Assignment 3: CS 754

Krushnakant Bhattad	Devansh Jain
190100036	190100044

March 23, 2021

Contents

Question 1

(a) Restricted Eigenvalues Condition

We say that a matrix $\boldsymbol{X}$ satisfies the restricted eigenvalues condition with respect to a constraint set $\mathcal{C}$ if there exists a constant $\gamma > 0$ such that the following holds for all non-zero $\nu \in \mathcal{C}$:

$$\frac{\frac{1}{N}\nu^T \boldsymbol{X}^T \boldsymbol{X}\nu}{\|\nu\|_{\ell_2}^2} \geq \gamma$$

Equivalently:

$$\frac{\frac{1}{N}\|\boldsymbol{X}\nu\|_{\ell_2}^2}{\|\nu\|_{\ell_2}^2} \geq \gamma$$

If we have the above condition, that the quantity on the LHS is bounded away from zero, then we know we have enough curvature so that a small loss difference implies a small error.

(b) Starting from equation 11.20 ...

We know that $\hat{\beta}$ is the minimizer of

$$K(\beta) = (1/2N)\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_{\ell_2}^2 + \lambda_N \|\boldsymbol{\beta}\|_{\ell_1},$$

that is we know that:

$$(1/2N)\left\|\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right\|_{\ell_2}^2 + \lambda_N \left\|\hat{\boldsymbol{\beta}}\right\|_{\ell_1} \leq (1/2N)\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta^*}\|_{\ell_2}^2 + \lambda_N \|\boldsymbol{\beta^*}\|_{\ell_1}$$

Due to this it follows from definition of G that $G(\hat{\nu}) \leq G(0)$

Page No. :
Date : / /

Q1 (c)

We observe that :

$$\|y - X\hat{\beta}\|_2^2 - \|y - X\beta^*\|_2^2 = \|y - X(\hat{v} + \beta^*)\|_2^2 - \|y - X\beta^*\|_2^2$$

$$= \|w - X\hat{v}\|_2^2 - \|w\|_2^2$$

$$= (w - X\hat{v})^T(w - X\hat{v}) - w^Tw$$

$$= -(X\hat{v})^Tw - w^TX\hat{v} + (X\hat{v})^T(X\hat{v})$$

$$= \|X\hat{v}\|^2 - 2w^TX\hat{v} \quad (\because \underbrace{w^TX\hat{v} = (X\hat{v})^Tw}_{\text{Scalars}})$$

Next, see that :

$$G(\hat{v}) \leq G(0)$$

$$\therefore \frac{1}{2N}\|y - X\hat{\beta}\|_2^2 + \lambda_N\|\hat{\beta}\| \leq \frac{1}{2N}\|y - X\beta^*\|_2^2 + \lambda\|\beta^*\|_1$$

i.e.

$$\lambda_N(\|\beta^*\|_1 - \|\hat{\beta}\|_1) \geq \frac{1}{2N}(\|y - X\hat{\beta}\|_2^2 - \|y - X\beta^*\|_2^2)$$

$$= \frac{1}{2N}(\|X\hat{v}\|_2^2 - 2w^TX\hat{v})$$

i.e.

$$\frac{\|X\hat{v}\|_2^2}{2N} \leq \frac{w^TX\hat{v}}{2N} + \lambda_N(\|\beta^*\|_1 - \|\beta^* + \hat{v}\|_1)$$

(d) Holder's inequality states that:

Let $p, q \in [1, \infty)$ with $\frac{1}{p} + \frac{1}{q} = 1$

Then, $\|fg\|_1 \leq \|f\|_p \|g\|_q$

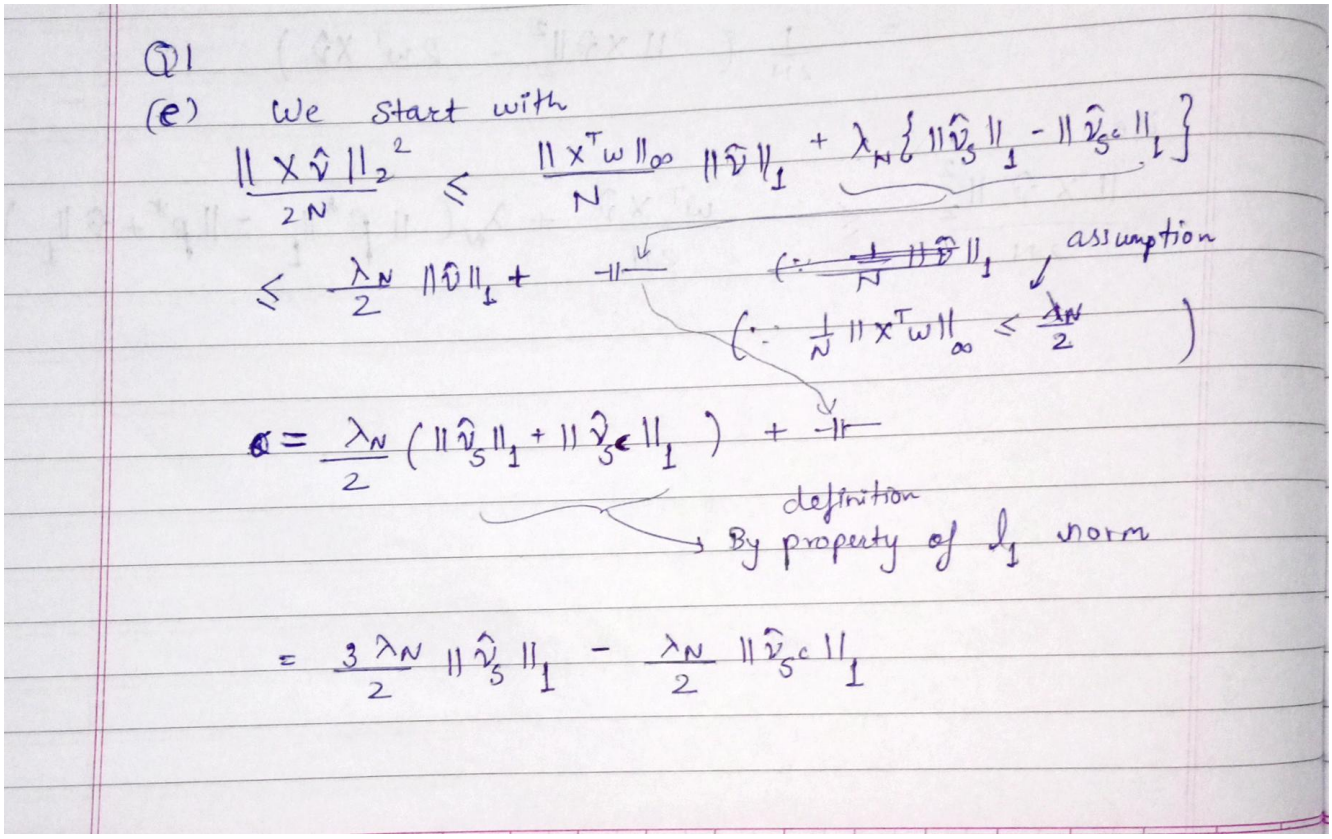
$\because |w^T X \hat{v}|$ is a scalar,

$$w^T X \hat{v} \leq |w^T X \hat{v}| = \|w^T X \hat{v}\|_{l_1}$$
$$= \|(X^T w)^T \hat{v}\|_{l_1}$$
$$\leq \|(X^T w)^T\|_\infty \|\hat{v}\|_{l_1}$$
$$= \|X^T w\|_\infty \|\hat{v}\|_1$$

Here, we use, $f = (X^T w)^T$, $g = \hat{v}$

($\because \frac{1}{\infty} + \frac{1}{1} = 1$) $p = \infty$ & $q = 1$

In this way we apply holder's inequality to obtain equation 11.22


Q1
(e) We start with
assumption
definition
By property of l_1 norm

Q1 (e) contd.

