Lecture IX. Variations in Sparse Signal Reconstruction

Hojin Kim

Department of Radiation Oncology,
Yonsei Cancer Center, Severance Hospital,
Yonsei University College of Medicine

In Lecture VII

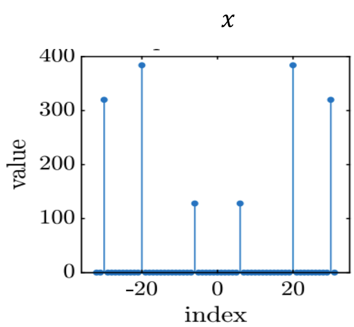
- Review of Iterative Methods
  - Features of under-determined problem
  - Sparse signal recovery with L_1-norm minimization
  - Basic algorithm for L_1-norm minimization with proximity operator

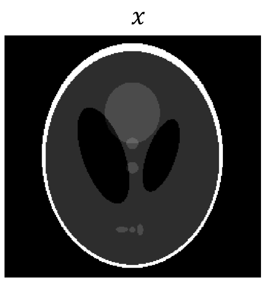
- Today, Let's take a look at further details about
  Variations in sparse signal recovery (a.k.a. Transform Sparsity)

Transform Sparsity

Real-World Application

- In reality: Image / Fluence-map has many non-zero elements





32412 non-zero / 65536 (49.5%)

$$\underset{x}{\text{minimize}} \quad f(x) = \| x \|_1 = \sum_{i=1}^{n} |x_i|$$

$$\text{subject to} \quad Ax = b$$

$$\Rightarrow \quad \underset{x}{\text{minimize}} \quad f(x) = \| Ax - b \|_2^2 + \lambda \| x \|_1$$

Real-World Application -- Transform Sparsity

- Transform sparsity: Making something sparse by transformation

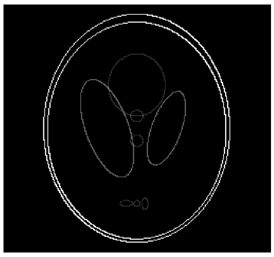
x

32412 non-zero / 65536 (49.5%)

Transform (D) →

Total-Variation (TV)

$$Dx = |x_{u-1,v} - x_{u,v}| + |x_{u,v-1} - x_{u,v}|$$



2184 non-zero / 65536 (3.3%)

- Minimizing transform sparsity (**Total-variation, Wavelet, Discrete cosine, etc.**)

$$\underset{x}{\text{minimize}} \;\; f(x) = \|Ax-b\|_2^2 + \lambda\|x\|_1 \quad \Rightarrow \quad \underset{x}{\text{minimize}} \;\; f(x) = \|Ax-b\|_2^2 + \lambda\|Dx\|_1$$

Limit of Basic L_1-minimization Algorithm

- Soft-thresholding operator

$$\underset{x}{\text{minimize}} \;\; f(x) = \|Ax-b\|_2^2 + \lambda\|x\|_1 = g(x) + h(x) \quad \Rightarrow \quad \begin{aligned} g(x) &= \|Ax-b\|_2^2 \\ h(x) &= \lambda\|x\|_1 \end{aligned}$$

* Proximity operator *h(x)* deals with very simple L_1-norm operator

$$prox_h(z,t) = \underset{x}{\text{argmin}}\left[\frac{1}{2t}\|x-z\|_2^2 + \|x\|_1\right] = SoftThreshold(z,t)$$

➔ Not sufficient for the real-world applications

Algorithms for Transform Sparsity

Alternating Direction Method of Multipliers (ADMM)

Chambolle-Pock Algorithm

YONSEI UNIVERSITY
YCC PHYSICS

Publications for ADMM (Split-Bregman Iterative Method)

SIAM J. IMAGING SCIENCES
Vol. 2, No. 2, pp. 323–343

The Split Bregman Method for L1-Regularized Problems*

Tom Goldstein† and Stanley Osher†

Abstract. The class of L1-regularized optimization problems has received much attention recently because of the introduction of "compressed sensing," which allows images and signals to be reconstructed from small amounts of data. Despite this recent attention, many L1-regularized problems still remain difficult to solve, or require techniques that are very problem-specific. In this paper, we show that Bregman iteration can be used to solve a wide variety of constrained optimization problems. Using this technique, we propose a "split Bregman" method, which can solve a very broad class of L1-regularized problems. We apply this technique to the Rudin–Osher–Fatemi functional for image denoising and to a compressed sensing problem that arises in magnetic resonance imaging.

