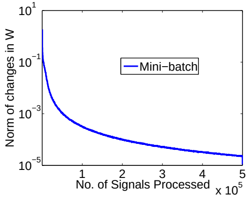
Empirical Convergence Behavior



Objective



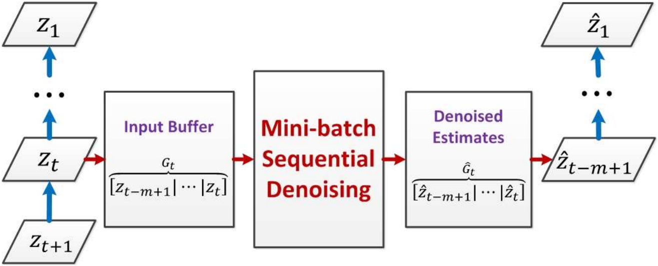
Sparsification error



$||\hat{W}_{t+1} - \hat{W}_t||_F$ $(M = 320)$

- $\{y_t\}$ generated as $\{W^{-1}x_t\}$ with random unitary 20×20 W, and random x_t with $\|x_t\|_0 = 3$.
- Objective converges quickly for both the online and mini-batch schemes.
- Sparsification error converges to zero, and $\kappa(W) \in [1.02, 1.04]$ for the schemes $\Rightarrow$ **learned a good model.**

Online Video Denoising by 3D Transform Learning



- z_t is a noisy video frame. $\hat{z}_t$ is its denoised version.
- G_t is a tensor with m frames formed using a sliding window scheme.
- Overlapping 3D patches in the G_t's are denoised sequentially using adaptive mini-batch denoising.
- Denoised patches averaged at 3D locations to yield frame estimates.

Video	σ	DCT	SK-SVD	VBM3D	VBM4D	**VIDOLSAT** ($n = 512$)	**VIDOLSAT** ($n = 768$)
Salesman	10	36.9	37.0	37.3	37.1	37.8	**38.0**
	20	33.1	33.2	34.1	33.3	34.0	**34.3**
	50	27.8	28.4	28.3	28.3	29.3	**29.7**
Miss America	10	39.5	39.7	39.6	39.9	40.3	**40.3**
	20	36.2	37.3	38.0	37.8	38.3	**38.4**
	50	30.6	33.4	34.6	34.3	35.2	**35.3**
Coastguard	10	34.6	34.8	34.8	35.4	**35.7**	35.7
	20	31.1	31.3	31.7	31.7	32.2	**32.3**
	50	26.6	27.1	26.9	27.1	28.0	**28.1**

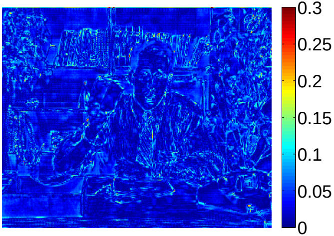
- Proposed VIDOLSAT is simulated at two patch sizes: $8 \times 8 \times 8$ ($n = 512$), and $8 \times 8 \times 12$ ($n = 768$).
- **VIDOLSAT provides 1.7 dB, 1.2 dB, 0.8 dB, and 0.8 dB better PSNRs than 3D DCT, sparse K-SVD[5], VBM3D[6], and VBM4D[7].**

[5] [Rubinstein et al. '10] [6] [Dabov et al. '07] [7] [Maggioni et al. '12]

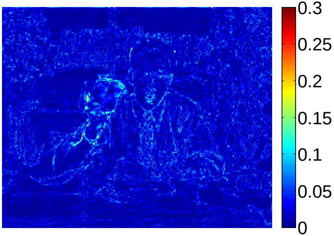
Video Denoising Example: Salesman

Noisy frame

VIDOLSAT (PSNR = 30.97 dB)



VBM4D Error (PSNR = 27.20 dB)



VIDOLSAT Error

Conclusions

- We introduced several data-driven sparse model adaptation techniques.
- Transform learning methods
  - are highly efficient and scalable
  - enjoy good theoretical and empirical convergence behavior
  - are highly effective in many applications
- Highly promising results were obtained using transform learning for denoising and compressed sensing.
- Future work: online blind compressed sensing.
- Acknowledgments: Yoram Bresler, Bihan Wen.
- **Transform learning webpage:** http://transformlearning.csl.illinois.edu

Thank you! Questions??