By Cauchy Schwartz inequality we have,

$$\sum_i u_i v_i \leq \sqrt{\sum_i u_i^2} \sqrt{\sum_i v_i^2}$$

which,
with $u_i = |\hat{v}_i|$ & $v_i = 1$

for $i \in S$, the support of the ~~k-s.~~ $\hat{v}_S$ with $|S| = k$

we get,

$$\|\hat{v}_S\|_1 \leq \sqrt{k}\, \|\hat{v}_S\|_1 \leq \sqrt{k}\, \|\hat{v}\|_1$$

Thus, $\frac{3\lambda_N}{2} \|\hat{v}_S\|_1 - \frac{\lambda_N}{2} \|v_{S^c}\|_1$

$$\leq \frac{3\lambda_N}{2} \|\hat{v}_S\|_1$$

$$\leq \frac{3\lambda_N}{2} \sqrt{k}\, \|\hat{v}\|_1$$

So finally, $\frac{\|X\hat{v}\|_2^2}{2N} \leq \frac{3}{2}\lambda_N \sqrt{k}\, \|\hat{v}\|_1$

Q1 (f)

$\because \hat{v}$ belongs to the cone set $\mathbf{C}(S;3)$, (by lemma 11.1); it allows us to apply the restricted eigenvalue condition with corresponding γ to get:

$$\frac{1}{N}\|X\hat{v}\|_2^2 \geq \gamma\|v\|_2^2$$

In (e) we proved:

$$\frac{1}{2N}\|X\hat{v}\|_2^2 \leq \frac{3}{2}\sqrt{k}\,\lambda_N\|\hat{v}\|_2$$

Which finally combined gives

$$\frac{\gamma}{2}\|\hat{v}\|_2^2 \leq \frac{3}{2}\sqrt{k}\,\lambda_N\|\hat{v}\|_2$$

~~Substituting~~ $\Rightarrow \frac{1}{2}\|v\|_2 \leq \frac{3}{2}\sqrt{k}\,\frac{1}{\gamma}\lambda_N$ ✝

Finally we get,

$$\|\hat{\beta}-\beta^*\|_2 \leq \frac{3}{\gamma}\sqrt{\frac{k}{N}}\sqrt{N}\,\lambda_N$$

(g)

The bound $\lambda_N \geq 2 \frac{\|X^T w\|_\infty}{N}$

is used in two places:

(1) To derive the first part of inequality (11.23) from the inequality (11.22)

(2) Assuming inequality (11.22), this bound is used to derive another inequality

$$0 \leq \frac{\lambda_w}{2} \|\hat{\nu}\|_1 + \lambda_N \{ \|\hat{\nu}_S\|_1 - \|\hat{\nu}_{S^c}\|_1 \}$$

in proof of Lemma 11.1, which converts inequality 11.23 to the bound 11.146.

(h) Why is the cone constraint required?

It is clear that the following least-squares objective function is always convex.

$$f_N(\beta) = \frac{1}{2N} \|\boldsymbol{y} - \boldsymbol{X\beta}\|_2^2$$

What we desire is that whenever the difference in function values $\Delta f_N = |f_N(\beta_1) - f_N(\beta_2)|$ converges to zero, the ℓ_2-norm of the parameter vector difference $\Delta\beta = \|\beta_1 - \beta_2\|_2$ also converges to zero.

This notion is a strengthening of ordinary convexity, hence called strong convexity.

$f_N(\beta)$ is also strongly convex whenever the eigenvalues of the $p \times p$ positive semi-definite matrix $\boldsymbol{X^T X}$ are uniformly bounded away from zero.

Simple linear algebra shows that if $N < p$, the matrix $\boldsymbol{X^T X}$ (which always has rank at most $\min\{N, p\}$) is always rank-deficient and thus not strongly convex.

For this reason, we need to relax our notion of strong convexity. Well we're anyway concerned about about a particular set that only contains some specific sparse vectors; hence we will see the condition constrained to that set only.

In the current condition, we suppose that the parameter vector β^* is sparse - say supported on some subset $S = S(\beta^*)$.

The restricted eigenvalues condition considers the corresponding specific constraint set $\mathcal{C}$, in which we can achieve restricted strong convexity. These notions are related because for this objective function, the hessian matrix $\nabla^2 f(\beta)$ is proportional to $\boldsymbol{X^T X}$

The LASSO error is defined as $\hat{\nu} = \hat{\beta} - \beta^*$.

Let $\hat{\nu}_S$ denote the subvector indexed by elements of S, and let $\hat{\nu}_{S^c}$ defined in an analogous manner.

As we proved in the lemma 11.1, it turns out (in those particular conditions, where it is called regularized form) that the lasso error satisfies a cone constraint of the form $\|\hat{\nu}_{S^c}\|_1 \leq 3\|\hat{\nu}_S\|_1$

For the constrained lasso with ball radius $R = \|\beta^*\|_1$, it can be proved that $\|\hat{\nu}_{S^c}\|_1 \leq \|\hat{\nu}_S\|_1$

A general cone constraint is $\|\hat{\nu}_{S^c}\|_1 \leq \alpha\|\hat{\nu}_S\|_1$ for some constant $\alpha \geq 1$.

Thus, in either its constrained or regularized form, the lasso error is restricted to a set of the form

$$\mathcal{C}(S;\alpha) := \{\nu \in \mathbb{R}^p \text{ such that } \|\hat{\nu}_{S^c}\|_1 \leq \alpha\|\hat{\nu}_S\|_1\} \text{ for some parameter } \alpha \geq 1.$$

Thus by the cone constraint we achieve strong convexity in the region of the lasso error, which thus helps us achieve our goal, that is, whenever the difference in function values $\Delta f_N = |f_N(\beta_1) - f_N(\beta_2)|$ converges to zero, the ℓ_2-norm of the parameter vector difference $\Delta\beta = \|\beta_1 - \beta_2\|_2$ also converges to zero.

(i) This theorem Vs Theorem 3 that we did in class.

1. There was a restriction on the value of δ_{2S} ($< \sqrt{2} - 1$) in Theorem 3, here, there is no restriction on γ (except > 0) in this Theorem.
2. As described on Pg. 296, "the rate (11.15)—including the logarithmic factor—is known to be minimax optimal, meaning that it cannot be substantially improved upon by any estimator.".
3. On Pg. 302, Theorem 11.3 states the uniqueness of the optimal solution $\hat{\beta}$, no false inclusions and exclusions of indices; and hence is variable selection consistent.
4. On applying Theorem 3 with error modelled as zero-mean Gaussian with standard deviation σ, we choose ϵ as $9m\sigma^2$ (whp), which gives the error bound proportional to σ^2.
   In this example, we showed that taking λ_N proportional to σ is also a valid choice, with high probability; which provides a tighter bound than Theorem 3.
5. In this theorem, we have restricted $\beta*$ to be a k-sparse vector, i.e. sparse in basis $\mathbf{X}$.
   Theorem 3 has the advantage that it gives error bounds for compressible signals as well.

(j) What is the common thread between the bounds on the 'Dantzig selector' and the LASSO?

$\sigma_k(x)$ is defined on Pg. 12 as the error incurred by approximating a signal x by some $\hat{x} \in \Sigma_k$:

$$\sigma_k(x)_p = \min_{\hat{x} \in \Sigma_k} ||x - \hat{x}||_p.$$

If $x \in \Sigma_k$, then clearly $\sigma_k(x)_p = 0$ for any p.

