Assignment 4: CS 754, Advanced Image Processing

Mohit Kumar Meena - 213070021
Shashwat Pathak - 213070010

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 $VE(g) \triangleq \sum_{i \in V} (y_i - \boldsymbol{\Phi^i x_g})^2 / m$. 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 $n = 500, m = 200, \|\boldsymbol{x}\|_0 = 18, \sigma = 0.05 \times \sum_{i=1}^{m} |y_i| / m$. 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 ±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.

   (a) 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?

   Solution

   The plots have been shown below.

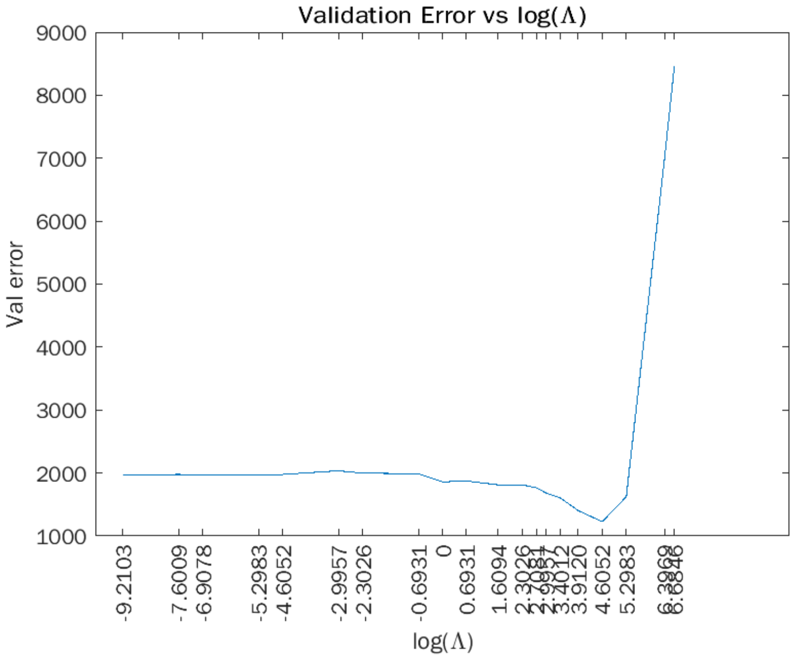

Figure 1: Figure showing plot of validation error vs $log(\Lambda)$

We can see the validation error vs the log of Λ in Figure(1) above. The validation error is stable initially at smaller values of λ, and then slowly starts to decrease as $\log(\lambda)$ crosses the zeros mark. It eventually hits a minimum in the graph and post the inflection point, the validation error shoots up. We can infer this from the RMSE curve as well (Figure(2)) that for increasing values in the set Λ, the loss keeps decreasing. This is observed because there exits a unique value of lambda for which there is sufficient penalization for the dictionary to generalize over the entire validation dataset, and can be found iteratively by looping over the dataset and computing the error for different λ.

