Exercise A: Least square problems

A1

Minimising $\|Ax-b\|_2$ with respect to x means finding x such that the derivative of $\|Ax-b\|_2$ with respect to x is equal to zero. Equivalently, we can minimise the square of the norm which can be developed as follows:

$$\begin{aligned}\|Ax-b\|_2^2 &= \|Ax\|_2^2 - 2 < Ax, b > + \|b\|_2^2 \\ &= x^T A^T Ax - 2x^T A^T b + b^T b\end{aligned}$$

The derivative with respect to x is:

$$\begin{aligned}\frac{\partial \|Ax-b\|_2^2}{\partial x} &= \frac{\partial (x^T A^T Ax - 2x^T A^T b + b^T b)}{\partial x} \\ &= \frac{\partial (x^T A^T Ax)}{\partial x} - \frac{\partial (2x^T A^T b)}{\partial x} + \frac{\partial (b^T b)}{\partial x} \\ &= 2A^T Ax - 2A^T b + 0\end{aligned}$$

This derivative has to be equal to zero. Thus:

$$\begin{aligned}\frac{\partial \|Ax-b\|_2^2}{\partial x} &= 0 \\ \Leftrightarrow 2A^T Ax - 2A^T b &= 0 \\ \Leftrightarrow 2A^T Ax &= 2A^T b \\ \Leftrightarrow A^T Ax &= A^T b\end{aligned}$$

It is indeed a minima and not a maxima as we consider $\|Ax-b\|_2^2$: second order havin only one extrema that is a minima. To demonstrate that when A has full column rank, i.e. rank(A)=n, the solution is unique, we first prove that Ker($A^T A$)=Ker(A):

$$\forall x \in Ker(A) : Ax = 0 \Leftrightarrow A^T Ax = 0 \Leftrightarrow x \in Ker(A^T A) \Rightarrow Ker(A) \subset Ker(A^T A)$$

$$\begin{aligned}\forall x \in Ker(A^T A) : A^T Ax = 0 \Leftrightarrow x^T A^T Ax = 0 \Leftrightarrow \|Ax\|_2 = 0 \Leftrightarrow Ax = 0 \Leftrightarrow x \in Ker(A) \\ \Rightarrow Ker(A^T A) \subset Ker(A)\end{aligned}$$

According to the rank-nullity Theorem:

$$\begin{aligned}rank(A) &= n - dim(Ker(A)) \\ yet : rank(A) &= n \\ \Rightarrow n &= n - dim(Ker(A)) \\ \Leftrightarrow dim(Ker(A)) &= 0 = dim(Ker(A^T A)) \\ \Rightarrow Ker(A^T A) &= \{0\}\end{aligned}$$

As $Ker(A^T A) = \{0\}$ and $A^T A$ is square, we deduce $A^T A$ is invertible and it follows from Theorem 2.1 of the course notes that the solution of the system is unique.

A2

Suppose the QR decomposition of A is given by $Q\binom{R}{0}$, where $Q \in \mathbb{R}^{m\times m}$ is unitary and $R \in \mathbb{R}^{n\times n}$ is upper triangular. We will express the solution of

$$A^T Ax = A^T b \tag{1}$$

in terms of the QR decomposition of A.
Let $R_f = \binom{R}{0}$. We can rewrite equation (1) as:

$$(QR_f)^T QR_f x = (QR_f)^T b$$
$$R_f^T Q^T QR_f x = R_f^T Q^T b$$

As Q is unitary and hence $Q^T Q = I$, we have:

$$\begin{pmatrix} R^T & 0 \end{pmatrix} \begin{pmatrix} R \\ 0 \end{pmatrix} x = \begin{pmatrix} R^T & 0 \end{pmatrix} Q^T b$$

If we call $\hat{Q}$ the matrix consisting of the first n columns of Q, equation (1) becomes:

$$R^T R x = R^T \hat{Q}^T b$$

From theorem 2.8 of the course notes, we know that every matrix $A \in \mathbb{C}^{m\times n}$ of full column-rank admits a factorization $A = Q_1 R_1$ where $Q_1 \in \mathbb{C}^{m\times n}$ is an isometry and $R_1 \in \mathbb{C}^{n\times n}$ is an upper triangular matrix with positive diagonal. The matrix Q_1 corresponds to $\hat{Q}$ and R_1 simply to R. Hence we deduce that R is invertible. This allows us to premultiply both sides of the equation by the inverse of R^T:

$$R^{-T} R^T R x = R^{-T} R^T \hat{Q}^T b$$
$$R x = \hat{Q}^T b$$

The solution of equation (1) is therefore $x = R^{-1}\hat{Q}^T b$ and the computation of the solution is reduced to the resolution of a single triangular system of linear equations (which can be solved efficiently using backward substitution).