Key words. constrained optimization, L1-regularization, compressed sensing, total variation denoising

AMS subject classification. 65K05

DOI. 10.1137/080725891

Foundations and Trends® in
Machine Learning
Vol. 3, No. 1 (2010) 1–122

DOI: 10.1561/2200000016

now
the essence of knowledge

Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers

Stephen Boyd[1], Neal Parikh[2], Eric Chu[3]
Borja Peleato[4] and Jonathan Eckstein[5]

[1] *Electrical Engineering Department, Stanford University, Stanford, CA 94305, USA, boyd@stanford.edu*
[2] *Computer Science Department, Stanford University, Stanford, CA 94305, USA, npparikh@cs.stanford.edu*
[3] *Electrical Engineering Department, Stanford University, Stanford, CA 94305, USA, echu508@stanford.edu*
[4] *Electrical Engineering Department, Stanford University, Stanford, CA 94305, USA, peleato@stanford.edu*
[5] *Management Science and Information Systems Department and RUTCOR, Rutgers University, Piscataway, NJ 08854, USA, jeckstei@rci.rutgers.edu*

Alternating Direction Method of Multiplier (ADMM) a.k.a Split-Bregman

- Concept of ADMM

$$\underset{x}{\text{minimize}} \;\; f(x) = g(x) + h(x) \;\; \Rightarrow$$

$g(x)$: Convex, Differentiable

$h(x) = \|Dx\|_1$: Convex, Non-differentiable

$$h(x) = \|(Dx)\|_1 \;\; \Rightarrow \;\; \boxed{d \triangleq (Dx)}$$

$$\Rightarrow \;\; \underset{x,d}{\text{minimize}} \;\; f(x,d) = g(x) + |d| + \|d - (Dx)\|_2^2$$

$$\Rightarrow \;\; \underset{x,d,q}{\text{minimize}} \;\; f(x,d,q) = g(x) + |d| + \frac{\lambda}{2}\|d - (Dx) - q\|_2^2$$

YONSEI UNIVERSITY YCC PHYSICS

Pseudo-Code of ADMM

$$\underset{x,d,q}{\text{minimize}} \;\; f(x,d,q) \;\; = g(x) + |d| + \frac{\lambda}{2}\|d - (Dx) - q\|_2^2$$

Step 1. Update x: $\underset{x}{\text{minimize}} \;\; g(x) + \frac{\lambda}{2}\|d - (Dx) - q\|_2^2$

Step 2. Update d: $\underset{d}{\text{minimize}} \;\; |d| + \frac{\lambda}{2}\|d - (Dx) - q\|_2^2 \;\; \Rightarrow \;\; d_{k+1} = SoftThreshold\left((Dx_{k+1}) + q_k, \frac{1}{\lambda}\right)$

Step 3. Update q: $\underset{b}{\text{minimize}} \;\; \frac{\lambda}{2}\|d - (Dx) - q\|_2^2 \;\; \Rightarrow \;\; q_{k+1} = q_k + \left((Dx_{k+1}) - d_{k+1}\right)$

Example of ADMM -- Image Denoising

- Denoising with TV min.