1. The theorem on LASSO error bound (Theorem 11.1) considering the signal $\beta*$ to be k-sparse.
   The error bound obtained is proportional to $\sqrt{k}$.
   If we apply the same condition on the signal x while using 'Dantzig selector' and using bounds mentioned (Theorem 1.10), the first term of the error bound would reduce to zero, thus making the error bound proportional to $\sqrt{k}$, same as above.
2. The bounds in LASSO estimation is proportional to the regularization parameter $\lambda_N \geq 2||\mathbf{X}^T\mathbf{w}||_\infty/N$.
   Upon applying similar condition, the bounds of 'Dantzig selector' are proportional to $\lambda \geq ||A^T e||_\infty$.

(k) What is the advantage of the square-root LASSO over the LASSO?

The LASSO:

The LASSO solves the following, where $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{X} \in \mathbb{R}^{n \times p}$:

$$\hat{\beta} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_{\ell_2}^2}{n} + \frac{\lambda}{n} \|\beta\|_{\ell_1} \qquad \dots (1)$$

with a suitable penalty level: $\lambda = c \cdot (2\sigma\sqrt{n}\Phi^{-1/2}(1 - \alpha/2p))$ where: $c > 1$ is some constant, σ is the standard deviation of noise, n is the number of observation samples, and Φ is the Gaussian probability distribution function.

The square root LASSO:

The square root LASSO solves the following:

$$\hat{\beta} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \sqrt{\frac{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_{\ell_2}^2}{n}} + \frac{\lambda}{n} \|\beta\|_{\ell_1} \qquad \dots (1)$$

where the suitable penalty level is: $\lambda = c \cdot (\sqrt{n}\Phi^{-1/2}(1 - \alpha/2p))$
All the terms have same meanings as above.

Advantages of the square root LASSO:

We observe that, in (2), the penalty level is independent of σ, the noise variance, which is contrasting with what we observed in (1). This observation is used in point 3 below.

Here are some advantages of the square root LASSO:

1. Despite taking the square-root of the least squares criterion function, the square root LASSO retains global convexity, making the estimator as computationally attractive as LASSO
2. The square-root LASSO can be formulated as a solution to a conic programming problem.
3. The original LASSO construction relies on knowing the standard deviation σ of the noise. Estimation of σ is nontrivial when p is large, particularly when $p \gg n$, and remains an outstanding practical and theoretical problem. The square root LASSO eliminates the need to know or to pre-estimate σ.
4. By using moderate deviation theory, we can do away with the assumption that noise follows the normal distribution under certain conditions.

Thus while keeping many of the advantages of LASSO intact, the square root LASSO adds more features to it.

Question 2

Instructions for running the code:

- After extracting submitted file, look for a directory named `q2`, and `cd` (change directory) to it.
- Run the files `q2a.m`, `q2b.m`, `q2c.m` and `q2d.m`. The results can be found in `./results/`.

Package l1_ls_matlab

Passing `A` and `At` as objects

```
Usage: [x,status]=l1_ls(A,At,m,n,y,lambda,rel_tol);
```

Common Steps

1. Reading the slice(s)
2. Padding the slice(s) to form square slice(s)
3. Angles of Projection[1]
4. Creating Tomographic Projections
5. Creating objects of Forward matrix (omitted in (a))
6. Reconstruction using
   (a) Filtered Back-projection with Ram-Lak filter
   (b) Compressed Sensing
   (c) 2 slice Coupled Compressed Sensing
   (d) 3 slice Coupled Compressed Sensing
7. Result

(b) Compressed Sensing

Minimization function

$$\arg\min_{\beta} \|\mathbf{y} - \mathbf{RU}\beta\|^2 + \lambda\|\beta\|_1$$

Class RU

Class `RU` present under folder `@RU`.
`@RU/RU.m` contains class constructor.
`@RU/ctranspose.m` contains overloaded transpose operator (').
`@RU/mtimes.m` contains overloaded product operator (*).

[1] Even though the question mentions to randomly generate projection angles, best results are found when the projection angles are uniformly spaced. Confirmed with Professor.

(c) 2 slice Coupled Compressed Sensing

Minimization function

$$\begin{aligned}
&\arg\min_{\beta_1,\beta_2} \|y_1 - R_1U\beta_1\|^2 + \|y_2 - R_2U\beta_2\|^2 + \lambda\|\beta_1\|_1 + \lambda\|\beta_1 - \beta_2\|_1 \\
&= \arg\min_{\beta_1,\Delta\beta} \|y_1 - R_1U\beta_1\|^2 + \|y_2 - R_2U(\beta_1 + \Delta\beta)\|^2 + \lambda\|\beta_1\|_1 + \lambda\|\Delta\beta\|_1 \\
&= \arg\min_{\beta_1,\Delta\beta} \left\| \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} R_1U & 0 \\ R_2U & R_2U \end{pmatrix} \begin{pmatrix} \beta_1 \\ \Delta\beta \end{pmatrix} \right\|^2 + \lambda \left\| \begin{pmatrix} \beta_1 \\ \Delta\beta \end{pmatrix} \right\|_1
\end{aligned}$$

where, $\beta_2 = \beta_1 + \Delta\beta$

Class CCS2

Class `CCS2` present under folder `@CCS2`.
`@CCS2/CCS2.m` contains class constructor.
`@CCS2/ctranspose.m` contains overloaded transpose operator (').
`@CCS2/mtimes.m` contains overloaded product operator (*).

(d) 3 slice Coupled Compressed Sensing

Minimization function

$$\begin{aligned}
&\arg\min_{\beta_1,\beta_2,\beta 3} \|y_1 - R_1U\beta_1\|^2 + \|y_2 - R_2U\beta_2\|^2 + \|y_3 - R_3U\beta_3\|^2 + \lambda\|\beta_2\|_1 + \lambda\|\beta_2 - \beta_1\|_1 + \lambda\|\beta_2 - \beta_3\|_1 \\
&= \arg\min_{\beta_2,\Delta\beta_1,\Delta\beta_2} \|y_1 - R_1U(\beta_2 + \Delta\beta_1)\|^2 + \|y_2 - R_2U\beta_2\|^2 + \|y_3 - R_3U(\beta_2 + \Delta\beta_2)\|^2 + \lambda\|\beta_2\|_1 + \lambda\|\Delta\beta_1\|_1 + \lambda\|\Delta\beta_2\|_1 \\
&= \arg\min_{\beta_2,\Delta\beta_1,\Delta\beta_2} \left\| \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} - \begin{pmatrix} R_1U & R_1U & 0 \\ R_2U & 0 & 0 \\ R_3U & 0 & R_3U \end{pmatrix} \begin{pmatrix} \beta_2 \\ \Delta\beta_1 \\ \Delta\beta_2 \end{pmatrix} \right\|^2 + \lambda \left\| \begin{pmatrix} \beta_2 \\ \Delta\beta_1 \\ \Delta\beta_2 \end{pmatrix} \right\|_1
\end{aligned}$$

Class CCS3

Class `CCS3` present under folder `@CCS3`.
`@CCS3/CCS3.m` contains class constructor.
`@CCS3/ctranspose.m` contains overloaded transpose operator (').
`@CCS3/mtimes.m` contains overloaded product operator (*).

