

## Math Modeling, Week 4

**1. Compare the Kalman filter to simple RL (with no cue). Look at their updating rules and explain how they relate. Extra challenge: building on this connection, try to derive a Bayesian version of Rescorla-Wagner (hint – assume the weights follow Gaussian random walks).**

The Kalman filter maintains a full probability distribution rather than a point estimate. Nevertheless, its best point estimate is the mean of that distribution,  $\mu$ . The update rule for  $\mu$  can be written as

$$\mu_{n+1} = \frac{\sigma_\epsilon^2}{\sigma_\epsilon^2 + s_n^2} \mu_n + \frac{s_n^2}{\sigma_\epsilon^2 + s_n^2} y_n.$$

Note that the two fractions here add to 1. Thus the new estimate ( $\mu_{n+1}$ ) is a weighted average of the old estimate ( $\mu_n$ ) and the current observation ( $y_n$ ). A little rearranging yields

$$\mu_{n+1} = \mu_n + \frac{s_n^2}{\sigma_\epsilon^2 + s_n^2} (y_n - \mu_n).$$

This equation maps directly onto simple RL, where  $\mu$  is the model's prediction ( $P$  or  $V$ ),  $y$  is the outcome ( $R$ ), and  $s_n^2 / (s_n^2 + \sigma_\epsilon^2)$  is the learning rate ( $\epsilon$ ). The difference is that Kalman tracks uncertainty and uses it to adapt its learning rate. The learning rate can be interpreted as the proportion of the total uncertainty that comes from the previous estimate ( $s_n^2$ ) versus the current observation ( $\sigma_\epsilon^2$ ). More of the former leads to greater adjustment toward the current observation, and more of the latter leads to less adjustment.

You might have noticed that the variable Kalman is learning about ( $x$ ) is different from the one it learns from ( $y$ ), whereas for RL these two are same ( $R$ ). However, because the distribution of  $y_n$  is centered on  $x_n$  (i.e., observation noise has zero mean), the best point estimate for  $x_n$  is also the best prediction for  $y_n$ . What this all means is that RL can be given a rational interpretation, wherein the reward on each trial is treated as a noisy sample of some true underlying reward rate, which randomly drifts over time.

### Bayesian Rescorla-Wagner

We need to add two assumptions to Rescorla-Wagner to map in onto the Kalman filter. First, assume the weight vector  $w$  follows a Gaussian random walk,

$$w_{n+1} \sim \mathcal{N}(w_n, \sigma_\eta^2 I_k),$$

where  $k$  is the number of cues and  $I_k$  is the  $k \times k$  identity matrix. Second, assume that the observed outcome on each trial includes noise:

$$R_n \sim \mathcal{N}(w_n \cdot s_n, \sigma_\epsilon^2),$$

where  $s_n$  is the current stimulus (i.e., vector of cue values). Then  $w$  and  $R$  respectively correspond to  $x$  and  $y$  in the Kalman filter. This Kalman filter is more complex than the simple one considered above, because the latent variable ( $w$ ) is now multidimensional, and the observation ( $R$ ) is based not directly on  $w$  but on its inner product with the current stimulus ( $w \cdot s$ ). This makes the update step from prior to posterior on each trial a bit more complicated. Let the prior be given by

$$w_n | \mathbf{s}_{n-1}, \mathbf{R}_{n-1} \sim \mathcal{N}(\mu_n, \Sigma_n),$$

where  $\mathbf{s}_i$  and  $\mathbf{R}_i$  are the vectors of stimuli and outcomes up through trial  $i$ . The posterior can be derived using the formula for multivariate Gaussian density, as follows:

$$\begin{aligned} p(w_n | \mathbf{R}_n, \mathbf{s}_n) &\propto p(w_n | \mathbf{R}_{n-1}, \mathbf{s}_{n-1}) \cdot p(R_n | w_n, s_n) \\ &\propto e^{-\frac{1}{2}(w_n - \mu_n)^T \Sigma_n^{-1} (w_n - \mu_n)} \cdot e^{-\frac{1}{2\sigma_\epsilon^2}(w_n \cdot s_n - R_n)^2} \\ &\propto e^{-\frac{1}{2} \left[ w_n^T \left( \Sigma_n^{-1} + \frac{1}{\sigma_\epsilon^2} s_n s_n^T \right) w_n - 2 \left( \Sigma_n^{-1} \mu_n + \frac{1}{\sigma_\epsilon^2} R_n s_n \right)^T w_n \right]} \\ &\propto e^{-\frac{1}{2} \left( w_n - (\sigma_\epsilon^2 I_k + \Sigma_n s_n s_n^T)^{-1} (\sigma_\epsilon^2 \mu_n + \Sigma_n R_n s_n) \right)^T \left( \Sigma_n^{-1} + \frac{1}{\sigma_\epsilon^2} s_n s_n^T \right) \left( w_n - (\sigma_\epsilon^2 I_k + \Sigma_n s_n s_n^T)^{-1} (\sigma_\epsilon^2 \mu_n + \Sigma_n R_n s_n) \right)}. \end{aligned}$$

