Notes on ARIMA

Jonas Pichat

10 November 2019

1 ARMA

Given a centred time series $y = y_1, \ldots, y_n$ of finite rank, ARIMA(p, d=0, q) equation is:

$$y_{t+1} = \mu(=0) + \epsilon_{t+1} + \Sigma_{i=1}^{q} \theta_i \epsilon_{t-i+1} + \Sigma_{i=1}^{p} \alpha_i y_{t-i+1} \tag{1}$$

where $\epsilon^{(h)}$ is the series of residuals referring to the h-th target[1], y_{t+h} (only 1 step ahead is used here, so the superscript was dropped for clarity).

The first part (in ϵ) is the MA(q) model and the last sum (in y) corresponds to the AR(p) model. As a side note, one could verify that the residuals are not correlated and iid variables sampled from a normal distribution with 0 mean.

It is also worth mentioning that we restricted ourselves to a (prediction) gap, g=0 here and used a non-differentiated series[2] (d=0) without any additional (exogenous) data; in such case, given k exogenous series $x^{(k)}$ (y still being our endogenous, unique target series) and a non-null gap, eq. (1) just becomes:

$$\begin{aligned} y_{t+g+1} = \epsilon_{t+g+1} + f(\epsilon_t, \ldots, \epsilon_{t-q+1}, y_t, \ldots, y_{t-p+1}, x_t^{(1)}, \ldots, x_{t-p+1}^{(1)}, \ldots, \\ x_t^{(k)}, \ldots, x_{t-p+1}^{(k)}, \theta, \alpha) \end{aligned} \tag{2}$$

where f is a polynomial function of degree 1 with more variables and coefficient sets θ and α (referring to MA and AR respectively).

1.1 Difference with the classical ARIMA

When $q > 0$, we do not compute the series of residuals **exactly**, like in the classical MA model (i.e. via backward recursion). The reason is that in order to compute a residual, one needs a target (so $q + h$ observations)...but this is not

[1] here, "target" is to be taken in the scope of a prediction horizon: we do consider a unique "target time series", y but one may be interested in predicting multiple (h) steps ahead: $y_{t+1}, y_{t+2}, \ldots, y_{t+h}$, or more generally $y_{t+g+1}, \ldots, y_{t+g+h+1}, \forall g \geq 0, h > 0$.

[2] when using ARIMA(p, d>0, q), y and x are replaced with their differentiated versions in (2). One could also encounter the lag operator $\mathscr{L}$ in the literature, which, when applied to y simply gives $\mathscr{L}^i y_t = y_{t-i}$ (it just avoids carrying all the $t - i + 1$ indices along); besides, note that $(1 - \mathscr{L}^1) y_t = y_t - y_{t-1}$. For $d > 0$, $g = 0$, (1) can therefore be rewritten as: $\big(1 - \sum_{i=1}^{p} \alpha_i \mathscr{L}^i\big)(1 - \mathscr{L}^1)^d y_{t+1} = \big(1 - \sum_{i=1}^{q} \theta_i \mathscr{L}^i\big)\epsilon_{t+1}$; see Section 2.1 for more details.

always possible (e.g. at the end of the series). So instead, we rather estimate a residual series $\hat{\epsilon}$, at once, from the set of all q consecutive observations available via a linear predictor, and use it in eq. (1). The ARMA equation thus becomes:

$$y_{t+1} - \hat{\epsilon}_{t+1} = \underbrace{\Sigma_{i=1}^{q}\theta_i\hat{\epsilon}_{t-i+1} + \Sigma_{i=1}^{p}\alpha_i y_{t-i+1}}_{\widetilde{y}_{t+1}}$$

Therefore, we seek α and θ that minimise:

$$\text{argmin}_{\alpha,\Theta}\Sigma_{t>\max(p,2q)}\Big[(y_{t+1} - \hat{\epsilon}_{t+1}) - \widetilde{y}_{t+1}\Big]^2$$