$$\underset{x}{\text{minimize}} \;\; f(x) = g(x) + h_h(x) + h_v(x) = \frac{\mu}{2}\|x-f\|_2^2 + \|D_h x\|_1 + \|D_v x\|_1$$

where $|Dx(i,j)| = |x(i+1,j) - x(i,j)| + |x(i,j+1) - x(i,j)| = |D_h x(i,j)| + |D_v x(i,j)|$

$$\Rightarrow \;\; \underset{x,d_h,d_v,b_h,b_v}{\text{minimize}} \;\; \frac{\mu}{2}\|x-f\|_2^2 + |d_h| + |d_v| + \frac{\lambda}{2}\|d_h - D_h x - q_h\|_2^2 + \frac{\lambda}{2}\|d_v - D_v x - q_v\|_2^2$$

Step 1. Update x: $\underset{x}{\text{minimize}} \;\; \frac{\mu}{2}\|x-f\|_2^2 + \frac{\lambda}{2}\|d_h - D_h x - q_h\|_2^2 + \frac{\lambda}{2}\|d_v - D_v x - q_v\|_2^2$

$$\Rightarrow \;\; x_{i,j}^{k+1} = \frac{\mu f}{\mu + 4\lambda} + \frac{\lambda}{\mu+4\lambda}[x_{i+1,j}^{k+1} + x_{i-1,j}^{k+1} + x_{i,j+1}^{k+1} + x_{i,j-1}^{k+1} + D_h^T(d_{h,(i,j)}^k - q_{h,(i,j)}^k) + D_v^T(d_{v,(i,j)}^k - q_{v,(i,j)}^k)]$$

$$\Rightarrow \;\; x_{i,j}^{k+1} = \frac{\mu f}{\mu + 4\lambda} + \frac{\lambda}{\mu+4\lambda}[x_{i+1,j}^{k+1} + x_{i-1,j}^{k+1} + x_{i,j+1}^{k+1} + x_{i,j-1}^{k+1}$$
$$+ d_{h,(i-1,j)}^k - d_{h,(i,j)}^k + d_{v,(i,j-1)}^k - d_{v,(i,j)}^k - q_{h,(i-1,j)}^k + q_{h,(i,j)}^k - q_{v,(i,j-1)}^k + q_{v,(i,j)}^k]$$

YONSEI UNIVERSITY
YCC PHYSICS

Example of ADMM -- Image Denoising

- Denoising with TV min.

$$\underset{x}{\text{minimize}} \;\; f(x) = g(x) + h_h(x) + h_v(x) = \frac{\mu}{2}\|x-f\|_2^2 + \|D_h x\|_1 + \|D_v x\|_1$$

$$\Rightarrow \;\; \underset{x,d_h,d_v,b_h,b_v}{\text{minimize}} \;\; \frac{\mu}{2}\|x-f\|_2^2 + |d_h| + |d_v| + \frac{\lambda}{2}\|d_h - D_h x - q_h\|_2^2 + \frac{\lambda}{2}\|d_v - D_v x - q_v\|_2^2$$

Step 2. Update d_u, d_v:

$$\underset{d_h}{\text{minimize}} \;\; |d_h| + \frac{\lambda}{2}\|d_h - D_h x - q_h\|_2^2 \;\; \Rightarrow \;\; d_h^{k+1} = SoftThreshold\left(D_h x^{k+1} + q_h^k, \frac{1}{\lambda}\right)$$

$$\underset{d_v}{\text{minimize}} \;\; |d_v| + \frac{\lambda}{2}\|d_v - D_v x - q_v\|_2^2 \;\; \Rightarrow \;\; d_v^{k+1} = SoftThreshold\left(D_v x^{k+1} + q_v^k, \frac{1}{\lambda}\right)$$

Step 3. Update q_h, q_v:

$$\underset{q_h}{\text{minimize}} \;\; \frac{\lambda}{2}\|d_h - D_h x - q_h\|_2^2 \;\; \Rightarrow \;\; q_h^{k+1} = q_h^k + \left(D_h x^{k+1} - d_h^{k+1}\right)$$

$$\underset{q_v}{\text{minimize}} \;\; \frac{\lambda}{2}\|d_v - D_v x - q_v\|_2^2 \;\; \Rightarrow \;\; q_v^{k+1} = q_v^k + \left(D_v x^{k+1} - d_v^{k+1}\right)$$