Exercise B: Low-rank approximation

B1

For every matrix $A \in \mathbb{R}^{m\times n}$, there exist unitary transformations $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ such that

$$A = U\Sigma V^*, \quad \text{where} \quad \Sigma = \left[\begin{array}{ccc|c} \sigma_1 & & 0 & \\ & \ddots & & 0_{r\times(n-r)} \\ 0 & & \sigma_r & \\ \hline & 0_{(m-r)\times r} & & 0_{(m-r)\times(n-r)} \end{array}\right],$$

with real positive singular values $\sigma_1 \geqslant \cdots \geqslant \sigma_r > 0$. These singular values are unique. Indeed, they can be derived from the eigenvalues of the hermitian matrix A^*A (as explained on page 39 of the course notes) which are uniquely defined.

Next, we show that the rank of a matrix is equal to its number of nonzero singular values.

Proof. We first prove that if A is a $K\times L$ matrix and B is a full-rank $L\times L$ matrix then $\text{rank}(AB) = \text{rank}(A)$.
Let S_1 be the space generated by the columns of A and S_2 the space generated by the columns of AB.
If $x \in S_1$ then there exists $y \in \mathbb{R}^{L\times 1}$ such that $x = Ay$. Considering $\hat{y} = B^{-1}y$ we note that $x = ABB^{-1}y = (AB)\hat{y}$ and so $x \in S_2$. If $x \in S_2$ then there exists $y \in \mathbb{R}^{L\times 1}$ such that $x = (AB)y$, we note that $x = A(By)$ and so $x \in S_2$. This shows that the spaces generated by the columns of A and AB coincide and hence $\text{rank}(A) = \text{rank}(AB)$.
Similarly we prove that if A is a $K\times L$ matrix and B is a full-rank $K\times K$ matrix then $\text{rank}(BA) = \text{rank}(A)$.
Let S_1 be the space generated by the rows of A and S_2 the space generated by the rows of BA.
If $x \in S_1$ then there exists $y \in \mathbb{R}^{1\times K}$ such that $x = yA$. Considering $\hat{y} = yB^{-1}$ we note that $x = yB^{-1}BA = \hat{y}(BA)$ and so $x \in S_2$. If $x \in S_2$ then there exists $y \in \mathbb{R}^{1\times K}$ such that $x = y(BA)$, we note $x = (yB)A$ and so $x \in S_1$. This shows that the spaces generated by the rows of A and AB coincide and hence $\text{rank}(A) = \text{rank}(BA)$.
We know that the rank of a diagonal matrix is equal to the number of its nonzero entries. We also note that in the decomposition $A = U\Sigma V^*$, U and V^* are of full-rank. Therefore, $\text{rank}(A) = \text{rank}(\Sigma) = r$ where r is the number of nonzero singular values. □

B2

Let $x \in \mathbb{R}^{m\times n}$ be such that $|X_{ij}| \leqslant \varepsilon$ for all $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$. Let $\|X\|_2$ be the 2-norm of X and let $\|X\|_F$ be its Frobenius norm. We show that $\|X\|_2 \leqslant \|X\|_F \leqslant \sqrt{mn}\varepsilon$.

Proof. First, we show the first inequality. We know from the lecture notes that

$$\begin{aligned}\|X\|_2 &= \sigma_{\max},\\ \|X\|_F &= \left[\sum_i \sigma_i\right]^{1/2},\end{aligned}$$

where σ_i are the singular values of X. From this, it is immediately clear that $\|X\|_2 \leqslant \|X\|_F$.

Next, we use an equivalent form of the Frobenius norm to show the second inequality:

$$\|X\|_F = \left[\sum_{i,j} |X_{ij}|^2\right]^{1/2}.$$

Knowing that $|X_{ij}| \leqslant \varepsilon$, it is immediate that $\|X\|_F \leqslant \left[\sum_{i,j} \varepsilon^2\right]^{1/2} = \left[mn\varepsilon^2\right]^{1/2} = \sqrt{mn}\varepsilon$. This concludes the proof. □

