ARIMA model

I. AR model description

Model $\mathrm{AR}(p)$ is given by a following expression

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \ldots + \varphi_p y_{t-p} + \varepsilon_t,$$

where:

- y_t – value of TS @ time t;
- C – constant term to be determined;
- φ_t – parameters of the model;
- ε_t – model error term at time t, a.k.a. noise term.

Assumptions and limitations:

1. $\mathbb{E}(y_t) = 0 \ \forall t$;
2. $\mathrm{Var}(y_t) = \mathrm{const} = \sigma^2 \ \forall t$;
3. $\mathrm{Cov}(\varepsilon_t, \varepsilon_s) = 0 \ \forall t \neq s$;
4. $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$

Key features:

1. Interpretability
2. $\mathrm{AR}(p)$ can be applied to stationary TS;

2.1. Influence of previous values fades over time;

3. ACF: autocorrelation function decays over time;
4. PACF: partial autocorrelation function breaks off after lag p, hence it can be used to find optimal p.

Training AR model

1. OLS

$$\sum_t \varepsilon_t^2 = \sum_t \left(y_t - c - \varphi_1 y_{t-1} - \ldots - \varphi_p y_{t-p}\right) \to \min_{c, \varphi_i}$$

2. MLE

Given that $Y = (y_1, \ldots, y_n)$:

$$L(\theta \mid y) \approx \prod_{t=p+1}^{n} f\left(y_t \mid y_{t-1}, ..., y_{t-p}, \theta\right).$$

Considering that $y_t | y_{t-1}, ..., y_{t-p} \sim \mathcal{N}\left(c + \sum_{i=1}^{p} \varphi_i y_{t-i}, \sigma^2\right)$,

$$L(\theta \mid y) = \prod_{i=p+1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(y_t - c - \sum_{i=1}^{p} \varphi_i y_{t-i}\right)^2\right) \to \max_{\theta}.$$

II. MA(q) model description

$$y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + ... + \theta_q \varepsilon_q$$

where:

- θ_i – model parameters;
- ε_i – time series error terms at time i;
- μ – configurable constant term;

Key features:

1. Interpretability;
2. Always stationary;
3. ACF: breaks off after lag q, hence used to determine the optimal q value;
4. PACF: decays gradually;

Training MA model

Let us assume that $\varepsilon_i = 0, \ i = 0, ..., q+1$. Then

1. Conditional LS. Denoting $\theta = \{\theta_1, ..., \theta_q\}$ we get the following:

$$\sum_{t=1}^{n} \varepsilon_t^2 = \sum_{t} \left(y_t - \mu - \sum_{i=1}^{q} \varepsilon_{t-i} \theta_i\right) \to \min_{\mu, \theta}$$

2. MLE. Denoting $Y = (y_1, ..., y_n)$, we get

$$L(\theta \mid y) \approx \prod_{t=p+1}^{n} f\big(y_t \mid \varepsilon_{t-1}, ..., \varepsilon_{t-q}, \theta\big)$$

ARMA(p, q) model description

$$y_t = c + \varphi_1 y_{t-1} + ... + \varphi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + ... + \theta_q \varepsilon_{t-q}$$

here p is defined as the first zero of PACF and q as the first zero of ACF

ARIMA(p, q, d) model desciption

ARIMA is an ARMA model fit to $\Delta^d y_t$:

$$\begin{aligned}
\Delta y_t &= y_t - y_{t-1} = (1 - L) y_t,\ L y_t = y_{t-1} \\
\Delta^2 y_t &= \Delta(y_t - y_{t-1}) = y_t - 2 y_{t-1} + y_{t-2} = (1 - L)^2 y_t \\
\Delta^d y_t &= (1 - L)^d y_t.
\end{aligned}$$

Thus:

$$\begin{aligned}
& y_t = c + \varphi_1 y_{t-1} + ... + \varphi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + ... + \theta_q \varepsilon_{t-q} \\
& (1 - \varphi_1 L - ... - \varphi_p L^p) y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + ... + \theta_q \varepsilon_{t-q} \\
& \left(1 - \sum_{i=1}^{p} \varphi_i L^i\right) \Delta^d y_t = c + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \\
& \left(1 - \sum_{i=1}^{p} \varphi_i L^i\right) (1 - L)^d y_t = c + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}
\end{aligned}$$

ARIMA model usage

Training:

1. $Y = (y_1, ..., y_n)$
2. Find $z_j^{(m)}$ terms (todo: find out how they are called)

$$z_j^{(m)} = \left[y_{j-m+1}, \dots, y_{j-1}, y_j\right] \to Z^m = \begin{pmatrix} z_m^{(m)} \\ z_{m+1}^{(m)} \\ \vdots \\ z_n^{(m)} \end{pmatrix}, Z^m \in \mathbb{R}^{n-m \times m}.$$