Results

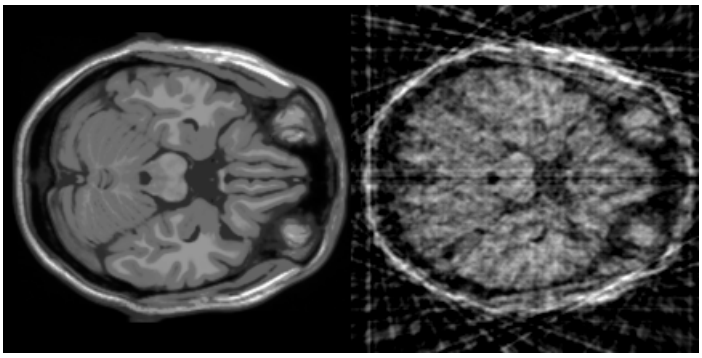


Figure 1: Original (`slice50.png`) Vs Reconstruction from Tomographic Projections (18 angles) using Filtered Back-projection

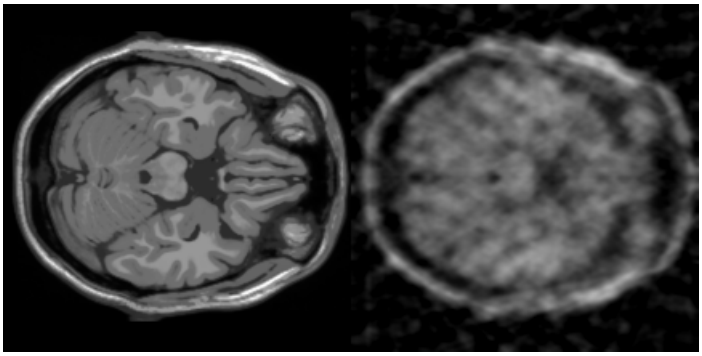


Figure 2: Original (`slice50.png`) Vs Reconstruction from Tomographic Projections (18 angles) using Compressed Sensing with $\lambda = 100$

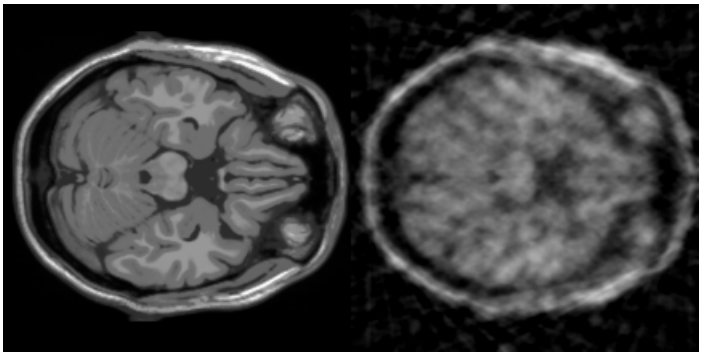


Figure 3: Original (`slice50.png`) Vs Reconstruction from Tomographic Projections (18 angles) using Compressed Sensing with $\lambda = 10$

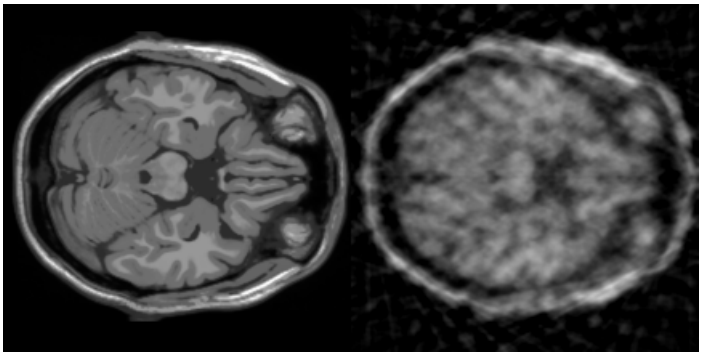


Figure 4: Original (`slice50.png`) Vs Reconstruction from Tomographic Projections (18 angles) using Compressed Sensing with $\lambda = 1$

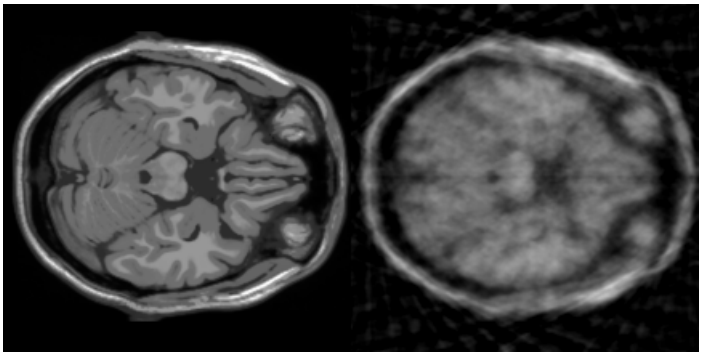


Figure 5: Original (`slice50.png`) Vs Reconstruction from Tomographic Projections (18 angles) using Compressed Sensing with $\lambda = 0.1$

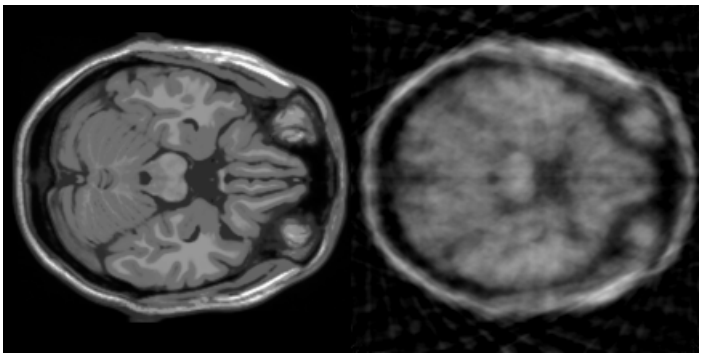


Figure 6: Original (`slice50.png`) Vs Reconstruction from Tomographic Projections (18 angles) using Compressed Sensing with $\lambda = 0.01$

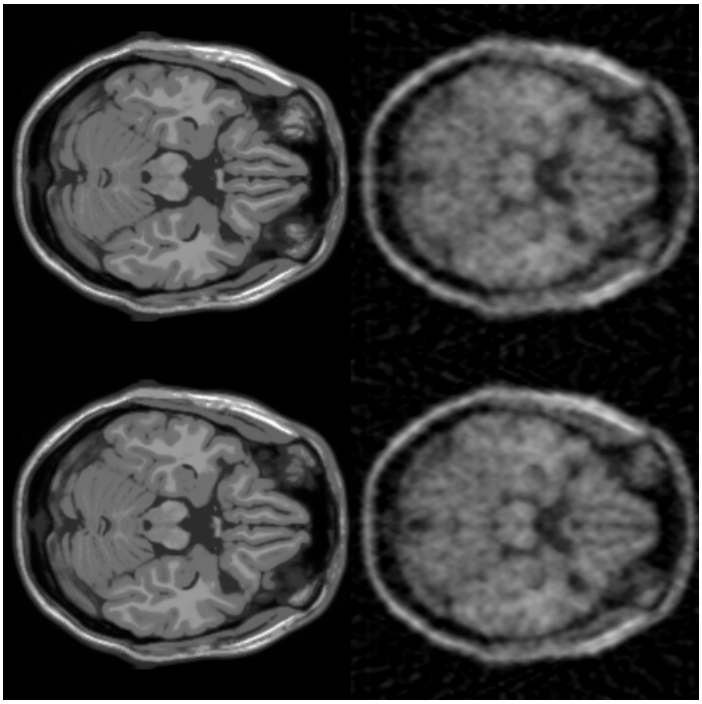


Figure 7: Original (`slice51.png`, `slice52.png`) Vs Reconstruction from Tomographic Projections (18 angles) using 2 slice Coupled Compressed Sensing with $\lambda = 100$

