Derivations for ARMA node

Wouter Kouw

January 13, 2021

1 Autoregressive Moving Average

Let y_k be an observation, generated by the following function:

$$y_k = \theta_1 y_{k-1} + \dots \theta_M y_{k-M} + \theta_{M+1} e_{k-1} + \dots \theta_{M+N} e_{k-N} + e_k \tag{1}$$

where θ_i are parameters and the error $e_k \sim \mathcal{N}(0, \tau^{-1})$ is a white noise term. M denotes the number of previous observations and N the number of previous error terms considered. I define the following vectors:

$$x_{k-1} = \begin{bmatrix} y_{k-1} \\ y_{k-2} \\ \vdots \\ y_{k-M} \end{bmatrix} , \quad r_{k-1} = \begin{bmatrix} e_{k-1} \\ e_{k-2} \\ \vdots \\ e_{k-N} \end{bmatrix} \tag{2}$$

These vectors can be interpreted as "state buffers"; they hold a number of previous observations and previous errors "in memory". In transitioning from k to k+11, the elements shift down. The final one drops out and the most recent observation is added as the first element.

1.1 EM-like procedure for residuals

Each error e_k is, in fact, the difference between the actual observation and the deterministic part of Equation 1:

$$e_{k-1} = y_{k-1} - \Big(\theta_1 y_{k-1} + \dots \theta_{M+N} e_{k-N}\Big) \tag{3a}$$

$$\vdots \tag{3b}$$

$$e_3 = y_3 - \Big(\theta_1 y_2 + \theta_2 y_1 + \theta_{M+1} e_2 + \theta_{M+2} e_1\Big) \tag{3c}$$

$$e_2 = y_2 - \Big(\theta_1 y_1 + \theta_{M+1} e_1\Big) \tag{3d}$$

$$e_1 = y_1 \, . \tag{3e}$$

In statistics, these are known as residuals[1]. One can see that the errors are computed recursively: we start with an initial error e_1, which does not depend on previous errors. The next error e_2 depends only on e_1, for which we have already computed a value. In frequentist statistics, we would have a point estimate for the coefficients θ (i.e., we would say $\hat{\theta} = [0.4 \dots 0.2]$ or something) and would therefore also get a point estimate for e_2. However, in Bayesian statistics, we have a full distribution θ, not a point estimate. That means we would also get a full distribution for e_2. For example, if θ were Gaussian distributed, we would also get a Gaussian distribution for e_2 (see rules for linear transformations and combinations of Gaussian random variables). But that now poses a problem for the computation of e_3: now, both θ and e_2 are Gaussian random variables and the product of two Gaussian r.v.'s does not produce another Gaussian distributed r.v.

One very simple way of getting around this is to collapse the posterior θ into a point estimate, e.g. picking the mode or expected value. Note that this is purely for the purposes of computing the residuals; anywhere else we would still use the full distribution for θ. If we fix θ to a particular point estimate, then we get point estimates for the e terms. This is exactly what happens in an Expectation-Maximization procedure. For example, in EM for Gaussian mixture models[2], the assignment variable is computed as a point estimate (i.e., to what component belongs each data point; Expectation step) and the Gaussian components are computed as full distributions (i.e. means and variance are updated; Maximization step).

As a side effect, the residuals are now promoted to the status of "observed variables": at each k, we have access to previous residuals $e_{k-1}, \dots$ for which we have already computed point values, much like the vector of previous observations. To provide a more concise mathematical description later on, we can define another vector $z_{k-1} = \big[y_{k-1}, \dots, y_{k-M}, e_{k-1}, \dots, e_{k-N}\big]^\top$, which joins the two buffer vectors x_{k-1} and r_{k-1}.