We also give an example where these bounds are tight. Indeed, consider the matrix $X = \varepsilon \in \mathbb{R}^{1\times 1}$. Clearly, we have $|X_{ij}| \leqslant \varepsilon$ for all i, j (only one value is possible for each). We know that the only singular value of this matrix is ε, and hence

$$\|X\|_2 = \|X\|_F = \sqrt{1 \cdot 1}\varepsilon = 1\varepsilon.$$

B3

We start by observing that if $B = A + X$, then by Theorem 3.28 in the lecture notes, we can write

$$\begin{aligned}\sigma_{\min(m,n)-j+1}(B) &= \min_{\mathcal{S}_j} \max_{x\in\mathcal{S}_j\setminus\{0\}} \frac{\|B\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}\\ &= \min_{\mathcal{S}_j} \max_{x\in\mathcal{S}_j\setminus\{0\}} \frac{\|(A+X)\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}\\ &\leqslant \min_{\mathcal{S}_j} \max_{x\in\mathcal{S}_j\setminus\{0\}} \left(\frac{\|A\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} + \frac{\|X\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}\right)\\ &\leqslant \sigma_{\min(m,n)-j+1}(A) + \sigma_{\min(m,n)-j+1}(X).\end{aligned}$$

However, we know that A has rank r, and hence by the result of B1, we find that if $\min(m,n) - j + 1 > r$, the singular value of A in the expression is zero, and hence that

$$\begin{aligned}\sigma_{\min(m,n)-j+1}(B) &\leqslant \sigma_{\min(m,n)-j+1}(X)\\ &\leqslant \sqrt{mn}\varepsilon,\end{aligned}$$

where the last inequality is a consequence of B2.

A criterion that can then be used to estimate the rank r of A is then to take the smallest r such that $\sigma_{r+1}(B) \leqslant \sqrt{mn}\varepsilon$. This is very similar to the description given on p. 60 of the lecture notes, concerning the numerical rank of the matrix A.

Exercise C: Realization theory

In this last exercise, we are interested in finding an AR model corresponding to the data obtained during the covid pandemic. Indeed, we want to find the parameters α_i of:

$$y(t) = \alpha_0 + \sum_{i=1}^{p} \alpha_i y(t-i)$$

Let's rewrite the problem as a system of linear equations:

$$\begin{pmatrix} 1 & y(p-1) & y(p-2) & \dots & y(0) \\ 1 & y(p) & y(p-1) & \dots & y(1) \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & y(N-1) & y(N-2) & \dots & y(N-p) \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_p \end{pmatrix} = \begin{pmatrix} y(p) \\ \vdots \\ y(N) \end{pmatrix} \tag{2}$$

Note that we consider $\hat{y}(0) = y(0) \dots \hat{y}(p-1) = y(p-1)$ as initial conditions.
We want to solve the system (2) to obtain estimation of the parameters α_i. We call A the matrix and we observe that it usually has more lines than columns. Therefore, it is of interest to solve the system using the least squares method. To do so, we can use what we obtained in question A2 to solve the problem with a QR decomposition and backward substitution. Indeed, we must simply solve the triangular system :

$$Rx = \hat{Q}^T b$$

where $\hat{Q}$ and R are derived from the QR decomposition of A.
We consider two cases : the confinement mode and the social distancing mode. Moreover, for each cases, we divide the data set into a training set (70% of the data) and a validation set (30% of the data).
Considering one mode, the system (2) has to be solved using the data of the training set. Next, an estimated y that we call $\hat{y}$ can be reconstructed using the AR model equation. The first p components of $\hat{y}$ are given by the initial conditions and then the next components are constructed based on the previous calculated estimates:

$$\hat{y}(t) = \alpha_0 + \sum_{i=1}^{p} \alpha_i \hat{y}(t-i)$$

The error between the true y and the estimated $\hat{y}$ can be computed on both the training and validation set. This gives an idea of the estimation quality.

Next we will show the results we obtained while varying the parameter p.

Confinement mode First we will study the confinement mode.

As you can see in Figure 1, when the parameter p increases, the training error has a tendency to decrease. Indeed, the AR model that we obtain fits better and better our training set data.

Then in Figure 2, we observe that the validation error decreases (approximatively) until p = 19. And from then on, the error rises again. Through this, we can understand that around p = 20, the model "overfits" our training set and so it gives a good approximation for the training set but gives a poor performance for the validation set.