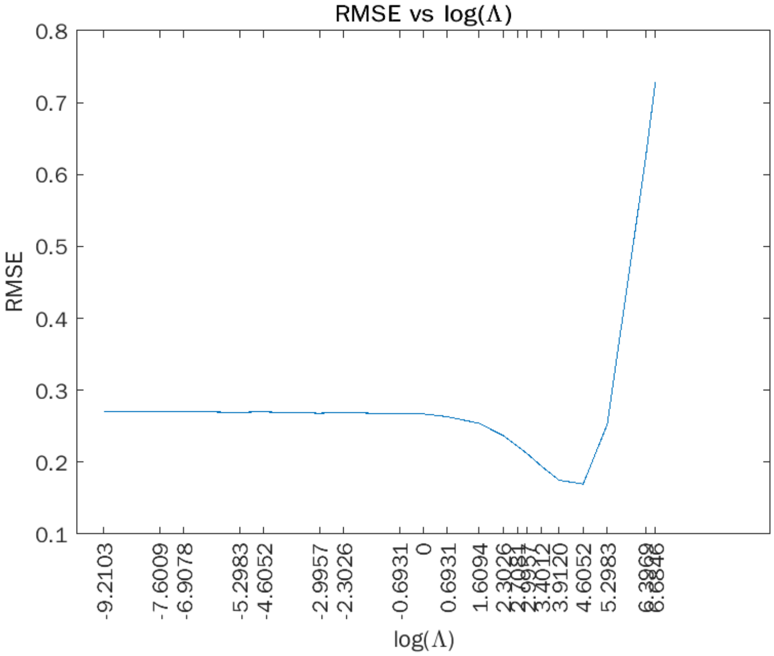

Figure 2: Figure showing plot of rmse vs $log(\Lambda)$

The optimal value of λ is also the same as inferred from both the validation error plot and the RMSE plot.

(b) What would happen if V and R were not disjoint but coincident sets?

Solution

In the case when R and V are coincident or same sets, we see that the validation error is initially very less, and then starts to increase as the regularization increases. This is intuitive, as we penalize the model for conforming too much to the dataset, there comes a point when regularization leads to divergence from good estimation.

The RMSE follows a different trend, it is close to zero which is expected as the validation error is close to the zero as well. While the vaidation error start to increase with increasing λ, the RMSE decreases, reaches an optimal value and then shoots up with the increase in lambda as shown in Figure(4)