1.2 Generative model

Given the shorthand definitions, we can simplify Equation 1 to

$$y_k = \theta^\top z_{k-1} + e_k \, , \tag{4}$$

where $\theta = \big[\theta_1, \dots, \theta_{M+N}\big]^\top$. Integrating out e_k yields:

$$y_k \sim \mathcal{N}(\theta^\top z_{k-1}, \tau^{-1}) \; . \tag{5}$$

This equation is the likelihood. We have two unknown variables: the coefficients θ and precision τ. The coefficients can be both positive and negative, consist of real-valued numbers and can (or even *should*) correlate with each other. It

[1] `https://en.wikipedia.org/wiki/Errors_and_residuals`

[2] `https://en.wikipedia.org/wiki/Expectation-maximization_algorithm`

therefore makes sense to define a multivariate Gaussian prior for them. The precision parameter τ is a strictly positive parameter and it makes most sense to define a univariate Gamma prior for it. More formally:

$$p(\theta) = \mathcal{N}(\theta \mid m_\theta^0, V_\theta^0) \tag{6a}$$

$$p(\tau) = \Gamma(\tau \mid a_\tau^0, b_\tau^0)\,, \tag{6b}$$

where the superscript 0 marks the parameters as belonging to the initial priors.

1.3 Recognition model

The recognition model will match the priors:

$$q(\theta) = \mathcal{N}(\theta \mid m_\theta, V_\theta) \tag{7a}$$

$$q(\tau) = \Gamma(\tau \mid a_\tau, b_\tau)\,, \tag{7b}$$

this time without the superscript 0.

2 Message computations

We define a composite ARMA node with interfaces described by Figure 1.

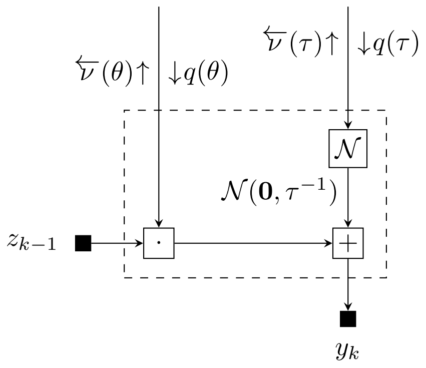
$\overleftarrow{\nu}(\tau)\uparrow$
$\downarrow q(\tau)$
$\overleftarrow{\nu}(\theta)\uparrow$
$\downarrow q(\theta)$
$\mathcal{N}$
$\mathcal{N}(\mathbf{0}, \tau^{-1})$
z_{k-1}
$\cdot$
$+$
y_k

Figure 1: Schematic of ARMA node internal operations.

To incorporate this into ForneyLab, we need to derive the custom messages sent by the node, $\overleftarrow{\nu}(\theta)$ and $\overleftarrow{\nu}(\tau)$. The general formula for computing a variational message is:

$$\nu(x_i) \propto \exp\Big(\mathbb{E}_{q(x_{j \neq i})}\Big[\log f(x_i, x_{j_1}, \dots)\Big]\Big)\,, \tag{8}$$

where f is the factor node [1]. In words, the message passed by a factor node to a particular connected edge x_i combines the beliefs for all other connected edges x_j.

2.1 Message to θ

$$\begin{aligned}
\log \overleftarrow{\nu}(\theta) &\propto \mathbb{E}_{q(\tau)}\Big[\log \mathcal{N}\Big(y_k \mid \theta^\top z_{k-1}, \tau^{-1}\Big)\Big] && \text{(9a)} \\
&\propto -\frac{1}{2}\mathbb{E}_{q(\tau)}\Big[\tau(y_k - \theta^\top z_{k-1})^2\Big] && \text{(9b)} \\
&\propto -\frac{1}{2}\Big[\frac{a_\tau}{b_\tau}(y_k^2 - 2y_k\theta^\top z_{k-1} + (\theta^\top z_{k-1})^2)\Big] && \text{(9c)} \\
&\propto -\frac{1}{2}\frac{a_\tau}{b_\tau}\Big[-2\theta^\top \underbrace{z_{k-1}y_k}_{\phi} + \theta^\top \underbrace{z_{k-1}z_{k-1}^\top}_{\Phi}\theta\Big]\,. && \text{(9d)}
\end{aligned}$$

We recognize both a linear function, $\theta^\top\phi$, and a quadratic function, $\theta^\top\Phi\theta$, in the log-domain. If we distribute $\Phi\Phi^{-1}$ (equivalent to multiplying by $3/3 = 1$) across all terms, we get:

$$-\frac{1}{2}\frac{a_\tau}{b_\tau}\Big[-2\theta^\top\Phi\Phi^{-1}\phi + \theta^\top\Phi\Phi^{-1}\Phi\theta\Big]\,. \tag{10}$$