1.2 ARIMA(p, 0, q)

In general (though here, exogenous features were excluded for clarity), this means that the "feature-target" matrix, X (with h=2 targets) fed to the final linear model (within the `fit` method of the class `ARIMA`) has the form:

$$\left[\begin{array}{ccc|cccc|cccc|cc}
 & & & & & & & & & & & & \\
 & & & & & & & & & & & y_1 - \hat{\epsilon}_1^{(2)} & \\
 & & & & & & \hat{\epsilon}_1^{(1)} & & & & \hat{\epsilon}_1^{(2)} & y_2 - \hat{\epsilon}_2^{(2)} & \\
 & & y_1 & & & & \hat{\epsilon}_2^{(1)} & & & \hat{\epsilon}_2^{(2)} & & y_3 - \hat{\epsilon}_3^{(2)} & \\
 & y_1 & y_2 & & & \hat{\epsilon}_1^{(1)} & \hat{\epsilon}_3^{(1)} & & & \hat{\epsilon}_3^{(2)} & & y_4 - \hat{\epsilon}_4^{(2)} & \\
y_1 & y_2 & y_3 & & \hat{\epsilon}_1^{(1)} & \hat{\epsilon}_2^{(1)} & \hat{\epsilon}_4^{(1)} & & \hat{\epsilon}_1^{(2)} & \hat{\epsilon}_4^{(2)} & & y_5 - \hat{\epsilon}_5^{(2)} & \\
y_2 & y_3 & y_4 & \hat{\epsilon}_1^{(1)} & \hat{\epsilon}_2^{(1)} & \hat{\epsilon}_3^{(1)} & \hat{\epsilon}_5^{(1)} & \hat{\epsilon}_1^{(2)} & \hat{\epsilon}_2^{(2)} & \hat{\epsilon}_5^{(2)} & & y_6 - \hat{\epsilon}_6^{(2)} & \\
y_3 & y_4 & y_5 & \hat{\epsilon}_2^{(1)} & \hat{\epsilon}_3^{(1)} & \hat{\epsilon}_4^{(1)} & \hat{\epsilon}_6^{(1)} & \hat{\epsilon}_2^{(2)} & \hat{\epsilon}_3^{(2)} & \hat{\epsilon}_6^{(2)} & & y_7 - \hat{\epsilon}_7^{(2)} & \\
y_4 & y_5 & y_6 & \hat{\epsilon}_3^{(1)} & \hat{\epsilon}_4^{(1)} & \hat{\epsilon}_5^{(1)} & \hat{\epsilon}_7^{(1)} & \hat{\epsilon}_3^{(2)} & \hat{\epsilon}_4^{(2)} & \hat{\epsilon}_7^{(2)} & & y_8 - \hat{\epsilon}_8^{(2)} & \\
y_5 & y_6 & y_7 & \hat{\epsilon}_4^{(1)} & \hat{\epsilon}_5^{(1)} & \hat{\epsilon}_6^{(1)} & \hat{\epsilon}_8^{(1)} & \hat{\epsilon}_4^{(2)} & \hat{\epsilon}_5^{(2)} & \hat{\epsilon}_8^{(2)} & & y_9 - \hat{\epsilon}_9^{(2)} & \\
y_6 & y_7 & y_8 & \hat{\epsilon}_5^{(1)} & \hat{\epsilon}_6^{(1)} & \hat{\epsilon}_7^{(1)} & & \hat{\epsilon}_5^{(2)} & \hat{\epsilon}_6^{(2)} & \hat{\epsilon}_9^{(2)} & & y_{10} - \hat{\epsilon}_{10}^{(2)} & \\
 & \vdots & & & & \vdots & & & \vdots & & & \vdots & \\
y_{n-(p+h-1)} & \cdots & y_{n-h} & \hat{\epsilon}_{n-(q+h-1)}^{(1)} & & \cdots & \hat{\epsilon}_{n-h}^{(1)} & \hat{\epsilon}_{n-(q+h-1)}^{(2)} & & \cdots & \hat{\epsilon}_{n-h}^{(2)} & y_{n-h+1} - \hat{\epsilon}_{n-h+1}^{(1)} & y_n - \hat{\epsilon}_n^{(2)}
\end{array}\right]$$

where the first $\max(p, 2q) + g + (h-1)$ (recall g=0 here) rows should be excluded from training, and $\Theta = [\theta_1, \theta_2, \theta_3]^T$ and $\alpha = [\alpha_1^{(1)}, \ldots, \alpha_4^{(1)}, \alpha_1^{(2)}, \ldots, \alpha_4^{(2)}]^T$ are the parameters to be optimised.

The first column block constitutes the AR(3) features, the second, the MA(4) features (residuals)[3] relative to the first target (see superscript "1"), the third, the MA(4) features relative the second target and the last block is the targets.

