Advanced Image Processing HW5

Anilesh Bansal (22b0928) Parth Pujari (210100106)

1 Question 1

Let $\mathcal{D} = \{\mathbf{d_i}\}$ be the set of dictionary vectors and let any image v be a linear combination of these dictionary vectors such that

$$\mathbf{v} = \sum a_i \mathbf{d_i}$$

a) Let f be the derivative operator on the image v. Since a derivative operator is linear, we get that

$$f\mathbf{v} = f\sum a_i \mathbf{d_i} = \sum a_i f \mathbf{d_i}$$

$$\implies \mathcal{D}' = \{f\mathbf{d_i}\}$$

Dictionary made by applying the derivative filter on $\mathcal{D}$ works

b) Let f_α be the rotation operator on a vector v that rotates it by angle α. Since f is linear, we get that

$$f_\alpha\mathbf{v} = f_\alpha\sum a_i \mathbf{d_i} = \sum a_i f_\alpha \mathbf{d_i}$$

$$\implies \mathcal{D}' = \{f_\alpha\mathbf{d_i}\}\bigcup\{f_\beta\mathbf{d_i}\}$$

Dictionary made by union of two dictionaries - one with rotation by angle α on it's vectors, the other wit rotation angle β

c)

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

$$\implies v' = \alpha(\sum a_i\mathbf{d_i})^2 + \beta\sum a_i\mathbf{d_i} + \gamma$$

$$\mathcal{D}' = \{d_i * d_j \text{ for } 1 \leq i,j \leq |\mathcal{D}|\}\bigcup\{\mathbf{d_i}\}\bigcup\{\mathbf{1}\}$$

works. Here $*$ is element-wise product of two $n \times 1$ vector d_i and d_j to give another $n \times 1$ vector. And $\mathbf{1}$ denotes a vector with all entries 1, and of the same dimension as vectors in $\mathcal{D}$

d) Let f be the blur kernel. Since f is linear in nature, we get that

$$f\mathbf{v} = f\sum a_i \mathbf{d_i} = \sum a_i f \mathbf{d_i}$$

$$\implies \mathcal{D}' = \{f\mathbf{d_i}\}$$

Dictionary made by applying the blur kernel on each vector in $\mathcal{D}$ works

e) Let f_i be the i^{th} blur kernel in set $\mathcal{B}$. Let $f = \sum b_i f_i$ be the blur kernel. Since each f_i is linear in nature, we get that

$$f\mathbf{v} = f\sum a_i \mathbf{d_i} = \sum a_i f \mathbf{d_i} = \sum\sum a_i b_j f_j d_i$$

$$\implies \mathcal{D}' = \bigcup_{f\in\mathcal{B}}\{fd_i\}$$

Dictionary made by applying the blur kernels in $\mathcal{B}$ on each vector in $\mathcal{D}$ and concatenating all of them together works.

f) Let R_θ be the Radon Transform operator.

$$
\begin{aligned}
R_\theta(f(\rho)+g(\rho)) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(f(x,y)+g(x,y))\delta(xcos\theta+ysin\theta-\rho)dxdy \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(xcos\theta+ysin\theta-\rho)dxdy + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)\delta(xcos\theta+ysin\theta-\rho)dxdy \\
&= R_\theta f(\rho) + R_\theta g(\rho)
\end{aligned}
$$

Since R_θ is linear, we get that

$$
\begin{aligned}
R_\theta \mathbf{v} = R_\theta \sum a_i \mathbf{d_i} &= \sum a_i R_\theta \mathbf{d_i} \\
\implies \mathcal{D}' &= \{R_\theta \mathbf{d_i}\}
\end{aligned}
$$

Dictionary made by applying the radon transform on each vector in $\mathcal{D}$ works

g) Let $f(x_1,y_1)$ be the translation operator on a vector v that rotates it by angle α. Since f is linear, we get that

$$
\begin{aligned}
f(x_1,y_1)\mathbf{v} = f(x_1,y_1)\sum a_i \mathbf{d_i} &= \sum a_i f(x_1,y_1)\mathbf{d_i} \\
\implies \mathcal{D}' &= \{f(x_1,y_1)\mathbf{d_i}\}\bigcup\{f(x_2,y_2)\mathbf{d_i}\}
\end{aligned}
$$

Dictionary made by union of two dictionaries - one with translation by (x_1,y_1) on it's vectors, the other with translation by (x_2,y_2)

2 Question 2

(1) The following objective function, $J(\mathbf{A_r}) = ||\mathbf{A}-\mathbf{A_r}||_F^2$ can be minimized using Singular value thresholding to find a low rank approximation of $\mathbf{A}$ until the approximation reaches a rank r.