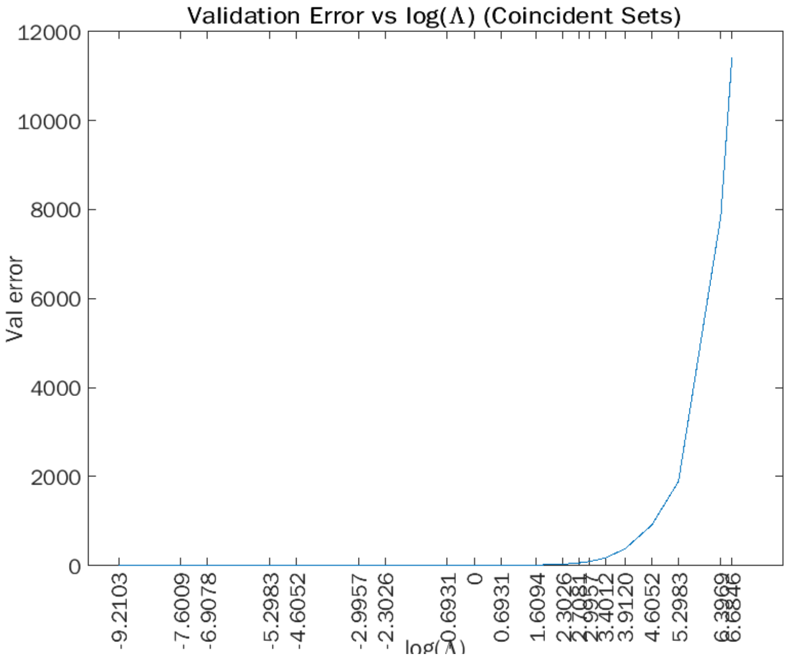

Figure 3: Figure showing plot of validation error vs $log(\Lambda)$

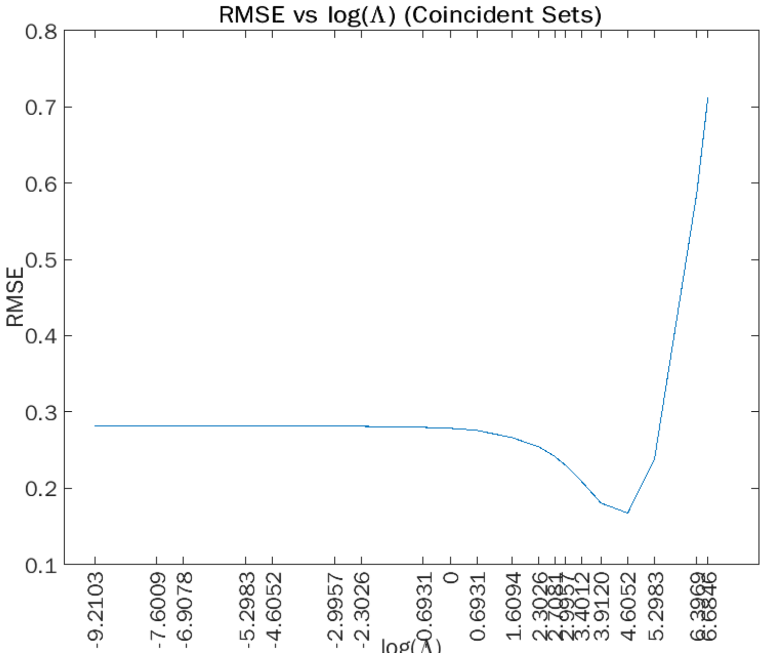

Figure 4: Figure showing plot of rmse vs $log(\Lambda)$

The optimal value of λ from both the graphs is different in this case.

(c) 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.

Solution

In the paper, it is stated that a direct consequence of **Lemma 1** is that the validation error can be used to provide an estimate of the recovery error, i.e the RMSE. This is shown more rigorously in **Theorem**

1 of the paper, which gives an error bound on the recovery error if the cross-validation error is known. As per Lemma 1, if $\hat{x}$ is the estimated signal and ϵ_x is the recovered error, if the size of the cross validation set is large enough, then

$$\epsilon_{cv} = ||\boldsymbol{y_{cv}} - \boldsymbol{Ax_{cv}}||_2^2 \sim N(\mu, \sigma^2)$$

such that, $\mu = \frac{m_{cv}}{m}(\epsilon_x + \sigma_n^2)$ and $\sigma^2 = \frac{2m_{cv}}{m}(\epsilon_x + \sigma_n^2)^2$.
Through this, we can infer that the ϵ_{cv} can be used as a bound to the actual recovery error with certain probability. This is shown more rigorously in Theorem 1 (Recovery error estimation).

(d) 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.

Solution

Theorem 11.1 gives us the value of λ as $2\sigma\sqrt{\tau\frac{logp}{N}}$ in the case of a classical gaussian linear noise model. This is obtained through the error bounds that are introduced through the Lasso regularization.

The advantage of the cross-validation based method over the theorem based method is that the cross validation method is purely data dependent. The choice of λ depends on the distribution of the data samples in the reconstruction and validation sets, rather than the recovery error.

2. Consider that you learned a dictionary $\boldsymbol{D}$ to sparsely represent a certain class $\mathcal{S}$ of images - say handwritten alphabet or digit images. How will you convert $\boldsymbol{D}$ to another dictionary which will sparsely represent the following classes of images? Note that you are not allowed to learn the dictionary all over again, as it is time-consuming.

   (a) Class $\mathcal{S}_1$ which consists of images obtained by applying a known derivative filter to the images in $\mathcal{S}$.

   (b) Class $\mathcal{S}_2$ which consists of images obtained by rotating a subset of the images in class $\mathcal{S}$ by a known fixed angle α, and the other subset by another known fixed angle β.

   (c) Class $\mathcal{S}_3$ which consists of images obtained by applying an intensity transformation $I^i_{new}(x,y) = \alpha(I^i_{old}(x,y))^2 + \beta(I^i_{old}(x,y)) + \gamma$ to the images in $\mathcal{S}$, where α, β, γ are known.

   (d) Class $\mathcal{S}_4$ which consists of images obtained by applying a known blur kernel to the images in $\mathcal{S}$.

   (e) Class $\mathcal{S}_5$ which consists of images obtained by applying a blur kernel which is known to be a linear combination of blur kernels belonging to a known set $\mathcal{B}$, to the images in $\mathcal{S}$.