Note that one should see this final linear model (when $q > 0$) as a "corrected predictor", where the error has been predicted and is hereby accounted for.

Finally, including exogenous data comes down to adding k column blocks similar to the leftmost one but with $x^{(k)}$ series (and the h residual series, $\epsilon^{(h)}$ would now be estimated using both endogenous and exogenous data[4]).

For the sake of completeness, we detail X for AR(p=3) (and h=2 targets) and MA(q=4) (and h=1 target) in the following.

[3] recall that $\hat{\epsilon}_{t+1} = f(y_t, \ldots, y_{t-q+1}, \beta)$ where f is a polynomial function of degree 1 which coefficients are optimised using `ResidualPredictor`. Hence, residuals are only defined for $t > q$ (and conversely, $\hat{\epsilon}_1, \ldots, \hat{\epsilon}_q$ are not defined).

[4] $\hat{\epsilon}_{t+h} = f(y_t, \ldots, y_{t-q+1}, x_t^{(1)}, \ldots, x_{t-q+1}^{(1)}, x_t^{(k)}, \ldots, x_{t-q+1}^{(k)}, \beta)$

1.3 ARIMA(3, 0, 0)

Considering h=2 targets, X has the form:

$$\left[\begin{array}{ccc|cc} . & . & . & . & y_1 \\ . & . & . & y_1 & y_2 \\ . & . & y_1 & y_2 & y_3 \\ . & y_1 & y_2 & y_3 & y_4 \\ y_1 & y_2 & y_3 & y_4 & y_5 \\ y_2 & y_3 & y_4 & y_5 & y_6 \\ & \vdots & & \vdots & \\ y_{n-(p+h-1)} & \cdots & y_{n-h} & y_{n-1} & y_n \end{array}\right]$$

where the first p rows should be excluded from training.

1.4 ARIMA(0, 0, 4)

Considering h=1 target, X has the form:

$$\left[\begin{array}{cccc|c} . & . & . & . & y_1 - \hat{\epsilon}_1 \\ . & . & . & \hat{\epsilon}_1 & y_2 - \hat{\epsilon}_2 \\ . & . & \hat{\epsilon}_1 & \hat{\epsilon}_2 & y_3 - \hat{\epsilon}_3 \\ . & \hat{\epsilon}_1 & \hat{\epsilon}_2 & \hat{\epsilon}_3 & y_4 - \hat{\epsilon}_4 \\ \hat{\epsilon}_1 & \hat{\epsilon}_2 & \hat{\epsilon}_3 & \hat{\epsilon}_4 & y_5 - \hat{\epsilon}_5 \\ \hat{\epsilon}_2 & \hat{\epsilon}_3 & \hat{\epsilon}_4 & \hat{\epsilon}_5 & y_6 - \hat{\epsilon}_6 \\ \hat{\epsilon}_3 & \hat{\epsilon}_4 & \hat{\epsilon}_5 & \hat{\epsilon}_6 & y_7 - \hat{\epsilon}_7 \\ \hat{\epsilon}_4 & \hat{\epsilon}_5 & \hat{\epsilon}_6 & \hat{\epsilon}_7 & y_8 - \hat{\epsilon}_8 \\ \hat{\epsilon}_5 & \hat{\epsilon}_6 & \hat{\epsilon}_7 & \hat{\epsilon}_8 & y_9 - \hat{\epsilon}_9 \\ & & \vdots & & \vdots \\ \hat{\epsilon}_{n-q} & & \cdots & \hat{\epsilon}_{n-1} & y_n - \hat{\epsilon}_n \end{array}\right]$$

where first $2q$ rows should be excluded from training.

