MSSC 6040 - Homework 2

Henri Medeiros Dos Reis

November 18, 2022

Instructions: To get started on Problems 1-4, unzip the folder `hw5` from the zip file `hw5.zip`. Inside the `hw5` folder you will find the scripts `problem1.m`, `problem2.m`, etc., which you should use to get started on each of the problems below. For these scripts to run you will need to have the *image processing toolbox* installed in MATLAB (to check if it is installed, type `ver` into the command line in MATLAB and see if "Image Processing Toolbox" shows up in the list of installed toolboxes; if not, install it by going to the "Apps" tab, and choose "Get More Apps"). Finally, be sure that the "current folder" in MATLAB as shown on the left-hand pane of the interface is set to the `hw5` folder, otherwise some scripts may not run correctly.

Problem 1 (5 pts, MATLAB). *Linear Regression.*
For this problem you will use linear regression to predict the date at which different bird species begin migration. Assume $a_1, ..., a_n$ are a collection of predictors that may be related to migration patterns for a given species, e.g., breeding latitude, wing-span, etc. Let b represent the median date of migration for that species, where $b = 1$ means April 1st, $b = 2$ means April 2nd and so on. We will fit a linear model of the form

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \approx b.$$

Here the x_i coefficients are unknown and can be learned from a collection of "training examples" of bird species where we know $a_1, ..., a_n$ and b. We will use 7 predictors $a_1, ..., a_7$ plus a constant predictor $a_8 = 1$; see `bird_migration.txt` for a description of these predictors.

Collecting all the predictor values for the different species into a matrix A and the responses into a vector b, the best fit coefficients $x = (x_1, x_2, ..., x_8) \in \mathbb{R}^8$ are the solution to the least squares problem:

$$\min_{x\in\mathbb{R}^n} \|Ax - b\|_2^2.$$

Tasks:

(a) In the script `problem1.m`, find the regression coefficients $x \in \mathbb{R}^8$ that solve the above least squares problem (Note: A and b are already defined for you in the script). Include the code snippet you wrote (should just be one or two lines) in your write-up.

(b) Examine the regression coefficients you found in part (a). Based on the values of the coefficients, what predictors are positively correlated with a later migration date (i.e., which coefficients have a positive sign)? What predictors are negatively correlated with a later migration date (i.e., which coefficients have a negative sign)? What predictors have negligible impact on migration date (i.e., which coefficients are close to zero)? You may ignore the constant predictor a_8 in this analysis.

(c) Use the regression coefficients you found in part (a) to predict the migration date for a hold-out set of five species. The predictor matrix for these species is provided in the variable `A1`. How do your predictions compare to the known values given in the vector `b1`? Among the five species, what is the largest absolute difference between the predicted and actual migration date (in days)? What is smallest absolute difference in migration date? And what is the mean absolute difference in migration date?

Solution 1. .

a-)

```
% just using matlab's backslash solution
x_ls = A\b;
```

b-)

$$x_{ls} = \begin{bmatrix} 0.1293 \\ -0.0710 \\ 0.1071 \\ -3.9731 \\ 2.1873 \\ -0.3728 \\ 0.3812 \\ 23.2828 \end{bmatrix}$$

It is possible to see that the coefficients x_1, x_3, x_5, x_7, x_8 are positively correlated with a latter migration date, while x_2, x_4, x_6 are negatively correlated with a latter migration date.

Predictors x_1, x_2 and x_3 seem to have negligible impact on migration date.

c-)

```
pred1 = A1*x_ls;
res = abs(pred1-b1);
disp(res)
mini = min(res);
```

```
maxi = max(res);
mean1= mean(res);
disp(strcat(min:  ,sprintf('%.6f',mini),  max:  , ...
    sprintf('%.6f',maxi),  mean:  , sprintf('%.6f',mean1)))
%%----------------OUTPUT --------------------
%    7.9383
%    4.7079
%    4.6807
%    3.5958
%    3.1180

%min: 3.118012 max: 7.938278 mean: 4.808130
```

The differences between the predicted values and the true values is pretty close for most values, only for the first specie the value is a little bit large.

We can see that the minimum difference in absolute value is in the prediction for the fifth specie. The largest difference is in the first specie. And the mean difference is 4.8.

