Assignment 4: CS 754, Advanced Image Processing

Group Details:
Vinit Awale (18D070067)
Piyush Bharambe (18D070019)

Question 1

- Consider a signal $\boldsymbol{x}$ which is sparse in the canonical basis and contains n elements, which is compressively sensed in the form $\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{\eta}$ where $\boldsymbol{y}$, the measurement vector, has m elements and $\boldsymbol{\Phi}$ is the $m \times n$ sensing matrix. Here $\boldsymbol{\eta}$ is a vector of noise values that are distributed by $\mathcal{N}(0, \sigma^2)$. One way to recover $\boldsymbol{x}$ from $\boldsymbol{y}, \boldsymbol{\Phi}$ is to solve the LASSO problem, based on minimizing $J(\boldsymbol{x}) \triangleq \|\boldsymbol{y} - \boldsymbol{\Phi x}\|^2 + \lambda \|\boldsymbol{x}\|_1$. A crucial issue is to how to choose λ. One purely data-driven technique is called cross-validation. In this technique, out of the m measurements, a random subset of (say) 90 percent of the measurements is called the reconstruction set R, and the remaining measurements constitute the validation set V. Thus V and R are always disjoint sets. The signal $\boldsymbol{x}$ is reconstructed using measurements only from R (and thus only the corresponding rows of $\boldsymbol{\Phi}$) using one out of many different values of λ chosen from a set Λ. Let the estimate using the g^{th} value from Λ be denoted $\boldsymbol{x_g}$. The corresponding validation error is computed using $VE(g) \triangleq \sum_{i \in V} (y_i - \boldsymbol{\Phi^i x_g})^2 / |V|$. The value of λ for which the validation error is the least is chosen to be the optimal value of λ. Your job is to implement this technique for the case when $n = 500, m = 200, \|\boldsymbol{x}\|_0 = 18, \sigma = 0.05 \times \sum_{i=1}^{m} |\boldsymbol{\Phi^i x}| / m$. Choose $\Lambda = \{5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-3}, 10^{-3}, 0.01, 0.05, 0.1, 0.5, 1, 2, 5\}$. Draw the non-zero elements of $\boldsymbol{x}$ at randomly chosen location, and let their values be drawn randomly from Uniform$(0, 1000)$. The sensing matrix $\boldsymbol{\Phi}$ should be drawn from $\pm$Bernoulli with probability of $+1$ being 0.5. Now do as follows. Use the L1-LS solver from `https://web.stanford.edu/~boyd/l1_ls/` for implementing the LASSO.

    1. Plot a graph of VE versus the logarithm of the values in Λ. Also plot a graph of the RMSE versus the logarithm of the values in Λ, where RMSE is given by $\|\boldsymbol{x_g} - \boldsymbol{x}\|_2 / \|\boldsymbol{x}\|_2$. Comment on the plots. Do the optimal values of λ from the two plots agree?
    Answer: The plot of VE and $RMSE$ versus the logarithm of the values in Λ are as follows:

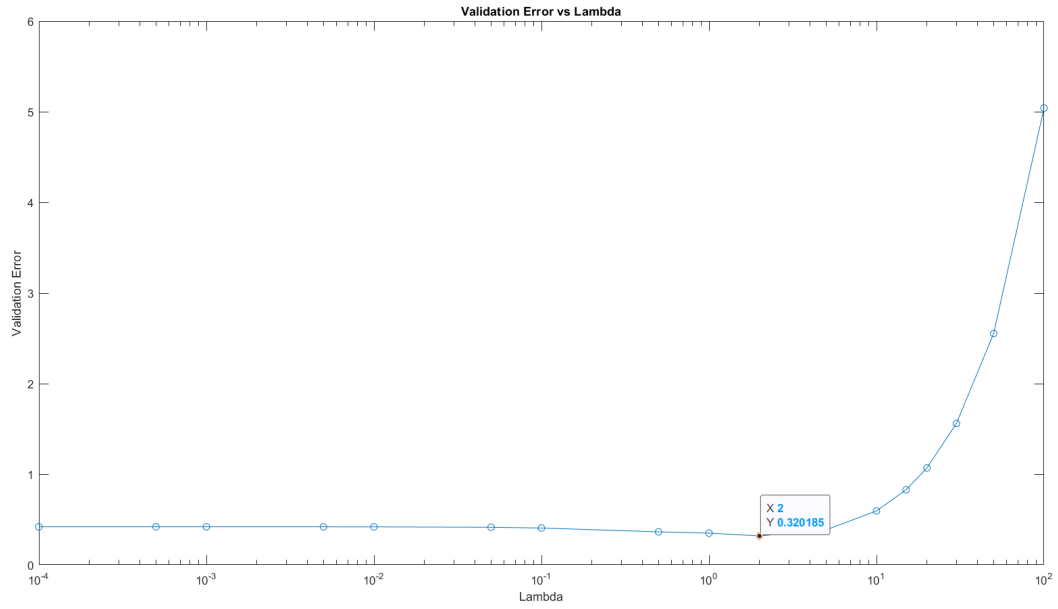

Figure 1: Plot of VE versus the logarithm of the values in Λ

From this plot we can easily see that the validation error is high for higher values of lambda. However, for the values of lambda lower than 5, the validation error is very low with very little variation. Also, the minimum validation error is obtained for $\lambda = 2$ as highlighted in the figure. Hence, the optimal value of λ is 2.

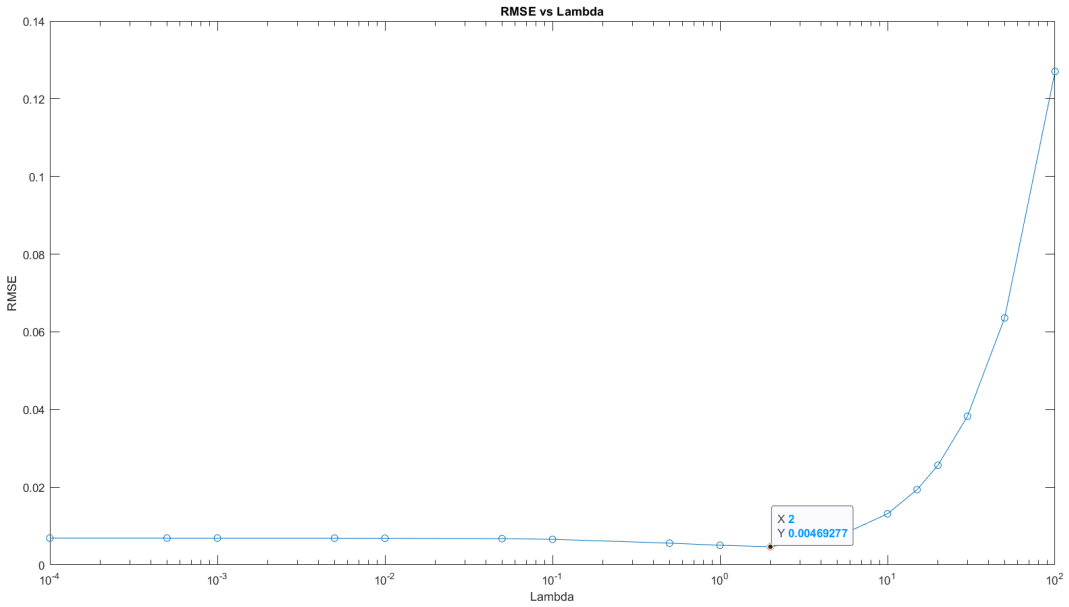

Figure 2: Plot of $RMSE$ versus the logarithm of the values in Λ

Clearly, the trend of the RMSE vs lambda plot is similar to that of the VE vs lambda plot. Similarly, the RMSE is very low for the values of λ greater than 5 and the optimal value of λ is 2.

Hence, the value of lambda that we obtain from both the methods is the same.