Example of ADMM -- Compressed Sensing

- Total-variation Minimization

$$\underset{x}{\text{minimize}}\ f(x) = g(x) + h_h(x) + h_v(x) = \frac{\mu}{2}\|Ax-b\|_2^2 + \|D_h x\|_1 + \|D_v x\|_1$$

where $|Dx(i,j)| = |x(i-1,j) - x(i,j)| + |x(i,j-1) - x(i,j)| = |D_h x(i,j)| + |D_v x(i,j)|$

$$\Rightarrow \underset{x,d_h,d_v,b_h,b_v}{\text{minimize}}\ \frac{\mu}{2}\|Ax-b\|_2^2 + |d_h| + |d_v| + \frac{\lambda}{2}\|d_h - D_h x - q_h\|_2^2 + \frac{\lambda}{2}\|d_v - D_v x - q_v\|_2^2$$

Step 1. Update x: $\underset{x}{\text{minimize}}\ \frac{\mu}{2}\|Ax-b\|_2^2 + \frac{\lambda}{2}\|d_h - D_h x - q_h\|_2^2 + \frac{\lambda}{2}\|d_v - D_v x - q_v\|_2^2$

???

$$x_{i,j}^{k+1} = \boxed{\frac{\mu A^T b}{\mu A^T A + 4\lambda} + \frac{\lambda}{\mu A^T A + 4\lambda}}[x_{i+1,j}^{k+1} + x_{i-1,j}^{k+1} + x_{i,j+1}^{k+1} + x_{i,j-1}^{k+1}$$
$$+ d_{h,(i-1,j)}^{k} - d_{h,(i,j)}^{k} + d_{v,(i,j-1)}^{k} - d_{v,(i,j)}^{k} - q_{h,(i-1,j)}^{k} + q_{h,(i,j)}^{k} - q_{v,(i,j-1)}^{k} + q_{v,(i,j)}^{k}]$$

Publication for Chambolle-Pock Algorithm

A first-order primal-dual algorithm for convex problems with applications to imaging

Antonin Chambolle*, and Thomas Pock†

June 9, 2010

Abstract

In this paper we study a first-order primal-dual algorithm for convex optimization problems with known saddle-point structure. We prove convergence to a saddle-point with rate $O(1/N)$ in finite dimensions, which is optimal for the complete class of non-smooth problems we are considering in this paper. We further show accelerations of the proposed algorithm to yield optimal rates on easier problems. In particular we show that we can achieve $O(1/N^2)$ convergence on problems, where the primal or the dual objective is uniformly convex, and we can show linear convergence, i.e. $O(1/e^N)$ on problems where both are uniformly convex. The wide applicability of the proposed algorithm is demonstrated on several imaging problems such as image denoising, image deconvolution, image inpainting, motion estimation and image segmentation.

Chambolle-Pock Algorithm

- Pseudo-code of Chambolle-Pock algorithm

$$\underset{x}{\text{minimize}} \quad F(Kx) + G(x) \quad \Rightarrow \quad \begin{aligned} y_l^{n+1} &= prox_{\sigma F^*}(y_l^n + \sigma \cdot K\tilde{x}^n) \\ x^{n+1} &= prox_{\tau G}(x^n - \tau \textstyle\sum_l K^T y_l^{n+1}) \\ \tilde{x}^{n+1} &= x^{n+1} + \theta(x^{n+1} - x^n) \end{aligned}$$

NOTE:

*Reminding 'Proximity Operator': $prox_h(z,t) = \underset{x}{\text{argmin}}\left[\frac{1}{2t}\|x-z\|_2^2 + h(x)\right]$

*Dual norm (denoted by *F**)

Assuming that ***F*** is ***L$_p$*-norm**, and **dual norm of *F* (*F**)** is ***L$_q$*-norm:** $\frac{1}{p} + \frac{1}{q} = 1$