Consider for a moment a multivariate Gaussian $\mathcal{N}(x \mid \mu, \sigma^2)$ and strip away the normalization terms as well as all terms in the exponent that don't depend on x, i.e.:

$$\exp\Big(-\frac{1}{2}\Big[-2x^\top\mu\frac{1}{\sigma^2} + x^\top\frac{1}{\sigma^2}x\Big]\Big)\,. \tag{11}$$

With this in the back of our mind, we can recognize a Gaussian distribution in the exponentiated form of Equation 10: the term $\Phi^{-1}\phi$ corresponds to the mean μ and the term Φ corresponds to the inverse variance $\frac{1}{\sigma^2}$. Hence, we can say that:

$$\overleftarrow{\nu}(\theta) \propto \mathcal{N}(\Phi^{-1}\phi, \Phi^{-1})\,. \tag{12}$$

2.2 Message to τ

$$\begin{aligned}
\log \overleftarrow{\nu}(\tau) &\propto \mathbb{E}_{q(\theta)}\Big[\log \mathcal{N}\Big(y_k \mid \theta^\top z_{k-1}, \tau^{-1}\Big)\Big] && \text{(13a)} \\
&\propto \frac{1}{2}\log\tau - \tau\frac{1}{2}\mathbb{E}_{q(\theta)}\Big[(y_k - \theta^\top z_{k-1})^2\Big] && \text{(13b)} \\
&\propto \frac{1}{2}\log\tau - \tau\frac{1}{2}\mathbb{E}_{q(\theta)}\Big[y_k^2 - 2y_k\theta^\top z_{k-1} + \theta^\top z_{k-1}z_{k-1}\theta\Big] && \text{(13c)} \\
&\propto \frac{1}{2}\log\tau - \tau\frac{1}{2}\Big(y_k^2 - 2y_k m_\theta^\top z_{k-1} + \mathbb{E}_{q(\theta)}\Big[z_{k-1}^\top\theta\theta^\top z_{k-1}\Big]\Big) && \text{(13d)} \\
&\propto \frac{1}{2}\log\tau - \tau\frac{1}{2}\Big(y_k^2 - 2y_k m_\theta^\top z_{k-1} + z_{k-1}^\top(m_\theta m_\theta^\top + V_\theta)z_{k-1}\Big)\,. && \text{(13e)}
\end{aligned}$$

where I have made use of the fact that you are allowed to flip inner products, $\theta^\top z = z^\top\theta$, and that the expectation of an outer product of Gaussian r.v.'s is the outer product of the means plus the covariance matrix [2].

Consider for a moment, a Gamma distribution $\Gamma(x \mid a, b)$ without the normalizing terms and the terms that don't depend on x:

$$x^{a-1} \exp(-bx) \tag{14}$$

Note that this is the shape-rate parameterization of the Gamma[3]. Let's take a look at the exponentiated form of Equation 13e:

$$\tau^{1/2} \exp\Big(- \tau \frac{1}{2} \big(y_k^2 - 2 y_k m_\theta^\top z_{k-1} + z_{k-1}^\top \big(m_\theta m_\theta^\top + V_\theta \big) z_{k-1} \big) \Big) \,. \tag{15}$$

We can recognize the following Gamma distribution in that:

$$\overleftarrow{\nu}(\tau) \propto \Gamma(\tau \mid \alpha, \beta) \,, \tag{16}$$

with

$$\alpha = \frac{3}{2} \,, \quad \beta = \frac{1}{2} \big(y_k^2 - 2 y_k m_\theta^\top z_{k-1} + z_{k-1}^\top \big(m_\theta m_\theta^\top + V_\theta \big) z_{k-1} \big) \,. \tag{17}$$

References

[1] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[2] KB Petersen, MS Pedersen, et al. The matrix cookbook, vol. 7. *Technical University of Denmark*, 15, 2008.

[3] `https://en.wikipedia.org/wiki/Gamma_distribution`