2. What would happen if V and R were not disjoint but coincident sets?
   Answer: If V and R are coincident sets, then we are actually attempting to find the value of lambda that minimizes the error on the reconstruction set itself. Clearly, that value of λ should be zero as by adding a positive term to $\|\boldsymbol{y} - \boldsymbol{\Phi x}\|^2$, we are effectively increasing the value of the objective function $J(\boldsymbol{x})$. Hence, the optimal value of λ that is obtained will be close to zero when V and R are coincident sets. This is the reason we need a disjoint validation set (for determining the optimal lambda) which verifies the ability of the algorithm to reconstruct using unseen data.

3. The validation error is actually a proxy for actual mean squared error. Note that you can never determine the mean squared error since the ground truth $\boldsymbol{x}$ is unknown in an actual application. Which theorem/lemma from the paper `https://ieeexplore.ieee.org/document/6854225` (On the theoretical analysis of cross-validation in compressed sensing) refers to this proxying ability? Explain how.
   Answer: In the given paper, the **Theorem 1** and **Theorem 2** prove the proxying ability of the validation error.

   Theorem 1 (Recovery error estimation) provides bounds on the recovery error ϵ_x (i.e. the actual MSE) in terms of the Cross validation (CV) residual ϵ_{CV} (i.e. the validation error). Moreover, it has been shown that this bound becomes tighter as the number of CV measurements (m_{CV}) increases. Hence if we have sufficiently large number of CV measurements, and we obtain the hyperparameters that reduce the validation error, then those hyperparameters also reduce the actual mean squared error.

   Theorem 2 (Recovery error comparison) states that if $\hat{x^p}$ and $\hat{x^q}$ are two recovered signals such that the actual MSE $\epsilon_x^p \geq \epsilon_x^q$, then the validation errors also follow $\epsilon_{CV}^p \geq \epsilon_{CV}^q$ with probability that increases as the number of CV measurements increases. Hence, if we have sufficiently large number of CV measurements, and we obtain the hyperparameters that reduce the validation error, then those hyperparameters also reduce the actual mean squared error.

   Hence, we can indeed use the validation error as a proxy for the actual mean squared error given that we have sufficiently large number of CV measurements.

4. In your previous assignment, there was a theorem from the book by Tibshirani and others which gave you a certain value of λ. What is the advantage of this cross-validation method compared to the choice of λ using that theorem? Explain.

Answer: The theorem from the book by Tibshirani and others gives us a bound on the value of λ as

$$\lambda_m > 2\frac{\phi^T \eta}{m}$$

where ϕ is a $m \times n$ matrix. Furthermore, for Gaussian noise the authors showed that the appropriate choice with high probability is

$$\lambda_m = 2\sigma\sqrt{\frac{\tau log(n)}{m}}$$

for some $\tau > 2$ and σ is the standard deviation of the noise.
However, this choice is only valid if sensing matrix ϕ satisfies the restricted eigenvalue condition with parameter τ for some cone C. The cross validation method does not require such assumption on

sensing matrix ϕ.
Moreover, checking the eigenvalue condition is computationally expensive compared to performing cross validation over a set of possible λ values. Hence, the cross validation method is more efficient.

Q2)

→ In dictionary representation, each pixel value of an image is represented as linear combination of corresponding pixels of dictionary atoms / template images (when the atoms are represented as images by reshaping columns of dictionary). Thus a linear operation on dictionary atoms is same as linear operation on image

Given a learned dictionary D containing images $\{d_k\}_{k=1,2..M}$, each of same dimension as images of class S.

Then for an image f_i in S, we have

$$f_i = \sum_{k=1}^{M} \alpha_k d_k \quad ; \text{ where } \{\alpha_k\}_{k=1}^{M} \text{ are scalar dictionary coefficients for image } f_i$$

$\vec{\alpha} = \{\alpha_k\}_{k=1}^{M}$ is sparse vector.

(a) Class S_1 consists of images obtained by applying a known derivative filter to images in class S.