   (f) Class $\mathcal{S}_6$ which consists of 1D signals obtained by applying a Radon transform in a known angle θ to the images in $\mathcal{S}$.

   (g) Class $\mathcal{S}_7$ which consists of images obtained by translating a subset of the images in class $\mathcal{S}$ by a known fixed offset (x_1, y_1), and the other subset by another known fixed offset (x_2, y_2). Assume appropriate zero-padding and increase in the size of the image canvas owing to the translation.

Solution:

2(a). It is given that dictionary D contains images d_i where $i \in \{1, 2, 3........N\}$ which is of same dimension as images in class $\mathcal{S}$. We know that differentiation is an linear operator, so we can apply known derivative filter f to the image in class **S** (denote it as s_k) to obtain image in class $\mathcal{S}_1$ (denote it as i_k).i.e.

$$i_k = f * s_k$$

we know that $s_k = \sum_{i=1}^{N} a_i d_i$ where $\{a_i\}_{i=1}^{N}$ is a sparse vector. so,

$$i_k = f * \sum_{i=1}^{N} a_i d_i$$

Now applying distributive property of convolution, we get

$$i_k = \sum_{i=1}^{N} a_i (f * d_i)$$

So the newly formed dictionary $\mathcal{D}_1$ can be obtained by applying derivative operation to each atom of initial dictionary D. i.e.

$$D_1 = \{f(d_1, f(d_2, ...f(d_N)\}$$

2(b). Similar to part (a), we can also show that rotation is also an linear operation. Consider a rotation matrix R_θ and apply the similar approach. i.e.

$$i_k = R_\theta.s_k = \sum_{i=1}^{N} a_i(R_\theta d_i)$$

But we have to careful in this case, since two different angles are provided α and β for applying rotation. This can be solved in three steps:
Step 1: Apply fixed rotation α to some selected columns of dictionary **D**, to obtain dictionary D_x.
Step 2: Apply fixed rotation β to remaining columns of dictionary **D**, to obtain dictionary D_y.
Step 3: Concatenate the dictionary $_x$ and $_y$ to obtain dictionary D_2. i.e.

$$D_2 = [D_x|D_y]$$

2(c). The intensity transformation is given as

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

This transformation is non-linear in nature. We know that,

$$s_k = \sum_{i=1}^{N} a_i d_i(x,y)$$

Now substituting these results in given non-linear transformation, we get.
I$^i_{new}(x,y) = \alpha\Bigg(\sum_{i=1}^N a_i d_i(x,y)\Bigg)^2 + \beta\Bigg(\sum_{i=1}^N a_i d_i(x,y))\Bigg) + \gamma$
$= \alpha\Bigg(\sum_{i=1}^N a_i^2 d_i^2(x,y) + 2\sum_{j\neq i} a_i a_j d_{mj}(x,y)d_{mi}(x,y)\Bigg) + \beta\Bigg(\sum_{i=1}^N a_i d_i(x,y))\Bigg) + \gamma$

So now it can be solved using similar approach as in part (b). Steps invlolved are listed below:
Step 1: Compute the dictionary D_x which contains each column squared in element wise manner. i.e.

$$D_x = \{d_1^2, d_2^2, ...d_N^2$$

Step 2: Compute the dictionary D_y which contains pairwise product of each column in element wise fashion. i.e.

$$\{didj\} \ \forall i \neq j, i, j \in \{1, 2.........N\}$$

Step 3: Compute the dictionary D_z which contains each column having ones which are then scaled by γ.
Step 4: Concatenate all the dictionaries obtained previous steps along with original dictionary **D**, to obtain dictionary D_3,

$$D_3 = [D_x|D_y|D|D_z]$$

2(d). This is almost same as part (a), since blur kernels are also linear function as that of derivative filter. So the new dictionary can be obtained by applying blur kernel function (denote it as Z(.)) to all the atoms of the original dictionary **D**. i.e.