*Hyper-parameters: $\sigma \cdot \tau \cdot L^2 \le 1$

Chambolle-Pock Algorithm

- Pseudo-code of Chambolle-Pock algorithm

$$\underset{x}{\text{minimize}} \quad F(Kx) + G(x) \quad \Rightarrow \quad \begin{aligned} y_l^{n+1} &= prox_{\sigma F^*}(y_l^n + \sigma \cdot K\tilde{x}^n) \\ x^{n+1} &= prox_{\tau G}(x^n - \tau \textstyle\sum_l K^T y_l^{n+1}) \\ \tilde{x}^{n+1} &= x^{n+1} + \theta(x^{n+1} - x^n) \end{aligned}$$

Acceleration

$$\begin{aligned} &\sigma_0 \cdot \tau_0 \cdot L^2 \le 1 \\ &\theta_n = \frac{1}{\sqrt{1+2\mu}}, \tau_{n+1} = \theta_n \tau_n, \sigma_{n+1} = \frac{\sigma_n}{\theta_n} \end{aligned} \quad \Rightarrow \quad \begin{aligned} y_l^{n+1} &= prox_{\sigma F^*}(y_l^n + \sigma_n \cdot K\tilde{x}^n) \\ x^{n+1} &= prox_{\tau G}(x^n - \tau_n \textstyle\sum_l K^T y_l^{n+1}) \\ \tilde{x}^{n+1} &= x^{n+1} + \theta_n(x^{n+1} - x^n) \end{aligned}$$

Example of Chambolle-Pock Algorithm

$$\underset{x}{\text{minimize}} \;\; F(Kx)+G(x) \quad \Rightarrow \quad \begin{aligned} y_l^{n+1} &= prox_{\sigma F^*}(y_l^{n}+\sigma \cdot K\tilde{x}^n) \\ x^{n+1} &= prox_{\tau G}(x^n-\tau \textstyle\sum_l K^T y_l^{n+1}) \\ \tilde{x}^{n+1} &= x^{n+1}+\theta(x^{n+1}-x^n) \end{aligned}$$

- Applying **Chambolle-Pock algorithm** to **Total-Variation Minimization**

$$\underset{x}{\text{minimize}} \;\; F(Kx)+G(x)=\frac{\mu}{2}\|Ax-b\|_2^2+\|D_h x\|_1+\|D_v x\|_1 \;(x\geq 0)$$

$$\Rightarrow \boxed{F(Kx)=F\left(\begin{bmatrix} A \\ D_h \\ D_v \end{bmatrix} x\right)} = f_1(Ax)+f_2(D_h x)+f_3(D_v x), \quad G(x)=[x]_+$$

$$\Rightarrow \; f_1(y_1)=\frac{\mu}{2}\|y_1-b\|_2^2 \;(y_1=Ax), \;\; f_2(y_2)=\|y_2\|_1\,(y_2=D_h x), \quad f_3(y_3)=\|y_3\|_1\,(y_3=D_v x)$$

Example of Chambolle-Pock Algorithm

$$\underset{x}{\text{minimize}} \;\; F(Kx)+G(x) \quad \Rightarrow \quad \begin{aligned} y_l^{n+1} &= prox_{\sigma F^*}(y_l^{n}+\sigma \cdot K\tilde{x}^n) \\ x^{n+1} &= prox_{\tau G}(x^n-\tau \textstyle\sum_l K^T y_l^{n+1}) \\ \tilde{x}^{n+1} &= x^{n+1}+\theta(x^{n+1}-x^n) \end{aligned}$$

- Applying **Chambolle-Pock algorithm** to **Total-Variation Minimization**

$$\boxed{\underset{x}{\text{minimize}} \;\; F(Kx)+G(x)=\frac{1}{2}\|Ax-b\|_2^2+\lambda\|D_h x\|_1+\lambda\|D_v x\|_1 \;(x\geq 0)}$$

$$\Rightarrow \boxed{F(Kx)=F\left(\begin{bmatrix} A \\ D_h \\ D_v \end{bmatrix} x\right)} = f_1(Ax)+f_2(D_h x)+f_3(D_v x), \quad G(x)=[x]_+$$

$$\Rightarrow \; f_1(y_1)=\frac{1}{2}\|y_1-b\|_2^2 \;(y_1=Ax), \;\; f_2(y_2)=\lambda\|y_2\|_1\,(y_2=D_h x), \; f_3(y_3)=\lambda\|y_3\|_1\,(y_3=D_v x)$$