An example situation where the solution to this problem is required is in image compression where a low rank approximation of an image could be stored as its approximation in its eigen basis instead.

(2) The objective $J(\mathbf{R}) = ||\mathbf{A}-\mathbf{RB}||_F^2$ is minimized using **Orthogonal Procrustes Algorithm**. The solution to it is UV^T where $U\Lambda V^T$ is the SVD for AB^T.

A situation where the solution to this problem could be required could be in image comparison or registration, to align or compare 2 images taken from the same scene from different angles or viewpoints and to compare them to check if they were indeed simply a rotation or reflection apart since an orthogonal matrix simulates rotation or reflection.

3 Question 3

3.1 Details of the Paper

Title: Wind Noise Reduction using Non-negative sparse coding

Authors: Mikkel N. Schmidt, Jan Larsen, Fu-Tien Hsiao

Published in: Proceedings of the 2007 IEEE Signal Processing Society Workshop, MLSP

Link: https://backend.orbit.dtu.dk/ws/portalfiles/portal/3848474/Schmidt.pdf

3.2 Description of the Problem

The paper uses NMF to do speech denoising under non-stationary noise. The noisy signal $x(t)$ can be written as

$$x(t) = s(t) + n(t)$$

where $s(t)$ is the speech signal, and $n(t)$ is the wind noise.

The aim is to seperate out $n(t)$ and retrieve the signal $s(t)$

3.3 The Algorithm

The algorithm for NMF denoising goes as follows. Two dictionaries, one for speech and one for noise, need to be trained offline. Once a noisy speech is given, we first calculate the magnitude of the Short-Time-Fourier-Transform. Second, separate it into two parts via NMF, one can be sparsely represented by the speech dictionary, and the other part can be sparsely represented by the noise dictionary. Third, the part that is represented by the speech dictionary will be the estimated clean speech.

$$X = X_s + X_n \approx [D_s D_n][H_s H_n]^T$$

Update Steps (till convergence):

$$H_s \longleftarrow H_s \circ \frac{D_s^T X}{D_s^T DH + l_s}$$

$$H_n \longleftarrow H_n \circ \frac{D_n^T X}{D_n^T DH + l_n}$$

$$D_s \longleftarrow \frac{XH_S^T + D_s \circ (\mathbf{1}(DHH_s^T \circ D_s))}{DHH_s^T + D_s \circ (\mathbf{1}(XH_s^T \circ D_s}))$$

Here D_s, H_s and D_n, H_n represent the dictionary and coded values for speech and noise respectively. $\mathbf{1}$ represents a matrix of suitable size with all elements equal to 1.

3.4 Importance of the Dictionary

The key idea is that clean speech signal can be sparsely represented by a speech dictionary, but non-stationary noise cannot. Similarly, non-stationary noise can also be sparsely represented by a noise dictionary, but speech cannot.

The dictionaries represent the differences between speech and wind noise, and that is the main reason why we are able to separate them out. Both speech and noise can be well characterized by dictionaries with low sparsity.

4 Question 4

$$\mathbf{y} \sim \text{Poisson}(I_0 e^{-Rf}) \implies y_i \sim \text{Poisson}(I_0 e^{-R_i f})$$

Let $\mathbf{f}$ be sparse in some basis $\mathbf{\Psi}$ such that $\mathbf{f} = \mathbf{\Psi}\theta$

The cost function for the above is given by the negative-log-likelihood term that maximizes $P(y|f)$ and a regularizer term to induce sparsity in f.

$$\begin{aligned}
NLL &= -\sum_{i=1}^{m} \log(P(y_i|f)) \\
&= -\sum_{i=1}^{m} \log((I_0 e^{-R_i f})^{y_i}) + \log(e^{-(I_0 e^{-R_i f})}) - \log(y_i!) \\
&= \sum_{i=1}^{m} \Big(-y_i \log(I_0) + y_i R_i f + I_0 e^{-R_i f} + \log(y_i!)\Big) \\
&= \sum_{i=1}^{m} \Big(y_i R_i f + I_0 e^{-R_i f}\Big)
\end{aligned}$$

where in we can ignore the terms $y_i \log(I_0)$ and $log(y_i!)$ since they are constant w.r.t f