$$D_4 = \{Z(d_1), Z(d_1), ...Z(d_N\}$$

2(e). Let us consider image i_k is obtained by applying blur kernel **Z**, which is linear combination of blur kernel belonging to a known subset β to the image s_k in **S** as menioned in the question. Now applying same approach as we did in part (a), we get

i$_k = Z * s_k$

$= \left(\sum_{x=1}^{B} \beta_x b_x \right) * s_k$

$= \left(\sum_{x=1}^{B} \beta_x b_x \right) * \left(\sum_{i=1}^{N} a_i d_i \right)$

$= \sum_{i=1}^{N} \sum_{x=1}^{b} a_i \beta_x (b_x * d_i \Big)$

Considering $b_i(D)$ to be the blur kernel applied to all the dictionary, New dictionary D_5 will be the concatenation of different blurred dictionary atoms.

$$D_5 = [b_i(D)|b_2(D)|............b_B(D)]$$

2(f). This part is similar to part (a), since radon transform is also an linear transformation. i.e.

$$\mathcal{R}\{c_1 f + c_2 g\} = c_1 \mathcal{R}\{f\} + c_2 \mathcal{R}\{g\}$$

Let us assume the usage is of dimension $m * m$ and is sparsely represented in a dictionary of dimension $m^2 * N$ such that

$$y = \sum_{i=1}^{N} a_i D_i$$

Here, $y \in \mathcal{R}^{m*m}$, $D_i \in \mathcal{R}^{m*m}$ is a dictionary matrix and a_i is a sparse coefficient. So applying linear radon operator, we get,

$$\mathcal{R}_\theta\{y\} = \mathcal{R}_\theta \left(\sum_{i=1}^{N} a_i D_i \right) = \sum_{i=1}^{N} (\mathcal{R}_\theta D_i) a_i$$

So, $D_i' = \mathcal{R}_\theta D_i$, therefore dictionary D_5 is the kronecker tensor product of $R_\theta D$ and I_{m*n}. i.e.

$$D_5 = kron(\mathcal{R}_\theta D, I_{m*n})$$

2(g). Translation is a operation of shifting image by a specified number of pixels in x and y direction. For translation, appropriate padding has to be done to produce meaningful results (e.g zero padding, mirror padding). To form a dictionary D_7 from a learned dictionary D, following steps can be applied.

Step 1: Apply zero padding to the columns of the dictionary D as per the dimension of image.

Step 2: Take a subset of **D**, let's assume it S_x. Form a translation matrix (T_1) given below:

$$T_1 = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \end{bmatrix}$$

Now apply affine transformation to each column of an subset S_x using T_1 to form dictionary D_x.

Step 3: Take remaining subset of **D**, let's assume it S_y. Form a translation matrix (T_2) given below:

$$T_2 \begin{bmatrix} 1 & 0 & x_2 \\ 0 & 1 & y_2 \end{bmatrix}$$

Now apply affine transformation to each column of an subset S_y using T_2 to form dictionary D_y.
Step 4: Concatenate D_x and D_y to form final dictionary D_7. i.e.

$$D_7 = [D_x|D_y]$$

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.

Solution:
3 (1). Solution of the objective function $J(\boldsymbol{A_r}) = \|\boldsymbol{A} - \boldsymbol{A_r}\|_F^2$ can be obtained by treating it as an low rank approximation problem. This problem has an analytic solution in terms of the singular value decomposition (SVD) of a data matrix as explained in the wiki article on Low rank approximation.