1.5 Prediction function

The information needed from the model for the prediction function consists of: the parameters of the model p, d, q and the weights and biases W_1, W_2, b_1, b_2[5] such that:

[5] W_1 and b_1 come from the residual predictor that maps nq observations (n is the number of series under consideration: exogenous + (1) endogenous) with h residual(s). Note that Eq. 3 holds for $d > 0$ except all series are differentiated first—and the prediction must then be back-transformed (see Section 2.1).

$$\begin{aligned}\widetilde{y}_{t+g+h} &= f(y_t, x_t^{(1)}, \dots, x_t^{(k)}; W, b) \\ &= W_2\Big[\mathscr{W}_p\big(y_t; x_t^{(1)}; \dots; x_t^{(k)}\big); \mathscr{W}_q\Big(\underbrace{W_1\big[\mathscr{W}_q\big(y_t; x_t^{(1)}; \dots; x_t^{(k)}\big)\big] + b_1}_{\epsilon_t^{(1)}, \dots, \epsilon_t^{(h)}}\Big)\Big] + b_2\end{aligned} \tag{3}$$

where f is linear, $\mathscr{W}_L$ is the windowing operator of size L, y is the target (endogenous) series and $x^{(k)}$ is the k-the exogenous series, $\epsilon^{(h)}$ is the residual vector associated with the h-th target (y_{t+g+h}), and [.] symbolises concatenation. $\widetilde{y}_{t+h}$ is a "corrected" prediction vector (of size h), since we actually predict $y_{t+h} - \epsilon_{t+h}^{(h)}$ here.

The prediction function requires at least $\max(p, 2q)+d$ terms in order to produce "feature-" or "lagged-" residuals: say one uses an n-long non-differentiated (d=0) series, y_n and is interested in forecasting the first time point $n+1$ after the last observation available ($p = 2, q = 3$); if one only provides the 3 last terms of the series (i.e., $\max(p, q)$), the residual predictor can only predict the one-term residual series, (starting) at $n+1$: $\epsilon^{(h)} = \epsilon_{n+1}$ ($h = 1$ here, and in practice, that series is prepended with nans for the sake of dimensions). However, according to eq. (1), one needs residuals from the series $\epsilon^{(1)}$ at $n-2, n-1$ and n in order to predict y_{n+1} (otherwise forecasting is purely autoregressive). Those are respectively given by $\{y_{n-5}, y_{n-4}, y_{n-3}\}$, $\{y_{n-4}, y_{n-3}, y_{n-2}\}$, and $\{y_{n-3}, y_{n-2}, y_{n-1}\}$, i.e. using up to $2q$ previous observations. Now, one can get the actual prediction using those residuals and the latest observations available y_{n-2}, y_{n-1}, y_n.

1.6 Results

Results on passengers series are shown in the following are shown in Figure 1.

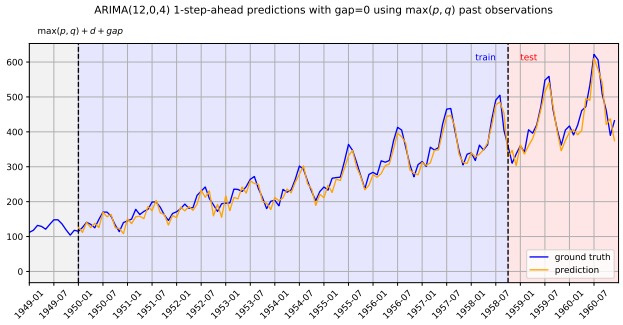
2 ARIMA

2.1 Differencing

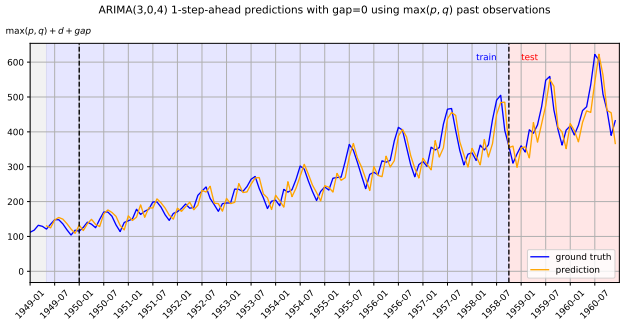
Differencing removes the changes in the level of a time series, eliminating trend and seasonality and consequently stabilising the mean of the time series. It is usually applied when the raw series in non-stationary, as a means to "stationarise" it (order 1, then 2).

The classical definition of ARIMA usually considers $\delta = 1$ (with order d =1 or 2 and rarely above), but in theory, nothing prevents from using larger values e.g., 5-day 7th order differences. As a matter of fact, our ARIMA supports differentiating series using any δ and d, though legitimately, one should then be careful about the nature of the resulting data to be modeled[6].