Problem 2 (5 pts, MATLAB). *Image Denoising.*
A simple approach to denoising images is to use the following regularized least squares (RLS) formulation:

$$\min_{x\in\mathbb{R}^n} \|x-b\|_2^2 + \lambda\|Lx\|_2^2$$

where $b \in \mathbb{R}^n$ is the noisy image, $\lambda > 0$ is a parameter, and L is a matrix computes all finite differences in the horiztonal and vertical directions.

Tasks:

(a) Compute the solution of the RLS problem above using the script `problem2.m` as a starting point (Note: I, L, and b are already defined for you in the script.) Include the code you wrote your report (should just be a few lines).

(b) Denoise the provided image for three choices of the regularizaiton parameter $\lambda > 0$:

- Choose λ such that too little noise is removed.
- Choose λ such that too much noise is removed (image is over-smoothed).
- Choose λ that gives the best visual result, according to your eye.

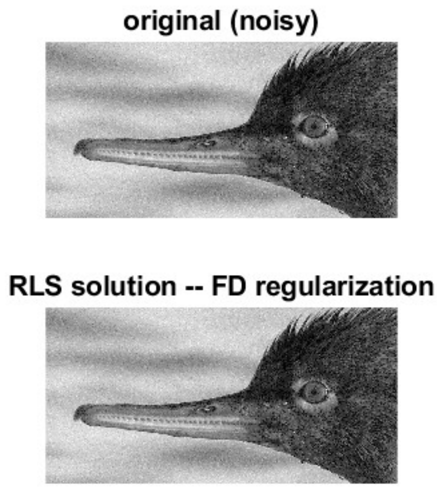
Include the particular choices of λ and the resulting denoised images in your write-up.

Solution 2. .

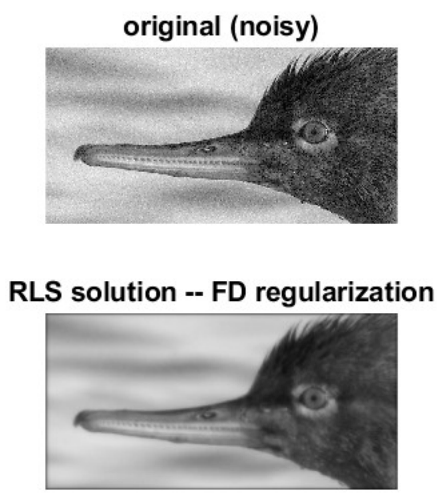
a-)
```
% Use the a modified version of the normalequations
lambda = 1;
x_rls = (I+(lambda*(L'*L)))\b;
```

b-) Images and λ:

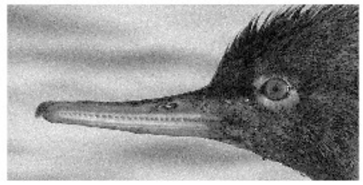
$\lambda = 0.01$



$\lambda = 15$



$\lambda = 1.2$



Problem 3 (5 pts, MATLAB). *Image Deblurring.*
Image deblurring can be posed as a least squares problem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \lambda \|x\|_2^2$$

where $A \in \mathbb{R}^{n \times n}$ is the blur matrix, $b \in \mathbb{R}^n$ is the blurred image, $\lambda > 0$ is a parameter.

Tasks:

(a) Compute the solution of the RLS problem above using the script `problem3.m` as a starting point (Note: A, I, and b are already defined for you in the script). Include the code snippet you wrote in your report (should just be a few lines).

(b) Deblur the provided image using three choices of the regularization parameter $\lambda \geq 0$:

- Choose $\lambda = 0$ (no regularization). What do you observe?
- Choose $\lambda > 0$ such that the image is sharper but there is too much noise.
- Choose λ that gives the best visual result, according to your eye.

Include the choice of λ you used and the resulting deblurred images in your report.