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

Decomposing A using SVD in terms of U,S and V, we get

$$A = USV^T \in \mathbf{R}^{m*n} \tag{1}$$

Here, U=$\begin{bmatrix}U_1 & U_2\end{bmatrix}$, S=$\begin{bmatrix}S_1 & 0 \\ 0 & S_2\end{bmatrix}$ and V=$\begin{bmatrix}V_1 & V_2\end{bmatrix}$. Now the minimizer A_r^* can be obtained by taking top r largest singular values as

$$A_r^* = U_1 S_1 V_1^T$$

such that

$$||\boldsymbol{A} - \boldsymbol{A_r^*}|||_F = \min_{rank(A_r \leq r} ||\boldsymbol{A} - \boldsymbol{A_r^*}|||_F$$

Application: This low rank approximation problem has several applications like **image inpainting**, matrix completion, recommender systems, etc. Talking briefly about **recommender systems** in which the matrix has missing values and let the column of data matrix be the number of products of a company and let rows be the user ratings for the products, this matrix will be highly incomplete and hence to recommend other products to user it can be solved by treating it as low rank approximation problem satisfying the various constraints.

3 (2). We know that $||Y - AX||_F^2 = \mathbf{T}((Y - AX)^T(Y - AX))$, where **T(.)** is the trace of a matrix, so using this property in given objective function, we get

$$\begin{aligned}
\min ||A - RB||_F^2 &= \min \mathbf{T}\Big((A - RB)^T(A - RB)\Big) \\
&= \min \mathbf{T}\Big((A^T - B^T R^T)(A - RB)\Big) \\
&= \min \mathbf{T}\Big((A^T A + B^T B - 2A^T RB)\Big) \qquad \text{Given } R^T R = I \\
&= \max \mathbf{T}(A^T RB) \\
&= \max \mathbf{T}(RBA^T) \qquad \text{Since } \mathbf{T}(\text{AB})=\mathbf{T}(\text{BA})
\end{aligned}$$

Now decomposing BA^T into U,S and V using SVD, we get

$$BA^T = USV^T \tag{2}$$

substituting the results obtained in (2) in the results obtained above, we get

$$\begin{aligned} \min ||A - RB||_F^2 &= \max \mathbf{T}(RBA^T) \\ &= \max \mathbf{T}(RUSV^T) \\ &= \max \mathbf{T}(V^T RUS) \\ &= \max \mathbf{T}(\mathbf{X}S) \qquad \text{Let } \mathbf{X} = V^T RU \end{aligned}$$

We know that singular values of S_{jj} is non-negative, thus the maximum can be obtained when $X = I$ or $V^T RU = I$, this results in $R = VU^T$.

Application: This objective function is used in numerous fields, like computer vision, tomography, medical imaging, etc. Specifically, in medical imaging applications, it is used in Procrustes's analysis for shape alignment, whose optimization function is given by

$$\min_{\theta,T,s} \sum_{n=1}^{N} ||Z_{1n} - sM_\theta Z_{2n} - T||_F^2 \tag{3}$$

the solution of the above problem can be obtained in closed form by solving linear set of equations or using iterative algorithms.

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?

Solution

Title : Low-complexity privacy preserving scheme based on compressed sensing and non-negative matrix factorization for image data
Author List : Jia Liang, Di Xiao, Mengdi Wang, Min Li, Ran Liu
Venue : Elsevier : Optics and Lasers in Engineering
Year of Publication : 2020
Link : `https://www.sciencedirect.com/science/article/abs/pii/S0143816619317518`

Task : The objective of this paper is to come up with an encryption and decryption strategy to encode images. Noisy compressive sensing is used to compress and encrypt the images. The basis matrix obtained through NMF is used to obtain a decoding and decryption matrix for the compressed images.

How NMF solves the problem
NMF is essentially used in the decryption and decoding stage. The owner who is sending the data, computes the decoding and the decryption matrix using NMF on the image. He also generates a noise matrix. The compressed sensing method is used to encrypt the image, noise is added to improve security. Now there are two kinds of users, the first ones, those who have access to the decoding matrix, and the second, who have access to both the decoding and the decryption matrix. The first cateogory of users can obtain the dimensional reduced data using the decoding matrix this is similar to the NMF eigenspaces. The second category of users can obtain an approximation to the original image using both the decoding and the decryption matrix. The users who have neither cannot obtain the original or the dimensional reduced data. Thus NMF solves the problem.