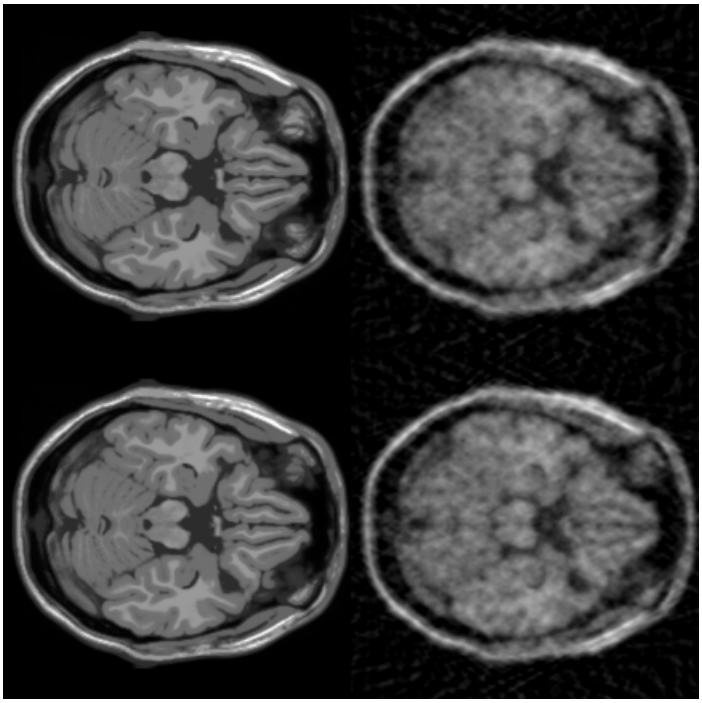


Figure 8: Original (`slice51.png`, `slice52.png`) Vs Reconstruction from Tomographic Projections (18 angles) using 2 slice Coupled Compressed Sensing with $\lambda = 10$

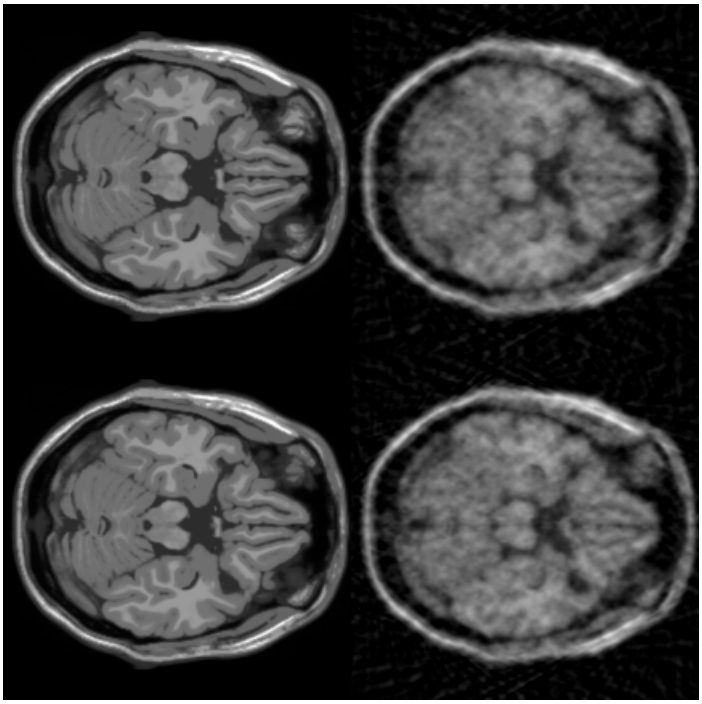


Figure 9: Original (`slice51.png`, `slice52.png`) Vs Reconstruction from Tomographic Projections (18 angles) using 2 slice Coupled Compressed Sensing with $\lambda = 1$

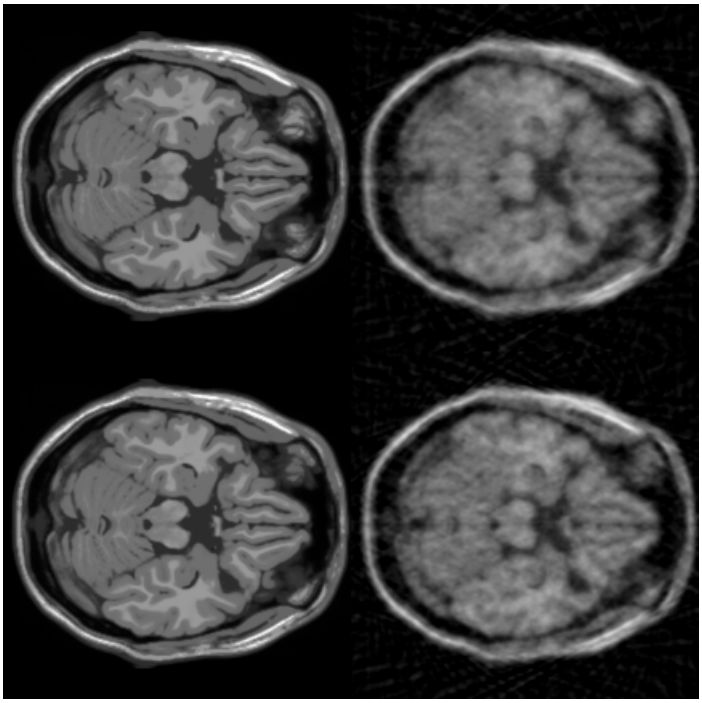


Figure 10: Original (`slice51.png`, `slice52.png`) Vs Reconstruction from Tomographic Projections (18 angles) using 2 slice Coupled Compressed Sensing with $\lambda = 0.1$