This is a Gaussian with mean shown in green (appears twice) and precision (inverse variance) shown in red. The prior on the next trial is obtained from the posterior by adding the variance of the random-walk step and leaving the mean unchanged. Therefore we have the update rules

$$\mu_{n+1} = (\sigma_\epsilon^2 I_k + \Sigma_n s_n s_n^T)^{-1} (\sigma_\epsilon^2 \mu_n + \Sigma_n R_n s_n)$$

and

$$\begin{aligned}\Sigma_{n+1} &= \left( \Sigma_n^{-1} + \frac{1}{\sigma_\epsilon^2} s_n s_n^T \right)^{-1} + \sigma_\eta^2 I_k \\ &= \sigma_\epsilon^2 \Sigma_n (\sigma_\epsilon^2 I_k + \Sigma_n s_n s_n^T)^{-1} + \sigma_\eta^2 I_k.\end{aligned}$$

These are a bit complicated, but you should be able to see how they parallel the update rules for the one-dimensional Kalman filter, generalizing the notion of precision-weighted averaging.

To compare this model to Rescorla-Wagner, it's helpful to write the update rule in terms of  $\Delta\mu$ :

$$\Delta\mu = \mu_{n+1} - \mu_n = (\Sigma_n^{-1} \sigma_\epsilon^2 + s_n s_n^T)^{-1} (R_n - s_n \cdot \mu_n) s_n$$

This is the same as the Rescorla-Wagner rule (for updating the point estimate of  $w$ ), except the learning rate  $\epsilon$  has been replaced with a more complicated matrix term that tracks the uncertainty of each weight and the covariance between weights.

**2. Generate data from a Kalman filter, meaning the sequence of mean predictions across trials, for some interesting sequence of observations. Fit the Kalman and RL models to the data and compute AIC for each model. If you want more, create data from an RL model on the same observation sequence, and then fit Kalman and RL models to these data and compute AICs.**

Done in code.

**3. Prove that  $p(\mathbf{x}_{\text{unobserved}} | \mathbf{x}_{\text{observed}})$  is the stationary distribution for Gibbs sampling.**

To simplify notation, ignore  $\mathbf{x}_{\text{observed}}$  and simply write  $p(\mathbf{x})$ , remembering that some components of this distribution are implicitly fixed. Let  $\mathbf{z}$  represent the sample at any step in the Markov chain, treated as a random variable with distribution matching  $p(\mathbf{x})$ . Define the next sample,  $\mathbf{z}'$ , by drawing  $z'_i$  from  $p(x_i | \mathbf{x}_{-i} = \mathbf{z}_{-i})$  for some  $i$ , with the other variables unchanged ( $\mathbf{z}'_{-i} = \mathbf{z}_{-i}$ ). Let  $\mathbf{y}$  be any possible value for  $\mathbf{x}$ . Then we have:

$$\begin{aligned}\Pr[\mathbf{z}' = \mathbf{y}] &= \Pr[\mathbf{z}'_{-i} = \mathbf{y}_{-i}] \cdot \Pr[z'_i = y_i | \mathbf{z}'_{-i} = \mathbf{y}_{-i}] \\ &= \Pr[\mathbf{z}_{-i} = \mathbf{y}_{-i}] \cdot \Pr[z'_i = y_i | \mathbf{z}_{-i} = \mathbf{y}_{-i}] && \text{(because } \mathbf{z}'_{-i} = \mathbf{z}_{-i} \text{)} \\ &= \Pr[\mathbf{x}_{-i} = \mathbf{y}_{-i}] \cdot \Pr[x_i = y_i | \mathbf{x}_{-i} = \mathbf{y}_{-i}] && \text{(because } \mathbf{z} \text{ has same distribution as } \mathbf{x}; \text{ because } z'_i \sim p(x_i | \mathbf{x}_{-i}) \text{)} \\ &= \Pr[\mathbf{x} = \mathbf{y}]\end{aligned}$$

Therefore  $\mathbf{z}'$  is also distributed according to  $p(\mathbf{x})$ , and hence  $p(\mathbf{x})$  is the stationary distribution for the Gibbs process.