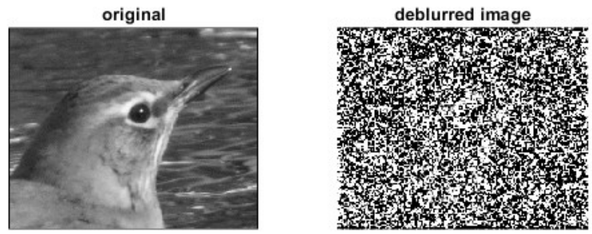
Solution 3. .

a-)

```
% Use a modified version of the normal equations
lambda = 1;
x_rls = ((A'*A)+(lambda*I))\(A'*b);
```

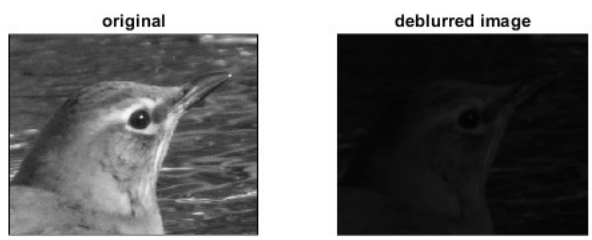
b-) Images and λ:

$\lambda = 0$

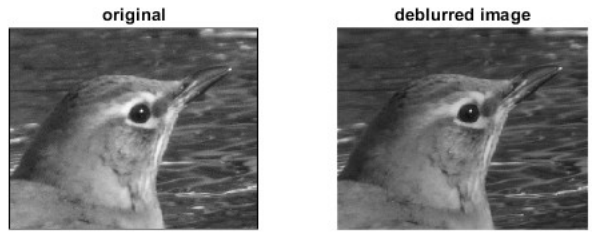


We can see that the result with $\lambda = 0$ is not even close to a good result. Instead of deblurring the image, it lookis like it is just picking random color for pixels. Since we are using $\lambda = 0$, this makes the solution just equal to a least squares problem, which does not do a great job for this purpose.

$\lambda = 7$



$\lambda = 0.2$



Problem 4 (5 pts, MATLAB)**.** *Inpainting Face Images with Eigenfaces.*
Let $\hat{U} \in \mathbb{R}^{n\times k}$ be the matrix whose columns represent the top-k eigenfaces, and let $y \in \mathbb{R}^n$ denote the mean face image. Recall that if $x \in \mathbb{R}^n$ is a face image the squared distance to face space is given by

$$d_{FS}^2(x) = \|(x-y) - \hat{U}\hat{U}^*(x-y)\|_2^2 = \|(I - \hat{U}\hat{U}^*)(x-y)\|_2^2.$$

This can be used as a regularization function for least squares problems involving face images, i.e., if $x \in \mathbb{R}^n$ is a face image and $b \in \mathbb{R}^m$ are measurements, and $A \in \mathbb{R}^{m\times n}$ is a given measurement matrix, we can consider the RLS problem

$$\min_{x\in\mathbb{R}^n} \|Ax - b\|_2^2 + \lambda\|L(x-y)\|_2^2$$

where $L = I - \hat{U}\hat{U}^*$.

In this problem we will consider the task of "inpainting" face images that have missing pixels. In this case A is a diagonal matrix with 1's and 0's along the diagonal, where a 1 indicates the pixel is present, and 0 indicates the pixel is absent. See the script `problem4.m` for an example, where a rectangle of pixels over the eyes is missing. The goal is to fill in these missing pixels to make a plausible looking face.

Tasks:

(a) Derive a formula for the solution x_{RLS} to the RLS problem above for a general A matrix. Include this in your report.

(b) Implement your formula for x_{RLS} in MATLAB in the script `problem4.m` to solve the inpainting problem. (Note: A, I, $\hat{U}$, b, and y are defined for you in the script). Include the code snippet you wrote in your report (should just be a few lines).

(c) Use your implementation to inpaint the missing pixels of the provided face image. Choose the value of $\lambda > 0$ that gives the most plausible looking face image while not deviating too much from the observed pixels. Include the resulting inpainted image in your report.

Solution 4. .

a-) Since we have $\min_{x\in\mathbb{R}^n} \|Ax - b\|_2^2 + \lambda\|L(x - y)\|_2^2$, we want to make this look like $\min_{\hat{x}\in\mathbb{R}^n} \|A\hat{x} - \hat{b}\|_2^2 + \lambda\|L\hat{x}\|_2^2$, where $\hat{x} = x - y$. Because we know what the solution for that is. Then:

$$\|A(x - y + y) - b\|_2^2 + \lambda\|L(x - y)\|_2^2 \Rightarrow$$

$$\|Ax - Ay + Ay - b\|_2^2 + \lambda\|L(x - y)\|_2^2 \Rightarrow$$

$$\|Ax - Ay - (b - Ay)\|_2^2 + \lambda\|L(x - y)\|_2^2 \Rightarrow$$

$$\|A(x - y) - (b - Ay)\|_2^2 + \lambda\|L(x - y)\|_2^2$$

Which is similar to $\min_{\hat{x}\in\mathbb{R}^n} \|A\hat{x} - \hat{b}\|_2^2 + \lambda\|L\hat{x}\|_2^2$, with $\hat{b} = b - Ay$.