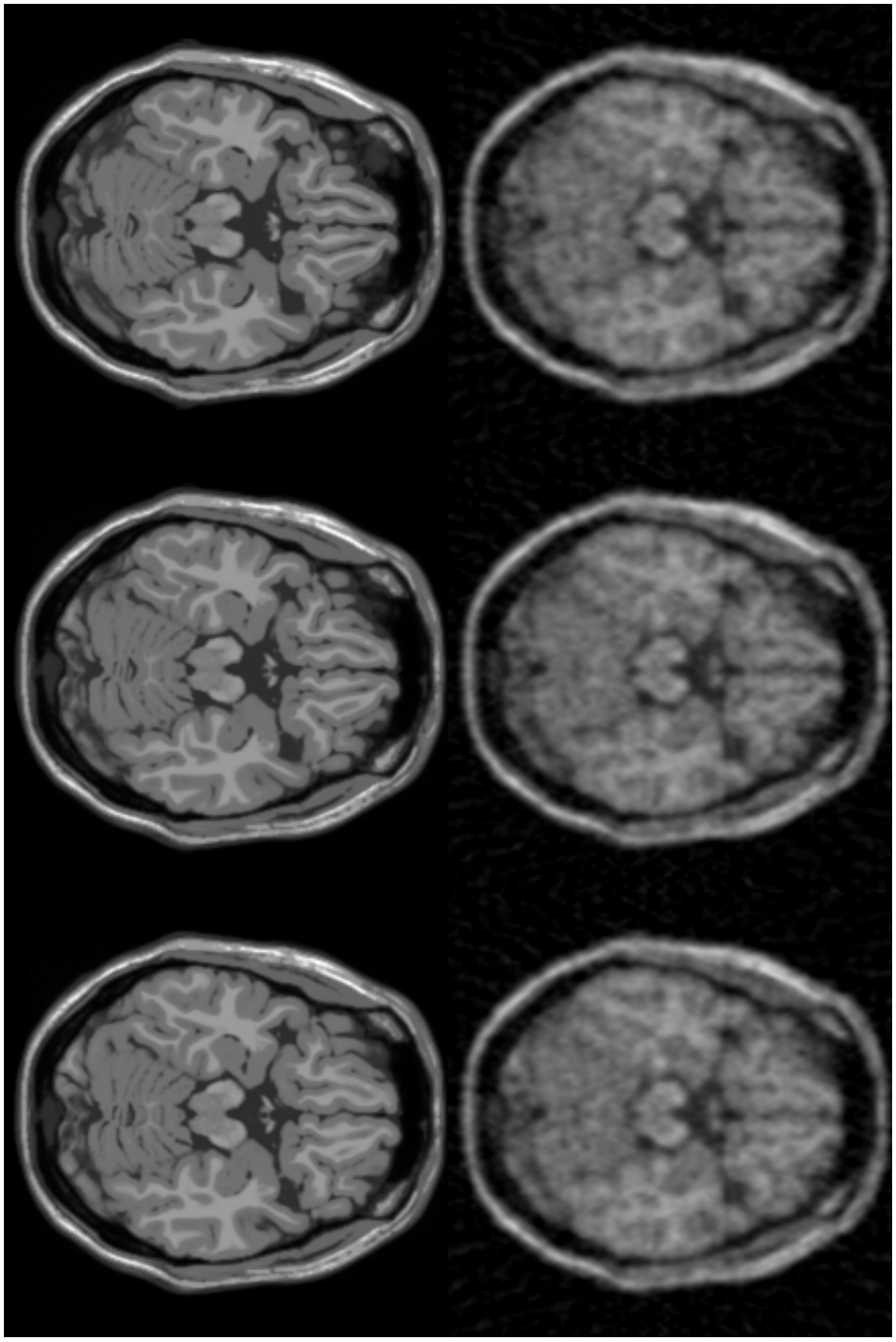


Figure 11: Original (`slice53.png`, `slice54.png`, `slice55.png`) Vs Reconstruction from Tomographic Projections (18 angles) using 3 slice Coupled Compressed Sensing with $\lambda = 100$

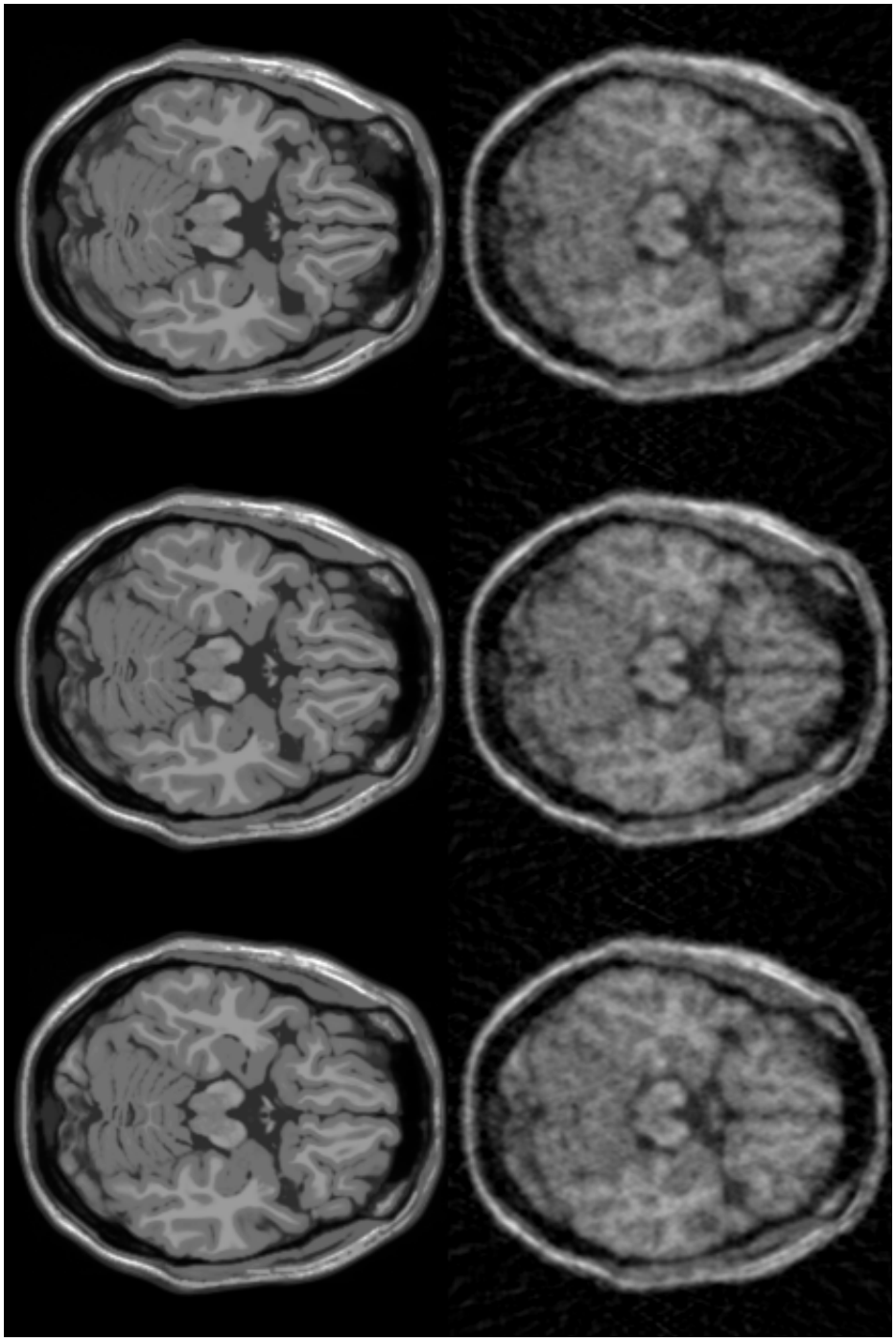


Figure 12: Original (`slice53.png`, `slice54.png`, `slice55.png`) Vs Reconstruction from Tomographic Projections (18 angles) using 3 slice Coupled Compressed Sensing with $\lambda = 10$