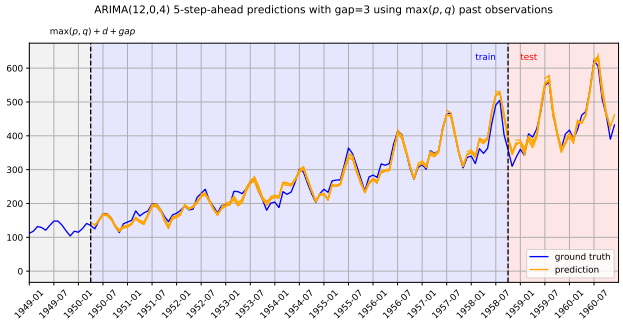
[6] as of now, δ is hard-coded to 1 and d is a free parameter that should not go over 2 (it is difficult to put a physical meaning behind those larger values).



(a)



(b)



(c)

Figure 1: results on passengers series using (1a) ARIMA(12,0,4) and (1b) ARIMA(3,0,4) with g=0 (`prediction_gap`), h=1 (`prediction_horizon`), trained on 75% data (until 1958-01). Both train and test set predictions are shown in orange (ground truth is in blue). First actual prediction is at $\max(p, q) + 1$. One can note that ARIMA(3,0,4) behaves like the persistence model (i.e., uses the previous value as the prediction for the future). (1c) shows ARIMA(12,0,2) for g=3, h=5, i.e. predicting $y_{t+4}, \ldots, y_{t+8}$ at once, starting from $\max(p, q) + g + 1$.

Given $d > 0$ and $\delta > 0$, we introduce the differentiation operator, D in relation to the lag operator from footnote 2, such that $\mathscr{D}_\delta^d = \mathscr{D}_\delta^1 \circ \mathscr{D}_\delta^1 \circ \cdots \circ \mathscr{D}_\delta^1$ (d times) $= (1 - L^\delta)^d$, where d refers to the number of applications of $\mathscr{D}_\delta^1$ and δ is the chosen lag.

For example, one can verify that for $t > d\delta$:

- $\mathscr{D}_\delta^1 y_t = (1 - L^\delta) y_t = y_t - y_{t-\delta}$
- $\mathscr{D}_\delta^2 y_t = \mathscr{D}_\delta^1(\mathscr{D}_\delta^1 y_t)$
  $= (1 - L^\delta)(y_t - y_{t-\delta})$
  $= y_t - 2y_{t-\delta} + y_{t-2\delta}$

From there, one can notice the pattern[7] that leads to a general expression of the "delta series", $\mathscr{D}_\delta^d y$ as a function of d and sequences of $d+1$ terms from the original series y:

$$\mathscr{D}_\delta^d y_t = \sum_{i=0}^{d} (-1)^i \binom{d}{i} y_{t-i\delta} \text{ for } t > d\delta \tag{4}$$

In practice, reversing a delta series only requires (i) the order of differentiation (which gives all the coefficients) and (ii) an interleave sequence, $s = y_1, \ldots, y_{d\delta}$ made of δ subseries $s_k = y_{k+i\delta}$ $(0 < k \leq \delta,\ 0 \leq i < d)$, each of which allows to recursively compute the unknown term left in eq. (4).

For example: given that $\mathscr{D}_3^2 y_t$ is only defined for $t > 6$, the first back-transformed term is $y_7 = \mathscr{D}_3^2 y_7 + 2y_4 - y_1$. This "round of initialisation" holds for $\mathscr{D}_3^2 y_8$ and $\mathscr{D}_3^2 y_9$ here (and in general for the first δ terms), since they use $s_2 = y_2, y_5$ and $s_3 = y_3, y_6$ respectively $(s_1 = y_1, y_4)$. Those 3 series are the δ subseries of the interleave sequence $s = y_1, \ldots, y_6$ defined above. The next terms are obtained in a sliding window fashion.

2.2 General comments

As mentioned previously, our ARIMA supports the use of a prediction gap, g. Because reconstruction of differentiated series relies on the knowledge of directly preceding terms, ARIMA(p, d>0, q) is incompatible with $g > 0$ (or one would have to also predict the target(s)—in the sense of footnote 1—within that gap in order to be able to reverse actual "delta-target(s)" starting at $g + 1$; this however goes against the whole point of choosing a non-null value for g).