Example of Chambolle-Pock Algorithm (Cont'd)

$$F(Kx) = f_1(Ax) + f_2(D_h x) + f_3(D_v x),\ \ G(x) = [x]_+$$
$$f_1(y_1) = \frac{\mu}{2}\|y_1 - b\|_2^2 \ \ (y_1 = Ax),$$
$$f_2(y_2) = \|y_2\|_1 \ (y_2 = D_h x),$$
$$f_3(y_3) = \|y_3\|_1 \ (y_3 = D_v x)$$

$$y_l^{n+1} = prox_{\sigma F^*}(y_l^n + \sigma \cdot K\tilde{x}^n)$$
$$x^{n+1} = prox_{\tau G}(x^n - \tau \sum_l K^T y_l^{n+1})$$
$$\tilde{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n)$$

Step 1. Update y's:

$$y_1^{n+1} = prox_{\sigma f_1^*}(y_1^n + \sigma \cdot A\tilde{x}^n) = \frac{(y_1^n + \sigma \cdot A\tilde{x}^n) + \sigma\mu b}{1 + \sigma\mu}$$

$$y_2^{n+1} = prox_{\sigma f_2^*}(y_2^n + \sigma \cdot D_h\tilde{x}^n) = Truncate(y_2^n + \sigma \cdot D_h\tilde{x}^n, \sigma)$$
$$= \begin{cases} y_2^n + \sigma \cdot D_h\tilde{x}^n & (|\, y_2^n + \sigma \cdot D_h\tilde{x}^n \,| < \sigma) \\ \sigma & (|\, y_2^n + \sigma \cdot D_h\tilde{x}^n \,| < \sigma) \end{cases}$$

Example of Chambolle-Pock Algorithm (Cont'd)

$$F(Kx) = f_1(Ax) + f_2(D_h x) + f_3(D_v x),\ \ G(x) = [x]_+$$
$$f_1(y_1) = \frac{\mu}{2}\|y_1 - b\|_2^2 \ \ (y_1 = Ax),$$
$$f_2(y_2) = \|y_2\|_1 \ (y_2 = D_h x),$$
$$f_3(y_3) = \|y_3\|_1 \ (y_3 = D_v x)$$

$$y_l^{n+1} = prox_{\sigma F^*}(y_l^n + \sigma \cdot K\tilde{x}^n)$$
$$x^{n+1} = prox_{\tau G}(x^n - \tau \sum_l K^T y_l^{n+1})$$
$$\tilde{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n)$$

Step 2. Update x:

$$x^{n+1} = prox_{\tau G}(x^n - \tau \sum_l K^T y_l^{n+1})$$
$$= \left[x^n - \tau(A^T y_1^{n+1} + D_h^T y_2^{n+1} + D_v^T y_3^{n+1}) \right]_+$$

Step 3. Update $\tilde{x}$:

(Acceleration)

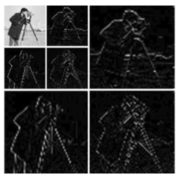
$$\tilde{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n)$$

Beyond Total-Variation (TV) Transform

Replacing TV-transform

Wavelet transform:

$$\underset{x}{\text{minimize}} \quad f(x) = \| Ax - p \|_2^2 + \sum_{l=1} \lambda_l \| W_l x \|_1$$



Non-local TV transform:

$$\underset{x}{\text{minimize}} \quad f(x) = \| Ax - p \|_2^2 + \lambda \| \nabla_{NLTV} x \|_k \quad (k \le 1)$$

$$\| \nabla_{NLTV} x \|_1 = \sum_i \sqrt{\sum_{j \in \Omega} (u_j - u_i)^2 w_{ij}}, \; w_{ij} = \exp\left(- \sum_{k=-a}^{a} G(k) \cdot \left| u_{j+k} - u_{i+k} \right|^2 / 2h_0^2 \right)$$