The final cost function is given by

$$J(f) = \sum_{i=1}^{m} \Big(y_i R_i f + I_0 e^{-R_i f}\Big) + \rho||\theta||_1 \text{ such that } \Psi\theta \succeq 0$$

If Gaussian Noise is added into our measurements then

$$\mathbf{y} \sim \text{Poisson}(I_0 e^{-Rf}) + \eta \implies y_i \sim \text{Poisson}(I_0 e^{-R_i f}) + \eta_i$$

where $\eta_i \sim \mathcal{N}(0, \sigma^2)$

$$P(y_i|f) = \sum_{\alpha=0}^{\infty} P(\text{Poisson}(I_0 e^{-Rf}) = \alpha) * P(Gaussian(0, \sigma^2) = y_i - \alpha)$$

$$P(y_i|f) = \sum_{\alpha=0}^{\infty} \frac{\lambda_i^{\alpha} e^{-\lambda_i}}{\alpha!} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i-\alpha)^2}{2\sigma^2}}$$

where $\lambda_i = I_0 e^{-R_i f}$. (essentially the convolution of PDF of poisson and gaussian PDF's)

$$NLL = -\sum_{i=1}^{m} \log(P(y_i|f))$$

$$\implies J(f) = NLL + \rho||\theta||_1 \text{ such that } \Psi\theta \succeq 0$$

5 Question 5

Given measurements $y = \Phi x + \eta$, the MAP estimate for x given y, Σ_x, Φ is the maximum of the probability distribution function $p(x|y, \Sigma_x, \Phi)$. Applying Bayes' rule we have:

$$\begin{aligned}
p(x|y, \Sigma_x, \Phi) =& \frac{p(y, \Sigma_x, \Phi|x)p(x)}{p(y, \Sigma_x, \Phi)} \\
p(x|y, \Sigma_x, \Phi)p(y, \Sigma_x, \Phi) =& p(x, y, \Sigma_x, \Phi) \\
=& p(y|x, \Sigma_x, \Phi)p(x, \Sigma_x, \Phi)
\end{aligned}$$

But $p(x, \Sigma_x, \Phi) = p(x|\Sigma_x, \Phi)p(\Sigma_x, \Phi)$ and $p(y, \Sigma_x, \Phi) = p(y|\Sigma_x, \Phi)p(\Sigma_x, \Phi)$. Making the substitutions we have,

$$p(x|y, \Sigma_x, \Phi) = \frac{p(y|x, \Sigma_x, \Phi)p(x|\Sigma_x, \Phi)}{p(y|\Sigma_x, \Phi)}$$

We know that η comes from a standard Gaussian distribution so $p(y|x, \Sigma_x, \Phi) = Ke^{-\frac{1}{2\sigma^2}||y-\Phi x||_2^2}$ $p(x|\Sigma_x, \Phi)$ is technically independent of Φ and it is $K_1 e^{-\frac{1}{2}x^T\Sigma_x^{-1}x}$. Lastly $p(y|\Sigma_x, \Phi)$ is independent of x and hence is not required for maximization. Putting it together we have;

$$\begin{aligned}
p(x|y, \Sigma_x, \Phi) =& K_3 \exp\{-\frac{1}{2}(||y - \Phi x||_2^2/\sigma^2 + x^T\Sigma_x^{-1}x)\} \\
\log(p(x|y, \Sigma_x, \Phi)) =& -\frac{1}{2}(||y - \Phi x||_2^2/\sigma^2 + x^T\Sigma_x^{-1}x) + C
\end{aligned}$$