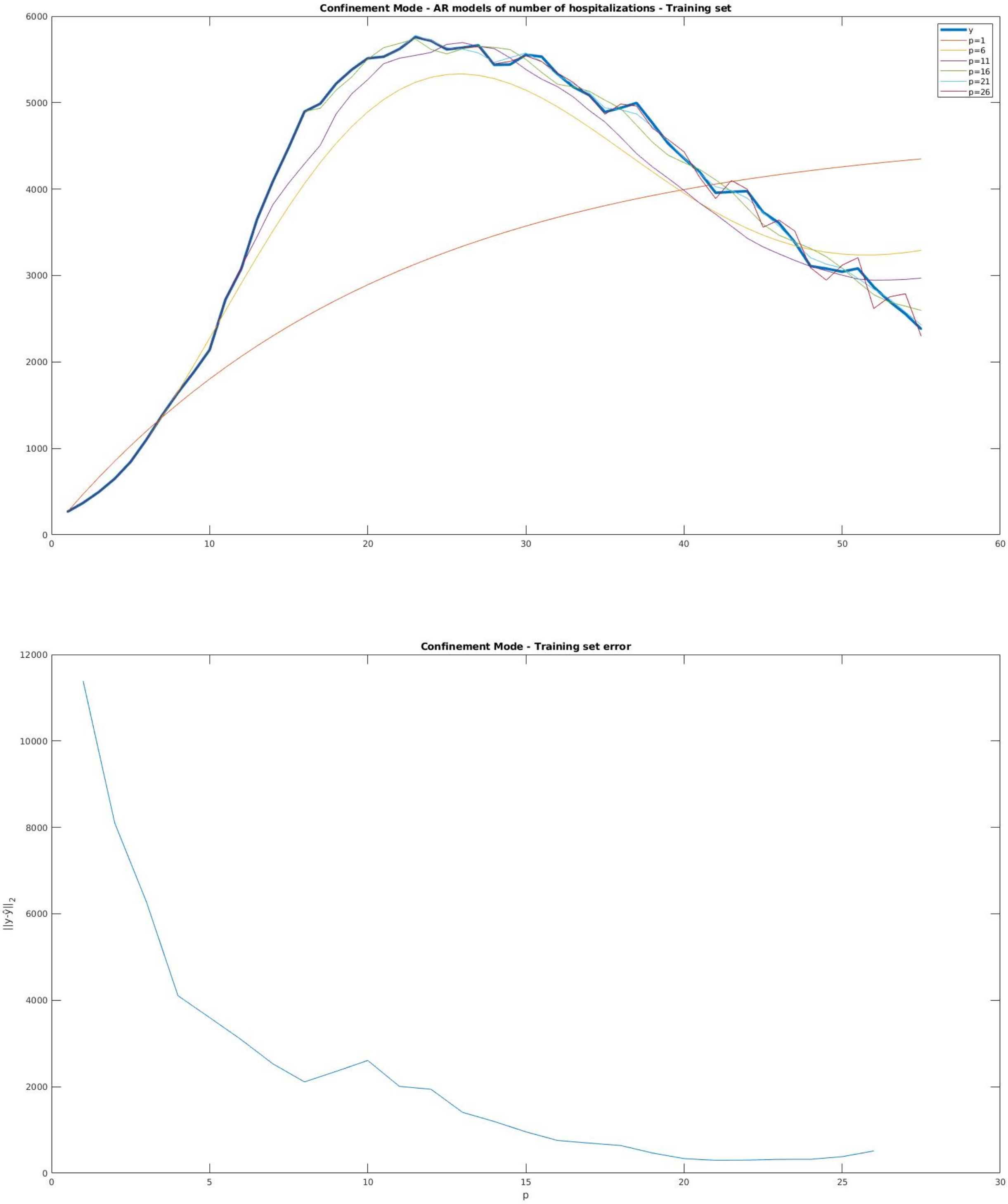
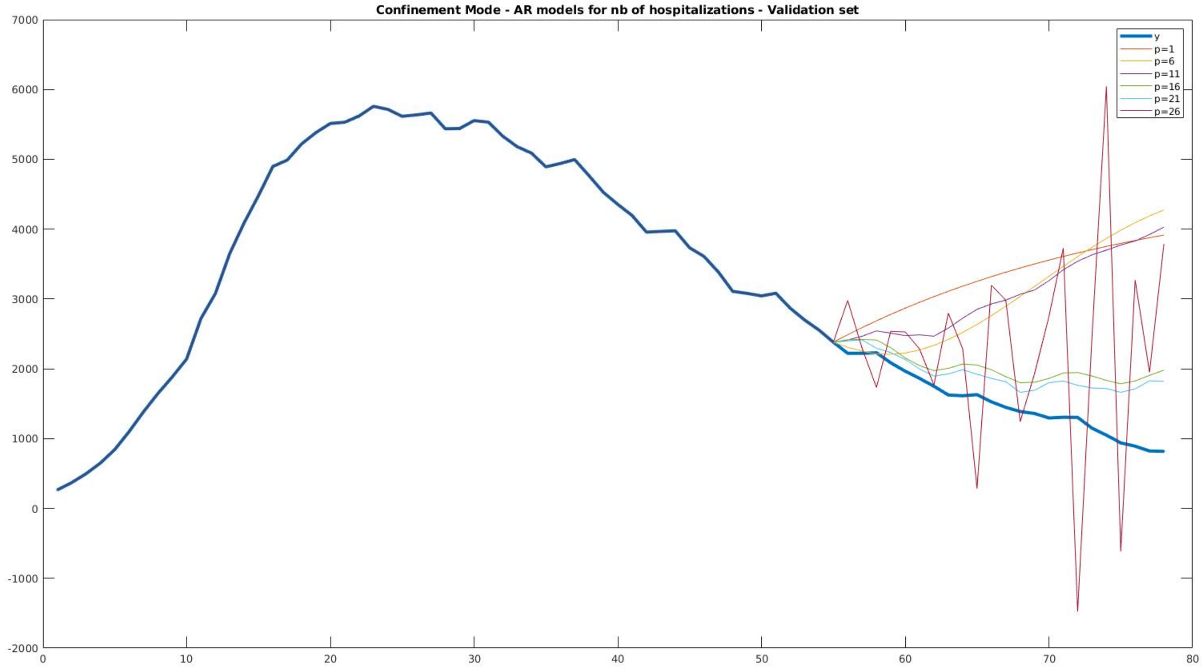