Block-Matching & 3D filtering:

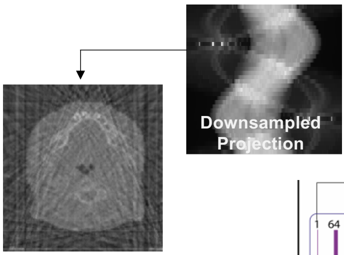
$$\underset{x}{\text{minimize}} \quad f(x) = \| Ax - p \|_2^2 + \lambda \| \Phi x \|_1$$

T1: 1D Hadamard
T2: 2D Wavelet

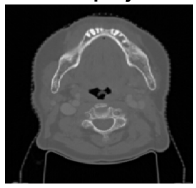
$$\omega_r = (T_1 \otimes T_2) \cdot (I_r \cdot x) \qquad \Phi = \begin{bmatrix} (T_1 \otimes T_2) \cdot I_{r=1} \\ (T_1 \otimes T_2) \cdot I_{r=2} \\ \vdots \\ (T_1 \otimes T_2) \cdot I_{r=R} \end{bmatrix}$$

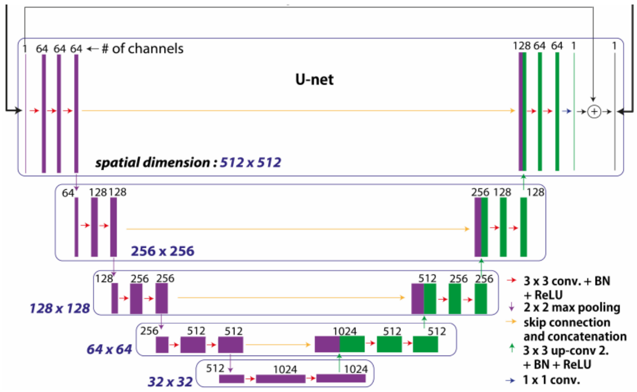
Replacing TV-transform (Cont'd)

Deep learning



CT image from
full-view projection





Notion of Group-Sparse Optimization

- Sparse Signal Optimization

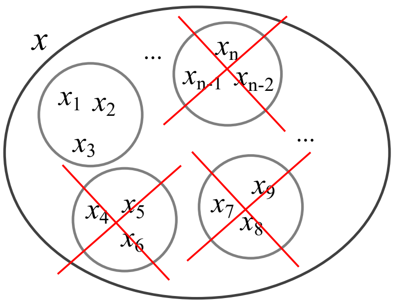
Lk-norm: $\|x\|_k = (|x_1|^k + |x_2|^k + |x_3|^k + \ldots + |x_n|^k)^{\frac{1}{k}}$

L2-norm: $\|x\|_2 = \sqrt{|x_1|^2 + |x_2|^2 + |x_3|^2 + \ldots + |x_n|^2}$

L1-norm: $\|x\|_1 = |x_1| + |x_2| + |x_3| + \ldots + |x_n|$

L0-norm: $\|x\|_0 = \sum_j 1\ \{x_j \neq 0\}$

- Group-Sparse Optimization



- 'Group is sparse': Entire x is divided into sub-groups

➔ **Very few groups** contribute to the optimization

$$\min_x .\ \|Ax - d\|_2^2 + \boxed{\lambda \|x\|_2^p}\ \ (p \leq 1)$$

$\mathbf{L_{2,p}}$**-norm**

- For treatment plans? ➔ **Beam Angle, Energy Layer!!!**

Summary

- Understanding **Necessity of Trasform Sparsity** in real-world applications
  ➔ Transforming 'Dense signal' into 'Sparse signal', so that L1-min. is applicable

- Important algorithms to solve transform sparsity: ADMM, C.P.
  ➔ Both have potential for the various applications

- Besides TV transform, the other options would be available

YONSEI UNIVERSITY
COLLEGE OF MEDICINE