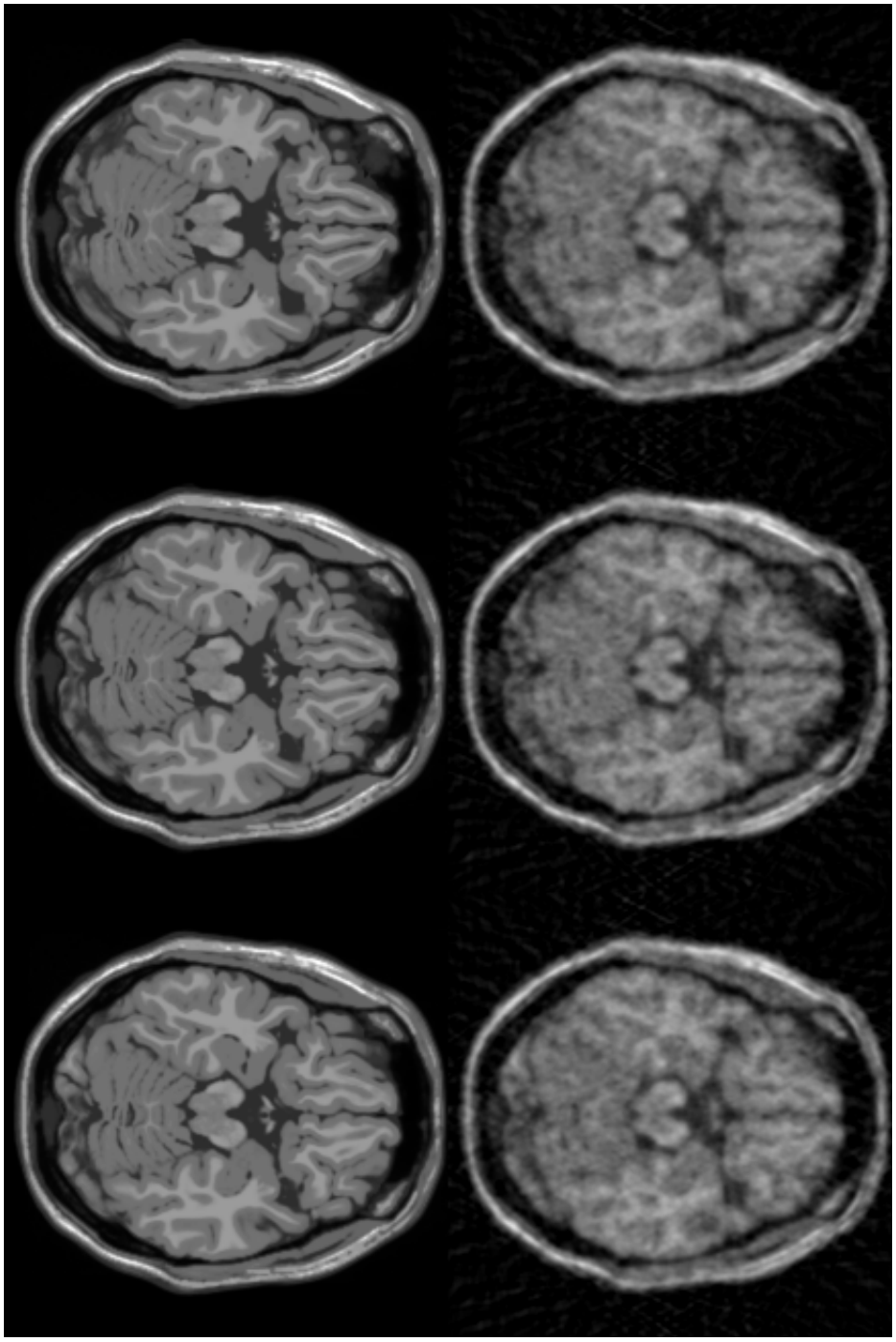


Figure 13: Original (`slice53.png`, `slice54.png`, `slice55.png`) Vs Reconstruction from Tomographic Projections (18 angles) using 3 slice Coupled Compressed Sensing with $\lambda = 1$

Question 3

(a) Shifting:

$LHS = R(g(x - x_0, y - y_0))(\rho, \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x - x_0, y - y_0)\delta(\rho - x\cos\theta - y\sin\theta)dxdy$

Now we'll do a change of variables. Let $x' = x - x_0$ and $y' = y - y_0$.

Thus we get,

$LHS = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x', y')\delta(\rho - x_0\cos\theta - y_0\sin\theta - x'\cos\theta - y'\sin\theta)dxdy$

$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x', y')\delta((\rho - x_0\cos\theta - y_0\sin\theta) - x'\cos\theta - y'\sin\theta)dx'dy'$

If we observe carefully, this is precisely the expression for the radon transform of $g(x, y)$ at an angle θ, with offset $\rho' = \rho - x_0\cos\theta - y_0\sin\theta$

Thus,

$LHS = R(g(x - x_0, y - y_0))(\rho, \theta) = R(g(x, y))(\rho - x_0\cos\theta - y_0\sin\theta, \theta) = RHS$

and we're done.

(b) Rotation:

Using change of variables to polar co-ordinates, the 2D radon transform can be written as:

$R(f(r, \psi))(\rho, \theta) = \int_{-\pi}^{\pi}\int_{0}^{\infty} f(r, \psi)\delta(\rho - r\cos\psi\cos\theta - r\sin\psi\sin\theta)rdrd\psi$

That is, $R(f(r, \psi))(\rho, \theta) = \int_{-\pi}^{\pi}\int_{0}^{\infty} f(r, \psi)\delta(\rho - r\cos(\psi - \theta))rdrd\psi$

Now let's get to proving.

$R(g(r, \psi - \psi_0))(\rho, \theta) = \int_{-\pi}^{\pi}\int_{0}^{\infty} g(r, \psi - \psi_0)\delta(\rho - r\cos(\psi - \theta))rdrd\psi$

Substitute $\psi' = \psi - \psi_0$.

$R(g(r, \psi - \psi_0))(\rho, \theta) = \int_{-\pi-\psi_0}^{\pi-\psi_0}\int_{0}^{\infty} g(r, \psi')\delta(\rho - r\cos(\psi' + \psi_0 - \theta))rdrd\psi'$

Noting that for functions with period 2π, as is the case here, $\int_{-\pi-\psi_0}^{\pi-\psi_0} f(\theta)d\theta = \int_{-\pi}^{\pi} f(\theta)d\theta$

$R(g(r, \psi - \psi_0))(\rho, \theta) = \int_{-\pi}^{\pi}\int_{0}^{\infty} g(r, \psi')\delta(\rho - r\cos(\psi' - (\theta - \psi_0))rdrd\psi'$

which is precisely the required radon transform:

$R(g(r, \psi - \psi_0))(\rho, \theta) = R(g(r, \psi))(\rho, \theta - \psi_0)$

(c) Convolution:

$$(f * k)(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1, y_1)k(x - x_1, y - y_1)dx_1dy_1$$

The radon transform of this is:

$$R((f * k)(x, y))(\rho, \theta)$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1, y_1)k(x - x_1, y - y_1)\delta(\rho - x\cos\theta - y\sin\theta)dx_1dy_1dxdy$$

Assuming the functions are "nice", and using properties of multiple integrals we can change the order of integration as follows:

$$R((f * k)(x, y))(\rho, \theta)$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1, y_1)\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} k(x - x_1, y - y_1)\delta(\rho - x\cos\theta - y\sin\theta)dxdy\right]dx_1dy_1$$

By the shifting property proved in (a), we can write the integral in the square brackets as:

$$R((f * k)(x, y))(\rho, \theta)$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1, y_1)\left[R(k(x, y))(\rho - x_1\cos\theta - y_1\sin\theta, \theta)\right]dx_1dy_1$$

Since $\int_{-\infty}^{\infty} f(x)\delta(x - x')dx = f(x')$,
we have
$$R(k(x, y))(\rho - x_1\cos\theta - y_1\sin\theta, \theta) = \int_{-\infty}^{\infty} R(k(x, y))(\rho - \rho_1, \theta)\delta(\rho_1 - x_1\cos\theta - y_1\sin\theta)d\rho_1$$
Thus we can write,

$$R((f * k)(x, y))(\rho, \theta)$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1, y_1)R(k(x, y))(\rho - \rho_1, \theta)\delta(\rho_1 - x_1\cos\theta - y_1\sin\theta)dx_1dy_1d\rho_1$$

Regrouping the terms, we have:

$$R((f * k)(x, y))(\rho, \theta)$$
$$= \int_{-\infty}^{\infty} R(k(x, y))(\rho - \rho_1, \theta)\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1, y_1)\delta(\rho_1 - x_1\cos\theta - y_1\sin\theta)dx_1dy_1\right]d\rho_1$$

Again, the terms in the square bracket allow us to write:

$$R((f * k)(x, y))(\rho, \theta)$$
$$= \int_{-\infty}^{\infty} R(k(x, y))(\rho - \rho_1, \theta)R(f(x, y))(\rho_1, \theta)d\rho_1$$

which is in fact the desired convolution, allowing us to write,

$$R((f * k)(x, y))(\rho, \theta)$$
$$= R(f(x, y))(\rho, \theta) * R(k(x, y))(\rho, \theta)$$

Question 4

Claim. Let $\boldsymbol{A}$ be the sensing matrix, such that every column of it is unit normalised.
If $\boldsymbol{A}$ has s-order restricted isometry constant δ_s and mutual coherence μ, $\delta_s \leq (s-1)\mu$

Proof:

Let S be a subset of columns of $\boldsymbol{A}$ with $|S| = s$.