Therefore, solution is:

$$\hat{x} = (A^*A + \lambda L^*L)^{-1}A^*\hat{b} \Rightarrow$$

$$(x - y) = (A^*A + \lambda L^*L)^{-1}A^*(b - Ay) \Rightarrow$$

$$x = [(A^*A + \lambda L^*L)^{-1}A^*(b - Ay)] + y$$

b-)

```
L = (I-Uhat*Uhat');
lambda = 0.1;
x_rls = ((A'*A)+(lambda*(L*L')))\(A'*b-(A*y))+y;
```

c-) I chose $\lambda = 0.1$, it seemed to give the best result while inpainting the eyes. Bigger values for λ were giving best results for the eyes, but they were causing some distortion on the rest of the face. Follow the images:

Problem 5 (5 pts)**.** Recall that $x \in \mathbb{R}^n$ solves the RLS problem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \lambda \|Lx\|_2^2$$

if and only if it satisfies the modified normal equations: $(A^*A + \lambda L^*L)x = A^*b$.

Prove that the modified normal equations have a unique solution (and hence so does RLS) if and only if $\mathrm{null}(A) \cap \mathrm{null}(L) = \{0\}$.

Solution 5. The RLS problem is $\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \lambda \|Lx\|_2^2$. Now let y be the unique minimizer for the equation. Then

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \lambda \|Lx\|_2^2 = \|Ay - b\|_2^2 + \lambda \|Ly\|_2^2$$

Now, if $null(A) \cap null(L) = \{0\}$, and we let a vector v not equal 0, such that $v \neq 0 \in null(A) \cap null(L)$, this implies

$$v \in null(A) \text{ and } v \in null(L) \Rightarrow Av = 0, Lv = 0$$

Where if it exists such vector, then there is another solution to this minimization problem, which would be $y + v$, since

$$\|A(y+v) - b\|_2^2 + \lambda \|L(y+v)\|_2^2 = \|Ay + Av - b\|_2^2 + \lambda \|Ly + Lv\|_2^2 \Rightarrow$$

$$\|Ay - b\|_2^2 + \lambda \|Ly\|_2^2 = \min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \lambda \|Lx\|_2^2, \text{ because} Av = 0, Lv = 0$$

Which is a contradiction, wince we are claiming that the solution is unique.

Therefore, we must have $null(A) \cap null(L) = \{0\}$.

In order to satisfy the modified normal equations $(A^*A + \lambda L^*L)x = A^*b$, we let $U = A^*A + \lambda L^*L$, and we claim that $Ux = 0$ if and only if $Ax = 0$ and $Lx = 0$.

Then, if we pick $x \in null(U) \Rightarrow Ux = 0$ and use our modified normal equations, we have

$$(A^*A + \lambda L^*L)x = (0)x \Rightarrow x^*(A^*A + \lambda L^*L)x = x^*x(0) \Rightarrow$$

$$\|Ax\|_2^2 + \lambda \|Lx\|_2^2 = 0$$

Which implies that both $\|Ax\|_2^2$ and $\|Lx\|_2^2$ are equal to 0, since the 2-norm will always be a positive or 0.

Therefore, $x \in null(A)$ and $x \in null(L)$, which means that

$$null(A^*A + \lambda L^*L) \subseteq null(A) \cap null(L) \text{ and } null(A) \cap null(L) \subseteq null(A^*A + \lambda L^*L)$$

Hence, $null(A) \cap null(L) = null(A^*A + \lambda L^*L)$, where $null(A) \cap null(L) = \{0\}$, which we just showed before. Then $null(A^*A + \lambda L^*L) = 0$.

Now going back to our modified normal equations

$$(A^*A + \lambda L^*L)x = A^*b, \text{ where } A^*A \in \mathbb{R}^{n \times n}, L^*L \in \mathbb{R}^{n \times n}$$

Finally, by using the rank-nullity theorem $(A^*A + \lambda L^*L)$ must be full rank, which with the the fact that they are square matrices, means invertible. Therefore, we have the solution:

$$x = (A^*A + \lambda L^*L)^{-1} A^*b$$