Figure 1: Confinement Mode - Training set - Fitting to AR Models



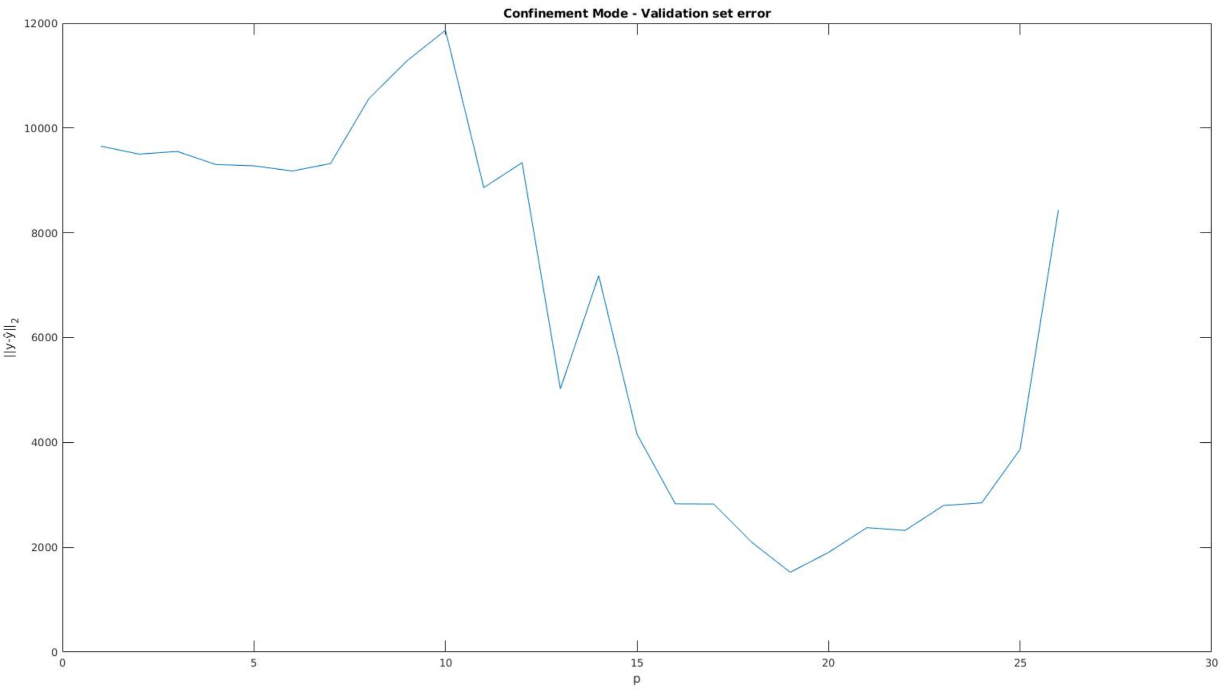


Figure 2: Confinement Mode - Validation set

Social Distancing mode In the social distancing mode, we can make the same observations (in Figures 3 and 4) as in the previous mode : the error in the training set decreases when p increases except at some peaks. For the validation set, we once again observe overfitting when p is higher than 19.