Let derivative filter be H

Then for an image $x_i \in S_1$ & $f_i \in S$,

($*$ - convolution operator)

$$x_i = H * f_i$$

$$= H * \sum_{k=1}^{M} \alpha_k d_k$$

$$x_i = \sum_{k=1}^{M} \alpha_k (H * d_k) \quad \text{— \{distributive property of convolution\}}$$

Thus, all images in S_1 are sparsely represented in dictionary $D_1 = \{H * d_k \mid d_k \in D\}$

(b) Similar to part (a), rotation of an image is a linear transformation on image pixels.
Let rotation matrix be R_θ
Then for image $x_i \in S_2$ & $f_i \in S$.

$$x_i = R_\theta f_i$$
$$= R_\theta \sum_k \alpha_k d_k$$
$$x_i = \sum_{k=1}^{m} \alpha_k (R_\theta d_k) \quad ; \{\alpha_k\}_{k=1}^{m} \text{ is sparse}$$

Now an image in class S_2 is obtained by rotating image in S_1 by either α' or β
{using α' to not confuse with dictionary coefficients}
Hence, we use two dictionarys & combine them to get an overcomplete dictionary as,

$$D_2 = \underbrace{[D_{\alpha'} | D_\beta]}_{\text{(after reshaping)}}, \text{ or } D_{\alpha'} \cup D_\beta$$

where $D_{\alpha'} = \{ R_{\alpha'} d_k \mid d_k \in D \}$
& $D_\beta = \{ R_\beta d_k \mid d_k \in D \}$

(c) The given intensity transformation is non linear.

$$I^i_{new}(x,y) = \alpha' (I^i_{old}(x,y))^2 + \beta (I^i_{old}(x,y)) + \gamma$$

where α', β, γ are known
(using α' to not confuse with α_ks)

Now for image $g_i \in S_3$ & $f_i \in S$, we have

$$g_i(x,y) = \alpha' (f_i(x,y))^2 + \beta (f_i(x,y)) + \gamma$$

Now, $f_i(x,y) = \sum_{k=1}^{M} \alpha_k d_k(x,y)$

$\therefore g_i(x,y) = \alpha' \left(\sum_k \alpha_k d_k(x,y) \right)^2 + \beta \left(\sum_k \alpha_k d_k(x,y) \right) + \gamma$

$= \alpha' \left(\sum_k \alpha_k^2 d_k^2(x,y) + \sum_{i \neq j} \alpha_i \alpha_j d_i(x,y) d_j(x,y) \right)$

$+ \sum_k \alpha_k \beta d_k(x,y) + \gamma$

$= \sum_{k=1}^{m} \alpha' \alpha_k^2 d_k^2(x,y) + \sum_{1 \leq i \neq j \leq m} \alpha' \alpha_i \alpha_j d_i(x,y) d_j(x,y)$

$+ \sum_{k=1}^{m} \alpha_k \beta d_k(x,y) + \gamma . 1$

From above expression, we can say we need a dictionary which is a concatenation of the various functions of dictionary atoms appearing in the expression, namely $\{ d_k^2(x,y), d_i(x,y) d_j(x,y), d_k(x,y), 1 \}$

Since $\{\alpha_k\}_{k=1}^{m}$ is sparse, the coefficients vector $\{\alpha' \alpha_k^2\}_{k=1}^{m}, \{\alpha' \alpha_i \alpha_j\}_{i,j=1}^{m}, \{\alpha_k \beta\}_{k=1}^{m}$ are sparse too.

Now, $D_S = \{ d_k .* d_k \mid d_k \in D \}$

$D_p = \{ d_i .* d_j \mid i \neq j \text{ and } d_i, d_j \in D \}$

$D_i = \{ \mathbb{1}$, which is a vector containing all ones of same dimension as atoms of $D \}$

Then, all images in S_3 are sparsely represented by the dictionary

$D_3 = D_S \cup D_p \cup D_i \cup D$

(d) Class S_4 consists of images obtained by applying blur kernel to images in S.
Applying blur kernel is a linear operation.
For image $g_i \in S_4$ & $f_i \in S$,

$$g_i = H * f_i \quad \{H \text{ is blur kernel}\}$$
$$= H * \sum_k \alpha_k d_k$$
$$g_i = \sum_k \alpha_k (H * d_k) \quad \text{– \{distributive property\}}$$

Images in S_4 are sparsely represented by the dictionary $D_4 = \{H * d_k \mid d_k \in D\}$