The following use cases are therefore recommended for any p, d non both null, and any prediction horizon, $h > 0$:

- $d = 0$, $g > 0$: the model is trained to predict h time points ahead $(y_{t+g+1}, \ldots, y_{t+g+h})$

[7] the coefficients of $\mathscr{D}_\cdot^d$ (in absolute value) are given by the $d+1$-th row of Pascal's triangle.

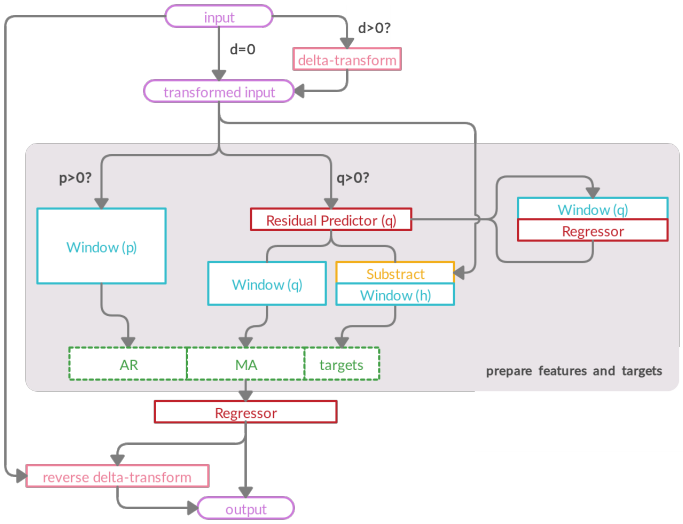


Figure 2: ARIMA flow chart

- $d > 0$, $g = 0$: if one is only interested in predicting y_{t+h} using differencing (i.e., $g = h - 1$), the model must be trained to predict all h time points ahead $(y_{t+1}, \ldots, y_{t+h})$, i.e. forcing $g = 0$, from which y_{t+h} may be cherry picked.

2.3 Summary

A flow chart is presented in Fig. 2.

2.4 Results

Results are presented in Fig. 3.

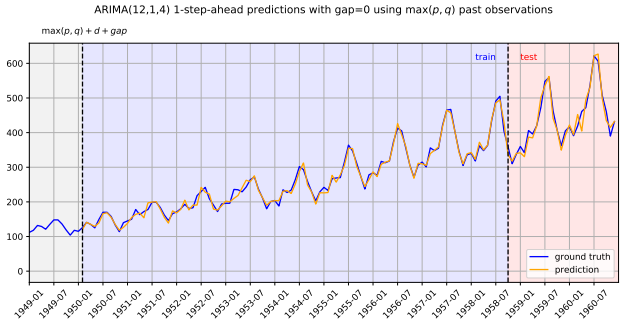
3 Forecasting

3.1 Restrictions

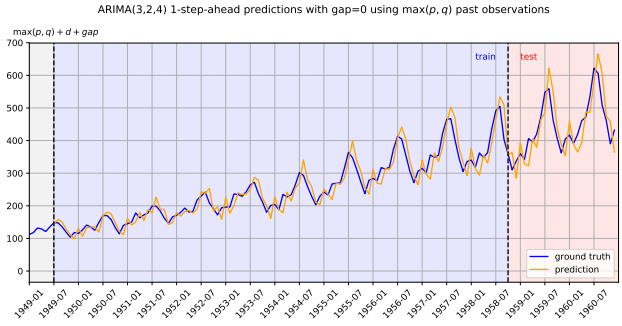
The following use cases[8] are recommended (distinction between prediction horizon, h_p and forecast horizon, h_f is made here[9]):

- $h_p = 1$, $h_f = 1$: the model is trained to predict 1 time point ahead; one can then get a 1 day forecast (h_f) using the latest window of observations (direct way).

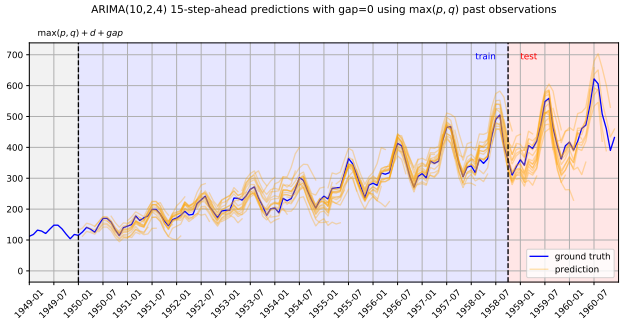
[8] a day is used as an arbitrary time unit for the example.
[9] note that when $h_f = 1$, forecast horizon and prediction horizon are equivalent notions.