Discussion

In both modes, we have confirmed our intuition that the error in the training set would decrease. However, it is not monotonic and we still observe some peaks. One way to try to explain it is that when we solve the least squares problem, we minimize the error between the real values $y(t)$ and $\tilde{y}(t) = \alpha_0 + \sum_{i=1}^{p} \alpha_i y(t-i)$: $e_1 = \|y - \tilde{y}\|_2$. But when we calculate the error of the training set, we consider $\hat{y}(t) = \alpha_0 + \sum_{i=1}^{p} \alpha_i \hat{y}(t-i)$:

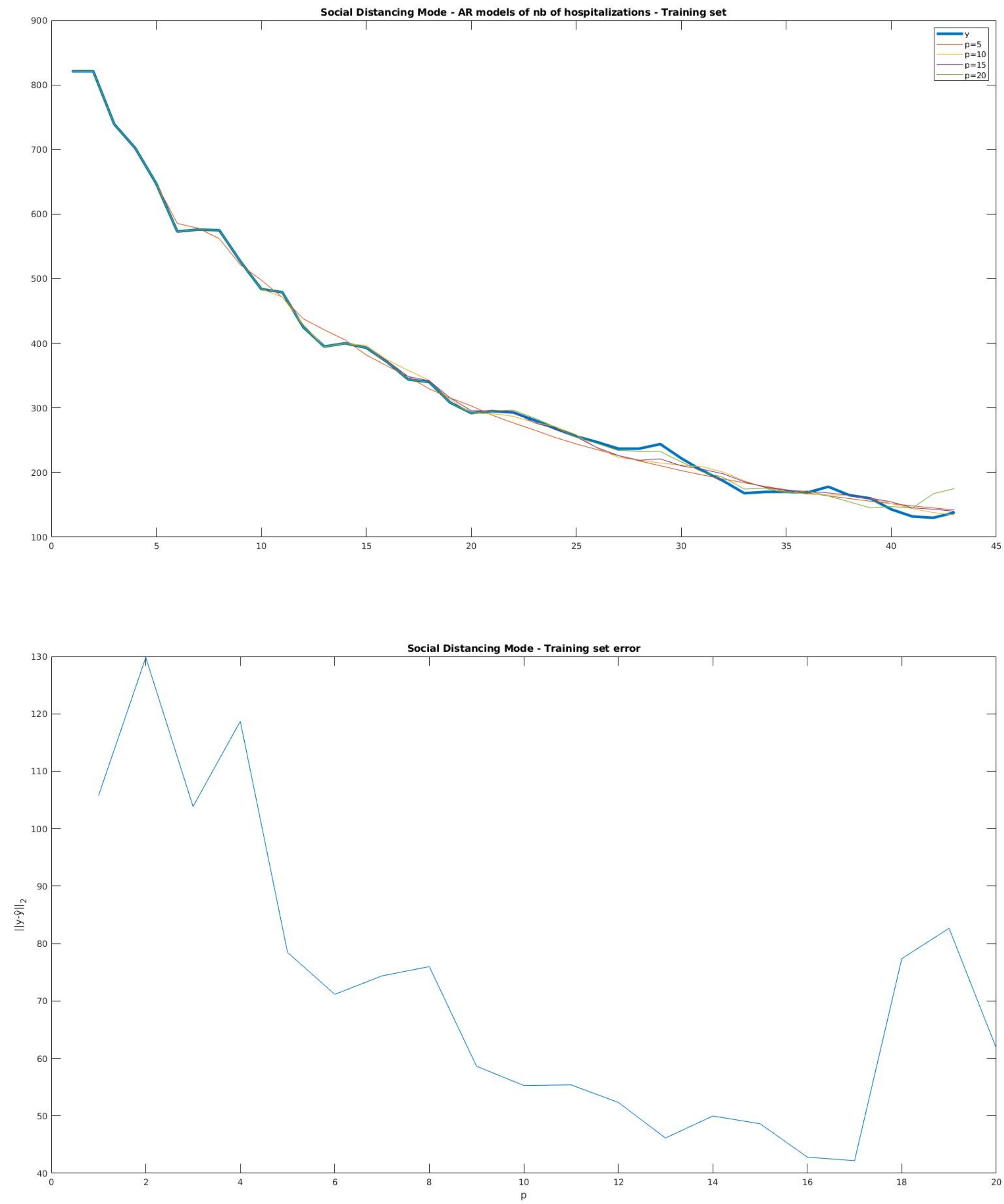
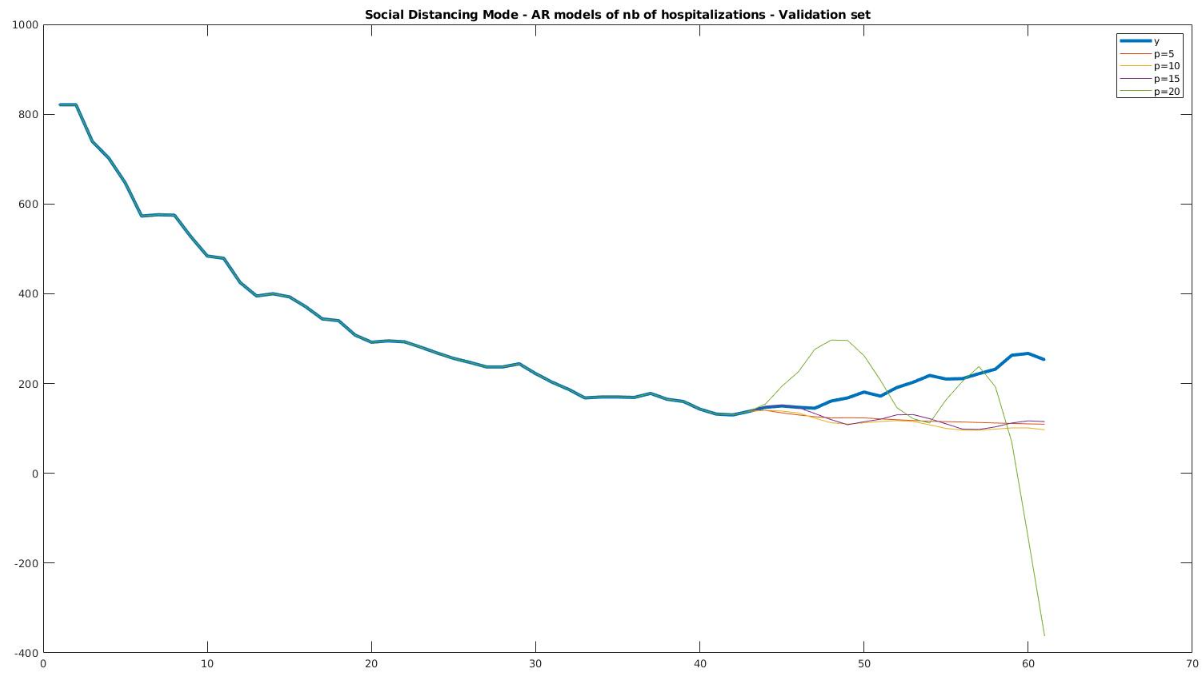