(e) Consider the set of blur kernels, B

$$B = \{b_1, b_2, \dots b_p\} \; ; \; b_i \text{ is a blur kernel.}$$

On applying a blur kernel H which is a linear combination of kernels from set B, to images in S, we get images in S_5.

$$\therefore H = \sum_{i=1}^{p} \beta_i b_i \quad , b_i \in B \; , \beta_i \text{ is scalar.}$$

Then for image $g_i \in S_5$ & $f_i \in S$,

$$g_i = H * f_i$$
$$= \sum_k \alpha_k (H * d_k)$$
$$= \sum_k \alpha_k \left(\left(\sum_i \beta_i b_i \right) * d_k \right) \quad \text{– \{distributive property of convolution\}}$$
$$= \sum_k \alpha_k \left(\sum_i \beta_i (b_i * d_k) \right)$$
$$= \sum_{k=1}^{m} \sum_{i=1}^{p} \alpha_k \beta_i (b_i * d_k)$$

$\{\alpha_k \beta_i\}_{\substack{k=1 \text{ to } m \\ i=1 \text{ to } p}}$ is sparse.

Images in class S_5 are sparsely represented by the dictionary $D_5 = \{ b_i * d_k \mid b_i \in B, d_k \in D \}$

(f) Radon transform in known angle θ is a linear transform i.e.

$R_\theta(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 R_\theta(f_1) + \alpha_2 R_\theta(f_2)$

for signals f_1 & f_2.

Class S_6 consists of 1D signals obtained by applying Radon transform on images in S given angle θ. (fixed).

For $g_i \in S_6$ & $f_i \in S$,

$g_i = R_\theta(f_i)$

$g_i = R_\theta\left(\sum_k \alpha_k d_k\right) = \sum_{k=1}^{M} \alpha_k R_\theta(d_k)$.

$\therefore$ Signals in S_6 can be sparsely represented by the dictionary consisting of 1D signals

$D_6 = \{ R_\theta(d_k) \mid d_k \in D \}$

(g) Assuming appropriate zero padding & increase in size of image canvas owing to translation, translation operation is a linear operation. Similar to previous part, by using the fact that the image is a linear combination of dictionary atoms and linearity of operations, we get that shifting image is equivalent of shifting all the dictionary atoms.

$g_i = H(f_i)$; $g_i \in S_7$, $f_i \in S$, $H(\cdot)$ is shift operation

$= H\left(\sum \alpha_k d_k\right)$

$= \sum_k \alpha_k H(d_k)$ — ($H(\cdot)$ is linear)

In class S_6, images are translated by either (x_1, y_1) or (x_2, y_2) to obtain images in class S_7. So, we require two dictionaries,

$$D_a = \{ H_1(d_k) \mid d_k \in D \}$$
$$D_b = \{ H_2(d_k) \mid d_k \in D \}$$

where $H_1(.)$ is the translation operation by offset (x_1, y_1) & similarly $H_2(.)$ is by (x_2, y_2)

The coefficients $\{\alpha_k\}_{k=1}^{m}$ remain same & are sparse.

To represent images in S_7, ~~we require the~~ sparsely, we need the dictionary

$$D_7 = D_a \cup D_b$$

Question 3

- How will you solve for the minimum of the following objective functions: (1) $J(\boldsymbol{A_r}) = \|\boldsymbol{A} - \boldsymbol{A_r}\|_F^2$, where $\boldsymbol{A}$ is a known $m \times n$ matrix of rank greater than r, and $\boldsymbol{A_r}$ is a rank-r matrix, where $r < m, r < n$. (2) $J(\boldsymbol{R}) = \|\boldsymbol{A} - \boldsymbol{RB}\|_F^2$, where $\boldsymbol{A} \in \mathbb{R}^{n \times m}, \boldsymbol{B} \in \mathbb{R}^{n \times m}, \boldsymbol{R} \in \mathbb{R}^{n \times n}, m > n$ and $\boldsymbol{R}$ is constrained to be orthonormal. Note that $\boldsymbol{A}$ and $\boldsymbol{B}$ are both known.
  In both cases, explain briefly any one situation in image processing where the solution to such an optimization problem is required.