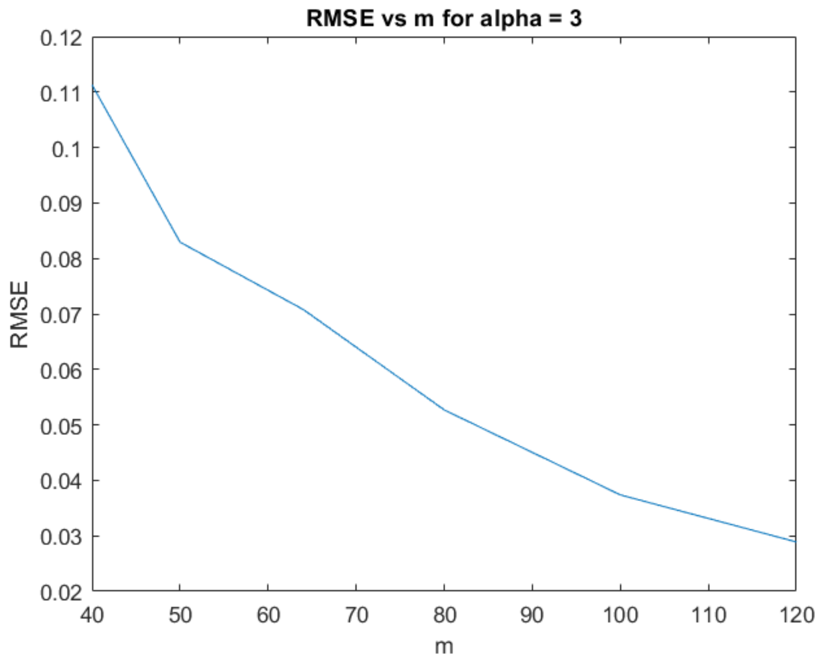
Taking the derivative wrt x and equating to 0 we have;

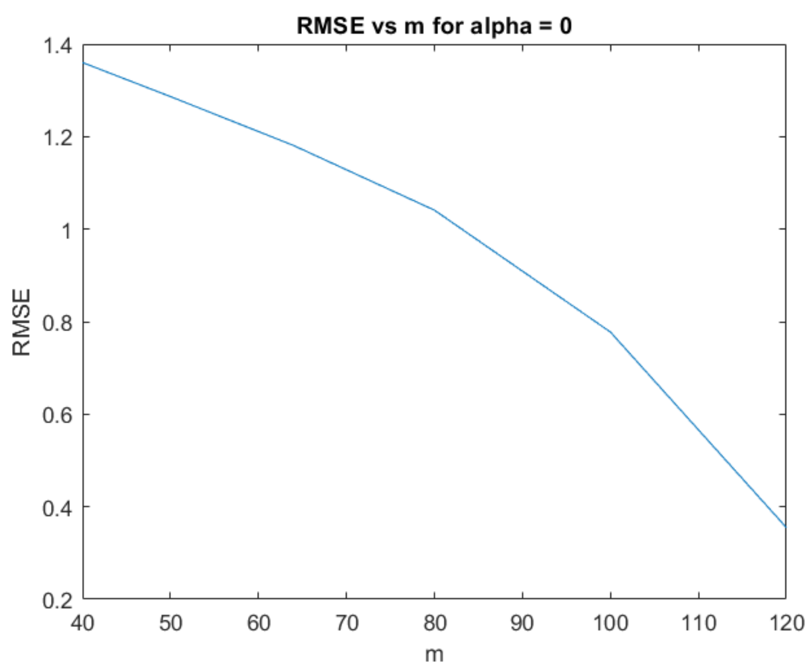
$$\begin{aligned}
0 =& -\frac{1}{\sigma^2}(-\Phi^T y + \Phi^T\Phi x) - \Sigma_x^{-1}x \\
0 =& \Phi^T y - (\Phi^T\Phi + \sigma^2\Sigma_x^{-1})x \\
x =& (\Phi^T\Phi + \sigma^2\Sigma_x^{-1})^{-1}\Phi^T y
\end{aligned}$$

Thus, the MAP estimate of x is:

$$x = (\Phi^T\Phi + \sigma^2\Sigma_x^{-1})^{-1}\Phi^T y \tag{1}$$

The following page has the results from the code.
A higher value of alpha gives a lesser average RMSE because for a higher decay in eigenvalues of Σ_x, x becomes sparser in the eigenbasis making reconstruction easier.


RMSE vs m for alpha = 3
RMSE
m
0.12
0.11
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
40
50
60
70
80
90
100
110
120


RMSE vs m for alpha = 0
RMSE
m
1.4
1.2
1
0.8
0.6
0.4
0.2
40
50
60
70
80
90
100
110
120