Significance of dictionary and dictionary coefficients
Assume, x is the original data. Now,

$$z = \Phi x + \eta$$

where, z = compressed signal, Φ = sensing matrix, and η = noise added for better security.

In case of dictionary learning (which is not used in this paper), Φ can be the learned dictionary and the x could be the sparse coefficients which represent the image in some domain. This dictionary and the sparse coding coefficients can be learned through alternating gradient descent based optimization with adaptive step size, or, through method of multiplicative updates.

Now using NMF, we know that

$$x = Wh + e'$$

here, W = basis matrix, h = column of the coefficient matrix H.

$$z = \Phi(Wh + e) = \Phi Wh + \Phi e$$

Since ΦW is full rank, we can say that the encoding matrix ΦW is pseudo-invertible. Let the pseudo-inverse be $\Phi^{\dagger}$. This is essentially the decoding matrix. Now, if we do pre matrix multiplication of the $\Phi^{\dagger}$ matrix with W, we get the decryption matrix, thus $W\Phi^{\dagger}$ is the decryption matrix which can be used to obtain the approximate results.

5. In parallel bean computed tomography, the projection measurements are represented as a single vector $\boldsymbol{y} \sim \text{Poisson}(I_o \exp(-\boldsymbol{Rf}))$, where $\boldsymbol{y} \in \mathbb{R}^m$ with m = number of projection angles × number of bins per angle; I_o is the power of the incident X-Ray beam; $\boldsymbol{R}$ represents the Radon operator (effectively a $m \times n$ matrix) that computes the projections at the pre-specified known projection angles; and $\boldsymbol{f}$ represents the unknown signal (actually tissue density values) in $\mathbb{R}^n$. If $m < n$, write down a suitable objective function whose minimum would be a good estimate of $\boldsymbol{f}$ given $\boldsymbol{y}$ and $\boldsymbol{R}$ and which accounts for the Poisson noise in $\boldsymbol{y}$. State the motivation for each term in the objective function. Recall that if $z \sim \text{Poisson}(\lambda)$, then $P(z = k) = \lambda^k e^{-\lambda}/k!$ where k is a non-negative integer. Now suppose that apart from Poisson noise, there was also iid additive Gaussian noise with mean 0 and known standard deviation σ, in $\boldsymbol{y}$. How would you solve this problem (eg: appropriate preprocessing or suitable change of objective function)?

   Solution

   In tomographic reconstruction, the projected intensity is obtained used the following formula,

$$I = I_0 \exp(-\boldsymbol{Rf})$$

   Thus, given $\boldsymbol{y} \sim \text{Poisson}(I_o \exp(-\boldsymbol{Rf}))$ is basically, $\boldsymbol{y} \sim \text{Poisson}(I)$. Using filtered back projection on Y, we get the Poisson corrupted noisy image. Now we can represent this image as a product of a non negative basis matrix and its corresponding coefficient matrix. The basis matrix W, and the corresponding coefficient matrix H can be obtained using NMF. The following loss function needs to be minimized with respect to the corrupted image obtained through filtered back projection. The minimization approach is alternating gradient descent optimization using adaptive gradient descent step.

$$E(W, H) = \sum_{k=1}^{N}\sum_{l=1}^{N} -y_{kl}log(Wh_k)_l + (Wh_k)_l + \rho\sum_{l=1}^{N}\sum_{c=1}^{r} h_{cl}$$

   such that, $W \geq 0$ and $H \geq 0$, $w_i^t w_i = 1$ for $1 \leq i \leq r$.

   In the case, when the signal is corrupted with both Gaussian and Poisson noise, we will need an overcomplete dictionary matrix such $A = [W|\Phi]$, where W is the basis obtained through NMF, and Φ is a generated as a random Bernoulli matrix and is updated through projected gradient descent on mutual coherence on columns of Φ and maintaining a non-negativity constraint. After learning the matrix Φ, we learn the matrix W and the sparse codes for the combined matrix A using alternating projected gradient descent with adaptive learning rate.