Answer:

1. We have to minimize the objective function $J(\boldsymbol{A_r})$ given by,

$$J(\boldsymbol{A_r}) = \|\boldsymbol{A} - \boldsymbol{A_r}\|_F^2$$

   such that $rank(\boldsymbol{A_r}) = r$.
   In this problem we are basically trying to find the best rank r approximation for the matrix A. The solution to this problem is given by Eckart-Young theorem.
   Suppose the singular value decomposition of the matrix A is given as,

$$\boldsymbol{A} = \boldsymbol{U\Sigma V}^T$$

   then, considering the r largest singular values and the corresponding singular vectors from U and V, we can find the rank r approximation of A as,

$$\boldsymbol{A_r} = \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^T$$

   where Σ_r contains the largest r singular values of A and $\boldsymbol{U}_r$ and $\boldsymbol{V}_r$ are the corresponding r columns of U and V.

   Application: The low rank approximation of a matrix can be used for robust patch based denoising of images/videos. Small patches from different spatial regions of an image are quite similar to each other. This similarity can be exploited to reduce the noise in the image without strong assumptions on the statistical properties of noise. Following is the link to the paper, which uses the low rank approximation to denoise an image.
   Low Rank Approximation for Image Denoising

2. In this part we have to minimize the objective function $J(\boldsymbol{R})$ given by,

$$J(\boldsymbol{R}) = \|\boldsymbol{A} - \boldsymbol{RB}\|_F^2$$

   such that R is an orthonormal matrix, i.e. $R^TR = RR^T = I$.

This is the Orthogonal Procrustes problem.
Consider,

$$\begin{aligned}\|\boldsymbol{A}-\boldsymbol{RB}\|_F^2 &= trace((\boldsymbol{A}-\boldsymbol{RB})^T(\boldsymbol{A}-\boldsymbol{RB}))\\ &= trace(\boldsymbol{A}^T\boldsymbol{A}-2\boldsymbol{A}^T\boldsymbol{RB}+\boldsymbol{BB}^T)\end{aligned}$$

Hence,

$$\begin{aligned}\min_{\boldsymbol{R},\boldsymbol{B}}\|\boldsymbol{A}-\boldsymbol{RB}\|_F^2 &= \min_{\boldsymbol{R},\boldsymbol{B}} trace(\boldsymbol{A}^T\boldsymbol{A}-2\boldsymbol{A}^T\boldsymbol{RB}+\boldsymbol{BB}^T)\\ &= \max_{\boldsymbol{R},\boldsymbol{B}} trace(\boldsymbol{A}^T\boldsymbol{RB})\\ &= \max_{\boldsymbol{R},\boldsymbol{B}} trace(\boldsymbol{RBA}^T)\ldots(\because trace(AB)=trace(BA))\\ &= \max trace(\boldsymbol{RUSV}^T)\end{aligned}$$

This last equality is using the SVD of $BA^T = USV^T$
Hence,

$$\begin{aligned}\min_{\boldsymbol{R},\boldsymbol{B}}\|\boldsymbol{A}-\boldsymbol{RB}\|_F^2 &= \max trace(\boldsymbol{SV}^T\boldsymbol{RU})\ldots(\because trace(AB)=trace(BA))\\ &= \max trace(\boldsymbol{SX})\end{aligned}$$

where $X = \boldsymbol{V}^T\boldsymbol{RU}$ is an orthogonal matrix.
Now,

$$trace(\boldsymbol{SX}) = \sum_i S_{ii}X_{ii}$$

Since, values S_{ii} are non-negative, and the above expression is maximized if $X_{ii} = 1$ all along its diagonal. As X is orthonormal, we must have

$$\begin{aligned}&\boldsymbol{X} = \boldsymbol{I}\\ \implies &\boldsymbol{V}^T\boldsymbol{RU} = \boldsymbol{I}\\ \implies &\boldsymbol{R} = \boldsymbol{VU}^T\end{aligned}$$

Application : We saw one application of this problem in Tomography. In tomography under unknown angles for 3D images, we are trying to find the best rotation matrix R such that the error in the reconstruction is minimized. In determining the best rotation matrix, we use the Orthogonal Procrustes problem.