Then, $\boldsymbol{A_S^T A_S}$ is a symmetric positive semi definite matrix.

Let λ_{max} and λ_{min} be the largest and smallest eigenvalues of $\boldsymbol{A_S^T A_S}$.
We have, $\lambda_{max} \geq \lambda_{min} \geq 0$
Suppose $\boldsymbol{x}$ be a $s-$sparse vector with the non-zero values in subset corresponding to S.
Then,

$$\begin{aligned}
\|\boldsymbol{Ax}\|^2 &= \|\boldsymbol{A_S x}_S\|^2 \\
&= (\boldsymbol{A_S x}_S)^T \boldsymbol{A_S x}_S \\
&= \boldsymbol{x}_S^T \boldsymbol{A}_S^T \boldsymbol{A_S x}_S \\
&= (\boldsymbol{A_S^T A_S x_S})^T \boldsymbol{x_S} \\
&= (\lambda \boldsymbol{x_S})^T \boldsymbol{x_S} \qquad \ldots \ \lambda \text{ is some eigenvalue of } \boldsymbol{A_S^T A_S} \\
&= \lambda(\boldsymbol{x_S}^T \boldsymbol{x_S}) \\
&\leq \lambda_{max} \|\boldsymbol{x_S}\|^2 \\
&= \lambda_{max} \|\boldsymbol{x}\|^2
\end{aligned}$$

Similarly, $\|\boldsymbol{Ax}\|^2 = \|\boldsymbol{A_S x}_S\|^2 \geq \lambda_{min} \|\boldsymbol{x}\|^2$

Thus, $\lambda_{min} \|\boldsymbol{x}\|^2 \leq \|\boldsymbol{Ax}\|^2 \leq \lambda_{max} \|\boldsymbol{x}\|^2$

Thus we get $\delta_s = \max\{1 - \lambda_{min}, \lambda_{max} - 1\}$.

Suppose λ be some eigenvalue of $\boldsymbol{A_S^T A_S}$.

Gershgorin's Disk Theorem says that:
For an eigenvalue λ of a square matrix $\boldsymbol{B}$ of $n \times n$, there is an index $j \in \{1, 2, \cdots, n\}$
such that $|\lambda - B_{jj}| \leq \sum_{i=0, i \neq j}^{i=n} |B_{ji}|$

In our case (where we consider $B = \boldsymbol{A_S^T A_S}$); due to unit normalisation, we have $(\boldsymbol{A_S^T A_S})_{jj} = 1$ and by definition of mutual coherence, we can also write the following:

$$\sum_{i=0, i \neq j}^{i=s} |(\boldsymbol{A_S^T A_S})_{ji}| = \sum_{i=0, i \neq j}^{i=s} |((\boldsymbol{A_S})_j)^T (\boldsymbol{A_S})_i| \leq \sum_{i=0, i \neq j}^{i=s} \mu(\boldsymbol{A}) = (s-1)\mu(\boldsymbol{A})$$

Therefore we have
$|\lambda - 1| \leq (s-1)\mu(\boldsymbol{A})$

Which implies $\lambda_{max} - 1 \leq (s-1)\mu(\boldsymbol{A})$ and $1 - \lambda_{min} \leq (s-1)\mu(\boldsymbol{A})$

Thus finally, $\delta_s \leq (s-1)\mu$

Question 5

Title : X-ray computed tomography for quality inspection of agricultural products: A review
Venue : Food Science & Nutrition - Wiley Online Library
First published : 23rd August 2019
Link : `https://onlinelibrary.wiley.com/doi/full/10.1002/fsn3.1179`

Mathematical Problem

X-ray CT technology is used to obtain two-dimensional slice images and three-dimensional tomographic images of samples. The review provides an overview of the working principle of X-ray CT technology, image processing, and analysis.

It models the tomographic projections based on the attenuation intensity as indicated by the **Beer–Lambert law** (Curry, Dowdey, & Murry, 1990; Wu, Yang, & Zhou, 2008; Ying & Han, 2005).

$$I = I_0 e^{-\mu_n l}$$

where I is the intensity of X-ray exiting through a sample in eV,
I_0 is the intensity of X-ray in eV,
μ_n is the linear attenuation coefficient of a sample on the wavelength in eV/mm, and
l is the path length through a sample in mm.

The calculation formula of CT number is as follows:

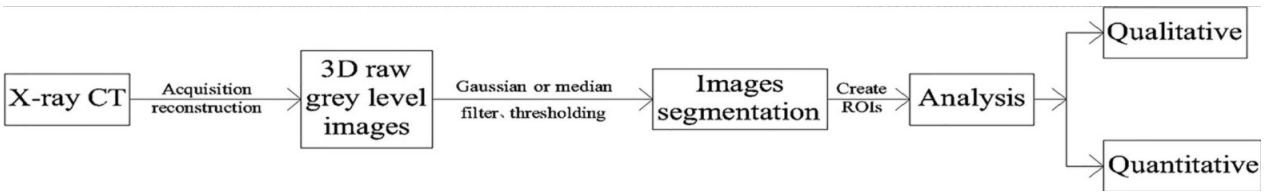
$$\text{CT Number} = \frac{\mu - \mu_w}{\mu} \times 1000$$

where μ is the linear attenuation coefficient of a sample, and
μ_w is the linear attenuation coefficient of water (approximately 0.195).
The measurement range of CT number is from –1,000 Hu to + 4,000 Hu. Besides, CT number for air is –1,000 Hu and the number for water is 0 Hu (Ogawa, 1998).

Image Processing and Analysis

Procedure:

1. 2D slice images are generated to reconstruct the 3D model
2. Gaussian or median filter is used to reduce noise with 3D raw gray-level images
3. The **threshold segmentation** is used to segment the image based on the gray value histogram of different regions. More on this below.
4. The final step is the qualitative and quantitative analysis on CT data of region of interests (ROIs).



Optimization Constructs

In the process, we used optimization in two different steps:

3D - reconstruction:

In the image processing part, we reconstruct the 3D model using various 2D slice images. This can be performed via various methods like using SURF(Speeded-Up Robust Features), or, say SIFT (Scale-Invariant Feature Transform) for feature detection steps: detection, description and matching, which can then be used with methods like SSD (Sum of Squared Differences) matching algorithm using image segmentation with aim to obtain accurate 3D models.
These methods involve optimization of parameters like matching cost($D(p)$):

$$D(p) = \exp(-SSD(p))$$

where $SSD(p)$ is the SSD score in a square neighborhood searching window.
We are omitting exact details of this optimization construct; as this is just a tool for accomplishing one particular step; however we mentioned this as the question asked for methods of optimization.

Threshold segmentation

In digital image processing, thresholding is a method of segmenting images. From a grayscale image, thresholding is used to create binary images.

Automatic thresholding is a great way to extract useful information encoded into pixels.

The **optimization** done for this purpose is to minimize background noise.

This is accomplished by utilizing a feedback loop to optimize the threshold value before converting the original grayscale image to binary. The idea is to separate the image into two parts; the background and foreground.