Figure 3: Social Distancing Mode - Training set - Fitting to AR Models

$e_2 = \|y - \hat{y}\|_2$, and so we propagate the errors along the time t. And clearly this last error e_2 is bigger than the first one e_1. So our intuition that the error should decrease monotonically can only apply to the error e_1 (which we have indeed observed in Figure 5) and not to e_2.

Then, about the validation set errors, we can conclude that overfitting a training set doesn't yield good results for future predictions.

Note that in the analysis above, we only considered values of p such that the linear system of equations has more rows than columns. In the case that we have more columns than rows, our system is underdetermined and so solving the system, so we get an infinity number of solutions that are exact solutions (if there are no linear combinations in the rows of course) which means that our training error is null in theory (in practice we have errors of order 10^{-4}, for confinement mode, and 10^{-9}, for social distancing mode, due to numerical



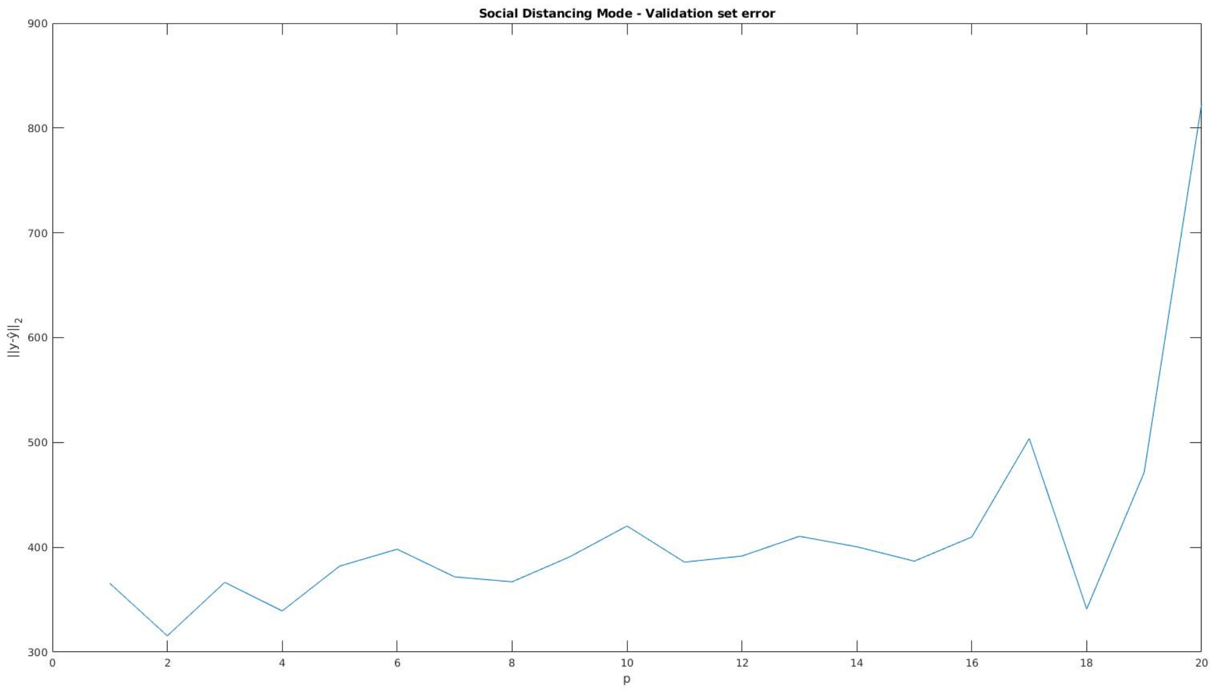


Figure 4: Social Distancing Mode - Validation set

errors). And so, we still have an overfitting that is visible in the validation set errors as they are very big for both modes.

Bonus question

If we approximate the model in real time with data that we receive at fixed interval, we will have to solve a new least squares problem (as defined in the beginning of this section) everytime we want to include new data to improve the system. Indeed, we will have to add new rows each time we receive new data and solve it again. This can be expensive especially if the system is big.

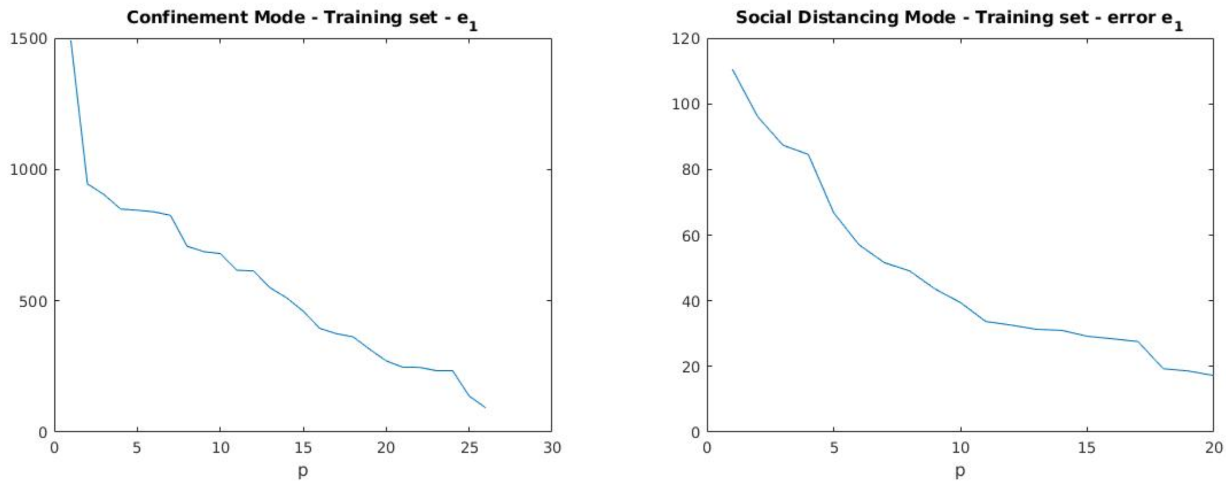


Figure 5: The error e_1 monotonically decreases.