Question 4

- We have studied the non-negative matrix factorization (NMF) technique in our course and examined applications in face recognition. I also described the application to hyperspectral unmixing. Your job is to find a research paper which explores an application of NMF in any task apart from these. You may look up the wikipedia article on this topic. Other interesting applications include stain normalization in pathology. Your job is to answer the following: (1) Mention the title, author list, venue and year of publication of the paper and include a link to it. (2) Which task does the paper apply NMF to? (3) How exactly does the paper solve the problem using NMF? What is the significance of the dictionary and the dictionary coefficients in solving the problem at hand? [15 points]

Solution:

1)

Title: A Sparse Non-Negative Matrix Factorization Framework for Identifying Functional Units of Tongue Behavior From MRI
Authors: Woo, Jonghye and Prince, Jerry L. and Stone, Maureen and Xing, Fangxu and Gomez, Arnold D. and Green, Jordan R. and Hartnick, Christopher J. and Brady, Thomas J. and Reese, Timothy G. and Wedeen, Van J. and El Fakhri, Georges
Journal: IEEE Transactions on Medical Imaging
Year: 2019
Link: https://arxiv.org/pdf/1804.05370

2)

Functional units are functional muscle groups of local structural elements within the tongue that compress, expand, and move in a cohesive and consistent manner. Identifying functional units can lead to improvement of diagnosis, treatment, and rehabilitation in patients with speech-related disorders, because it provides better understanding of the tongue function. The paper presents a new algorithm to determine local functional units that link muscle activity to surface tongue geometry during protrusion and simple speech tasks.
NMF method is used in it to give a low dimensional representation for the tongue motion features for clustering purposes.

3)

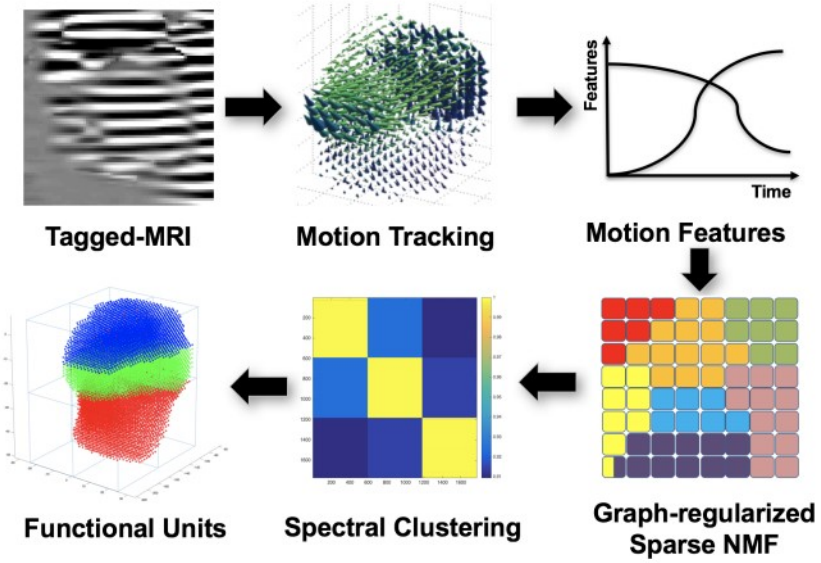

Figure 1: Overview of method

Given tagged MRI data of tongue, first by performing motion tracking motion features are extracted. The NMF method is then used to get the building blocks and weighing maps of the motion features matrix. NMF here is seen as a method to give low dimensional representation for clustering interpretation. The authors also apply a regularisation and sparsity constraint on the weighing map obtained. Further on spectral clustering, functional units are detected.

The dictionary and dictionary coefficients are referred to as building blocks and weighing maps respectively. The weighing maps holds great significance here as the paper points that it is a measure of tissue point similarity. The spectral clustering is thus done on the weighing map. The building blocks on the other hand are representatives of the low-dimensional and non-euclidean manifolds. It aspires to captures the intrinsic and geometric relation between motion quantities after solving by adding regularisation and minimizing.