(a)



(b)



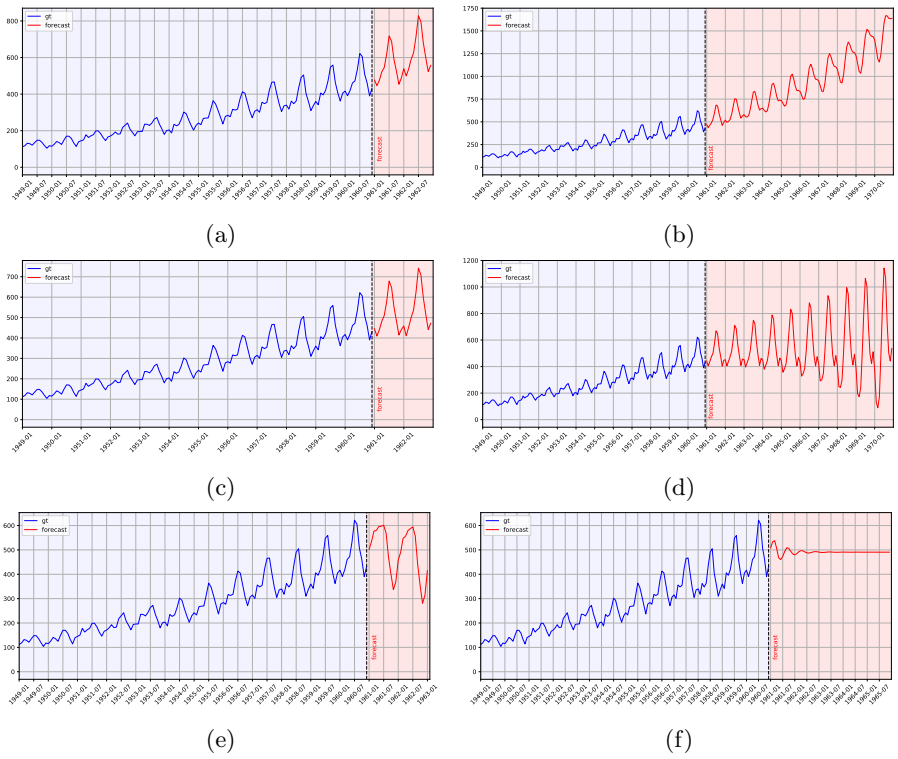
(c)

Figure 3: results on passengers series using (3a) ARIMA(12,1,4) and (3b) ARIMA(3,2,4) with g=0 (`prediction_gap`), h=1 (`prediction_horizon`). (3c) shows ARIMA(10,2,4) for g=0, h=15.

- $h_p = 7$, $h_f = 7$: the model is trained to predict 7 time points at once; one can then get a 7 day forecast using the latest window of observations (direct way).
- $h_p = 1$, $h_f = 7$: the model is trained to predict 1 time point; one can then get a 7 day forecast using previously forecasted values in a sliding fashion (recursive way). Note that the first sliding window is made of the latest observations; it then slides and gets postpended with forecasts. One should thus be aware that errors will accumulate in this setting.
- $h_p = 7$, $h_f = 15$: the model is trained to predict 7 days; one can then get a 15 day-ahead forecast corresponding to forecasting 3 times 1 week, by reusing the forecasted days in a sliding fashion (recursive way). It is similarly to the previous case except 7 days are predicted at each iteration. Hence diagonal averaging is preformed on the "forecast matrix" to recover a 15 day-ahead forecast vector. Errors will accumulate like in the previous case.

3.2 Results

Results are presented in Fig. 4.



(a) (b)

(c) (d)

(e) (f)

Figure 4: results on passengers series using (4a) ARIMA(12,0,5) and (4b) ARIMA(12,0,5) with h_p=24, $h_f = h_p$ (direct way) and h_p=1, h_f=120 (recursive way) respectively (g=0). (4c) and (4d) show ARIMA(12,1,5) and ARIMA(12,2,5) with h_p=24, $h_f = h_p$ and h_p=1, h_f=30 respectively. Finally, (4e) and (4f) show ARIMA(3,1,5) with h_p=25, $h_f = h_p$ and h_p=1, h_f=60 respectively.