Inference:

3. $z_{n+1}^{(m)} = \left[y_{n-m+2}, \dots, y_n, \hat{y}_{n+1}\right]$

   $\tilde{z}_{n+1}^{(m)} = z_{n+1}^{(m)}[: -1] = \left[y_{n-m+2}, \dots, y_n\right]$

4. Collect a set of possible predictions:

$$S_{n+1} = \left\{z_i^{(m)}[m] \mid ||\tilde{z}_i^{(m)} - \tilde{z}_{n+1}|| < \varepsilon\right\}.$$

5. Choose single prediction:

- $\hat{y}_{n+1} = \text{mean}(S_{n+1})$
- $\hat{y}_{n+1} = \text{mode}(S_{n+1})$
- $\text{Cusler}(S_{n+1}) \overset{\text{cluster}}{\to} \{C_1, \dots, C_k\}$

$\hat{y}_{n+1} = \text{mean}(C_i),\ i = \text{argmax}_j |C_j|.$

Multistep ahead prediction: $\tilde{z}_{n+2}^{(m)} = \left[y_{n-m+3}, \dots, y_n, \hat{y}_{n+1}\right].$

2*. $Z^m \overset{\text{cluster}}{\to} C_1, \dots, C_k$ clusters of $z^{(m)}$ vectors

$$X^m = \begin{pmatrix} \xi_1^{(m)} \\ \vdots \\ \xi_k^{(m)} \end{pmatrix},\ \xi_i = \text{"central element"}$$

3*. $\tilde{z}_{n+1}^{(m)} = \left[y_{n-m+2}, \dots, y_n\right]$

4*. $S_n = \left\{\xi_i^{(m)}[m] |\ ||\tilde{\xi}_i^{(m)} - \tilde{z}_{n+1}^{(m)}|| < \varepsilon\right\}$

Let us modify the algorithm:

1. $Y = (y_1, \dots, y_n)$

2. Given K, L : generate patterns (rus. *шаблоны*). For example:

$$K = 10, L = 4: \;\; A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 \\ \vdots & \vdots & \vdots & \vdots \\ 10 & 10 & 10 & 10 \end{pmatrix}$$

$$z_i^{(10,10,10,10)} = [y_{i-40}, ..., y_{i-10}, y_i]$$

Note that in this example A can have only elements with values from 1 to 10.

3. $\forall \alpha \in A : \alpha = (K_1, ..., K_L), K_i \in \overline{1, .., K}$

$$z_i^* = \left[y_{i-K_L-K_{L-1}-...K_1}, ..., y_{i-K_L-K_{L-1}}, y_{i-K_L}, y_i\right]$$

Generate $Z^\alpha = \begin{pmatrix} z^\alpha_{K_{m-K_L}} \\ z^\alpha_{K_{m-K_{L+1}}} \\ \vdots \\ z^\alpha_n \end{pmatrix} \;\; \forall \alpha \in A$

4. $\forall \alpha \in A$:

$$\tilde{z}^\alpha_{n+1} = \left[y_{n+1-K_L-K_{L-1}-...}, ..., y_{n+1-K_L}\right]$$

5. $\forall \alpha \in A$

$$S^\alpha_{n+1} = \left\{z^\alpha_i[i+1] \mid ||\tilde{z}^\alpha_i - \tilde{z}^\alpha_{n+1}||\right\}$$
$$S_{n+1} = \cup_\alpha S^\alpha_{n+1}$$

6. Classify point the point as predictable or not prediactable.

6.1 Cluster $S_{n+1} \to C_1, ..., C_l, C_0$, where C_0 is a noise cluster and $C_i, \; i = \overline{1, .., l}$ are sorted by size from largest (C_1) to smallest (C_l).

$$\eta_1 = \frac{|C_1|}{|C_2|} \gg 1 \to \text{goto } 6.2$$

$$\eta_2 = \frac{|C_1|}{\sum_{i=1}^{l} |C_i|} > \varepsilon_1 \to \text{goto } 6.2$$

6.2 Estimate the variance of cluster C_1. If $\mathrm{Var}(C_1) < \varepsilon_1 \to \text{goto } 7$, if $\mathrm{IQR}(C_1) < \varepsilon_2 \to \text{goto } 7$

6.3 Identify the point as non-predictable and nove to the next point.

7. Obtain single prediction.

example on the photo (what do we do if a value is skipped)

Self-healing algorithms

1. Forecast h steps ahead.
2. Run self-healing unill convergence or max iteration is reached.
3. Move to step 1.