QUESTION 5

Soln:)

$$y \sim \text{Poisson}\left(I_0 \exp(-Rf)\right)$$

$$\text{i.e. } y_i \sim \text{Poisson}\left(I_0 \exp\left(-(Rf)_i\right)\right)$$

Now, we calculate the Likelihood of the observed data

$$P(y \mid R, f) = \prod_{i=1}^{m} P(y_i \mid R, f)$$

... (Assuming independent measurements)

$$= \prod_{i=1}^{m} \left(\frac{e^{-I_0 \exp(-(Rf)_i)} \left(I_0 \exp^{-(Rf)_i}\right)^{y_i}}{y_i!} \right)$$

Taking negative log of the calculated likelihood

$$-\log P(y \mid R, f) = \sum_{i=1}^{m} \left(I_0 \exp\left(-(Rf)_i\right) - y_i \left(\log I_0 - (Rf)_i\right) + \log(y_i!) \right) \longrightarrow (i)$$

Prior Calculation

Assuming truncated Laplacian Prior for f (with parameter λ)

$$P(f_i) \sim C\, e^{-\lambda f_i} \quad \text{... } (C \text{ is the normalization constant})$$

$$\Rightarrow P(f) = \prod_{i=1}^{m} P(f_i)$$

Taking negative log of the prior

$$-\log P(f) = \sum_{i=1}^{m} (\lambda f_i - \log C) \longrightarrow (ii)$$

- From Bayes Theorem we know that

$$P(f|R,y) \propto P(y|R,f)\,P(f) \quad \text{---(iii)}$$

Therefore to estimate f we using Maximum Aposterion Estimation (MAP Estimation)

$$\Rightarrow \max_f P(f|R,y) \equiv \min_f \left(-\log P(f|R,y)\right)$$

Now from (iii)

$$\min_f \left(-\log P(f|R,y)\right) = \min_f \left[(-\log c') - \log P(y|R,f) - \log(f)\right]$$

... (c' is the proportionality constant in (iii))

$$= \min_f \left[-\log P(y|R,f) - \log(f)\right]$$

... (c' is independent of f)

---(iv)

Hence using (i), (ii) and (iv), we can formulate the following optimization problem to estimate f.

$$\min_f \sum_{i=1}^{m} y_i (Rf)_i + I_0 \exp\left(-(Rf)_i\right) + \lambda \sum_{i=1}^{n} f_i$$

such that $f_i \geq 0 \quad \forall \; 1 \leq i \leq n$

Now, we assume that apart from Poisson noise, there is also iid Gaussian noise with mean 0 and std. deviation σ in y.

Let the measurements corrupted with Gaussian noise be Z.

For each measurement we can write

$$z_i = y_i + \eta$$

where $\eta \sim N(0, \sigma^2)$, $y_i \sim \text{Poisson}(I_0 \exp(-Rf))$

Now, let us try to find the likelihood

$$P(z_i | y_i, R, f) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(z-y)^2}{2\sigma^2}\right)$$

$$\therefore P(z_i | R, f) = \sum_{y_i=0}^{\infty} P(z_i | y_i, R, f)\, P(y_i | R, f)$$

$$= \sum_{y_i=0}^{\infty} \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(z-y)^2}{2\sigma^2}\right) \exp\left(-I_0 \exp(-(Rf)_i)\right) \times \frac{I_0 \exp(-(Rf)_i)^{y_i}}{(y_i)!} \right]$$

we can see from the above expression that the likelihood calculation is very complicated.

Hence using MAP estimation for this problem will be very difficult.

Hence we use an alternative pre processing step. First we use a low pass filter (using a smoothing kernel) on the measured signal z. This steps helps us reduces the effect of Gaussian noise on the signal.

After this step we perform MAP estimation using the optimization problem mentioned in previous part to estimate f. This step helps us get rid of the Poisson noise.

Note: The MAP estimation will not be accurate in this case because low pass filtering will make the measurements correlated. This breaking our independent measurement assumption in Likelihood calculation in the previous part.