In the Name of God

Introduction to Machine Learning (25737-2)

Project Phase 2

Spring Semester 1401-02

Department of Electrical Engineering

Sharif University of Technology

Instructor: Dr. S. Amini

Due on Khordad 20, 1402 at 23:55

(*) Should you have any questions concerning the project, please feel free to ask via Telegram.

1 Abstract

In the previous phase we have learned how to find parameters of a GMM distribution using EM Algorithm. In this phase we want to apply this algorithm to infer "clean" image from corrupted images (in this case, MNIST dataset). This process is known as "Regularization Inverse Problem". We will solve this problem in multiple steps.

2 Implementation

Step 1: Data Preparation

In the first step we need to extract numerous patches from each image. In order to do that, consider a $m \times m$ square at the upper left corner of an image. This is the first patch of the image. Next, sweep that square one pixel to the right. This is the second patch. Continue this until you reach the upper right corner of the image. Then, place the square at the upper left corner again, this time shift this corner one pixel down. Continue this procedure until you sweep all the image and now you have extracted all of the patches (each $m \times m$ pixels), with maximum overlap (each time you have shifted the window by 1 pixel.)

Simulation Question 1. Extract patches of some 1000 images from the train part of the MNIST dataset, then vectorize each patch in order to get $m^2 \times 1$ vectors and finally save these patches in an array. One may assume that these patches are samples of a patch distribution which we will call $\mathbf{Z}$. Consider patch size as a hyperparameter (set $m = 8$ for this section but at end you must fine-tune your model by setting $m = \{4, 8, 12, 16, 20, 28\}$) and find the best patch size after completing the denoising algorithm.

Step 2: Denoising algorithm

We will assume that the distribution of patches ($\mathbf{Z}$) is a GMM with K components.

Simulation Question 2. Use EM algorithm (which probably you have already implemented in previous phase of project) to find parameters of this GMM. You may consider K (number

of components) as a hyperparameter and use validation set to fine-tune it, so write your code such that it would be easy for you to change K.

In the next step, you are given some corrupted images as your test dataset. These images are corrupted by two methods, namely, blurring and additive noise. We will define $\mathbf{Y}$ as random variable indicating corrupted patches. We know that each clean patch is corrupted by first getting multiplicated by a blurring (wieght) matrix, $\mathbf{W}$ (which is attached in files) and then we have added some normal noise to it. So we may infer that the conditional distribution of corrupted images given a clean image is:

$$p_{\mathbf{Y}|\mathbf{Z}}(\mathbf{y}|\mathbf{z}) = \mathcal{N}(\mathbf{W}\mathbf{z}, \sigma^2\mathbf{I}) \tag{1}$$

Where σ^2 is variance of the added noise.(suppose that σ is among 5, 10, 20, 30, 50 and you have to find best σ for the dataset using cross validation. We want to infer what the clean image is after getting the noisy image so we have to calculate the posterior probability which is proportional to multiplication of $p_{\mathbf{Y}|\mathbf{Z}}$ and prior probability of patches ($p_{\mathbf{Z}}$)

Theory Question 1. if we have normal prior distribution $p_{\mathbf{Z}}(\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{z}})$ and normal likelihood distribution $p_{\mathbf{Y}|\mathbf{Z}}(\mathbf{y}|\mathbf{z}) = \mathcal{N}(\mathbf{W}\mathbf{z} + \mathbf{b}, \boldsymbol{\Sigma}_{\mathbf{Y}|\mathbf{Z}})$, show that joint distribution will be Normal and find its parameters $(\boldsymbol{\mu}_{\mathbf{Y},\mathbf{Z}}, \boldsymbol{\Sigma}_{\mathbf{Y},\mathbf{Z}})$.

$$p_{\mathbf{Z},\mathbf{Y}}(\mathbf{z},\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{Z},\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{Z},\mathbf{Y}}) \tag{2}$$

$$\boldsymbol{\mu}_{\mathbf{Z},\mathbf{Y}} = \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{Z}} \\ \mathbf{W}\boldsymbol{\mu}_{\mathbf{Z}} + \mathbf{b} \end{bmatrix}$$

$$\boldsymbol{\Sigma}_{\mathbf{Z},\mathbf{Y}} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{Z}} & \boldsymbol{\Sigma}_{\mathbf{Z}}\mathbf{W}^T \\ \mathbf{W}\boldsymbol{\Sigma}_{\mathbf{Z}} & \boldsymbol{\Sigma}_{\mathbf{Y}|\mathbf{Z}} + \mathbf{W}\boldsymbol{\Sigma}_{\mathbf{Z}}\mathbf{W}^T \end{bmatrix}$$

Show that the posterior distribution is also as folows:

$$p_{\mathbf{Z}|\mathbf{Y}}(\mathbf{z}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{Z}|\mathbf{Y}}, \boldsymbol{\Sigma}_{\mathbf{Z}|\mathbf{Y}}) \tag{3}$$

$$\boldsymbol{\mu}_{\mathbf{Z}|\mathbf{Y}} = \boldsymbol{\Sigma}_{\mathbf{Z}|\mathbf{Y}}[\mathbf{W}^T\boldsymbol{\Sigma}_Y^{-1}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Sigma}_{\mathbf{Z}}^{-1}\boldsymbol{\mu}_{\mathbf{Z}}]$$

$$\boldsymbol{\Sigma}_{\mathbf{Z}|\mathbf{Y}}^{-1} = \boldsymbol{\Sigma}_{\mathbf{Z}}^{-1} + \mathbf{W}^T\boldsymbol{\Sigma}_{\mathbf{Y}}^{-1}\mathbf{W}$$

Theory Question 2. Assume that prior distribution is GMM and $p_{\mathbf{Y}|\mathbf{Z}}$ is a normal distribtion. Is posterior distribution normal? Is it also a GMM? If it is GMM, what are parameters of this GMM if we know parameters of the aforementioned distributions?

Theory Question 3. Why GMM is prefered prior distribution for $\mathbf{Z}$?

The goal of a denoising algorithm is estimating the clean image after seeing corrupted version of it. So we need to calculate the MAP estimate for $p_{\mathbf{Z}|\mathbf{Y}}$. Calculating the MAP estimation for a GMM does not have a closed form solution. We use an approximation to find an estimated solution for MAP of a GMM. We assume that the answer to this problem is that to first find the component with maximum probability ($\pi_k \cdot p_{\mathbf{Z}|\mathbf{Y},k}(\mathbf{z}|\mathbf{y}, k)$) and then find the MAP estimation of $p_{\mathbf{Z}|\mathbf{Y},k}(\mathbf{z}|\mathbf{y}, k)$, which in case of a normal distribution, is equivalent to its mean.

Simulation Question 3. Use the algorithm that we have just discussed to denoise patches of the test dataset (recall that you have vectorized image patches.) Reshape the resulted (denoised) patches to $m \times m$ size. Reconstruct the denoised images from denoised patches so that pixels that are in multiple patches get intensity equall to average of those patches that

include this specific pixel. For example the upper left pixel of an image is only in one patch but the center pixel is in 64 patches and should have intensity equall to average of pixels at position of this pixel.

Simulation Question 4. Plot some of noisy images and the denoised version. Sweep hyper-parameters of the model, namely, K, m and σ to get better results. Plot the final results and discuss whether the first order approximation that we used is working for this problem or not.