NOTES ON TYLER'S AMGEN PROJECT

ABSTRACT. Here are some notes on the project so far

1. BASIC CONCEPTS

The Kullback-Leibler Divergence can be seen as a measure of the distance between two probability distributions. While it is not a true distance because it is not the same regardless of order, this is a good way to look at the measure. The equation for the KL Divergence is:

$$D_{KL}(p(x)||q(x)) = \int p(x) \log \frac{p(x)}{q(x)} \tag{1.1}$$

Another important quantity is the Fisher Information. The Fisher Information represents a measure of the importance of a given parameter to a probability distribution. For a parameter θ, $P(x|\theta)$ represents the likelihood of a random variable x conditioned on θ. The Fisher Information for this conditional probability distribution is:

$$F_I = \left\langle \left(\frac{\partial}{\partial \theta} \log P(x|\theta) \right)^2 \right\rangle_{P(x|\theta)} \tag{1.2}$$

For many parameters $\theta_1, ..., \theta_n$, a matrix J, known as the Fisher Information Matrix, can be created, such that:

$$J_{ij} = \left\langle \frac{\partial}{\partial \theta_i} \log P(x|\theta_i) \frac{\partial}{\partial \theta_j} logP(x|\theta_j) \right\rangle \tag{1.3}$$

Using an input to a neural network as our functional parameters, we can write $P(x|\mathbf{s})$, where s $= (s_1, ..., sn)$ is the input signal history and s_k is the signal k timesteps in the past. With this, we can write the Fisher Information Matrix as

$$J_{ij} = \left\langle \frac{\partial^2}{\partial s_i \partial s_j} \log P(x|\mathbf{s}) \right\rangle_{P(x|\mathbf{s})} \tag{1.4}$$

The Fisher Information Matrix represents the rate of change of the KL Divergence, as

$$D_{KL} \approx \delta s^T J \delta s \tag{1.5}$$

For Gaussian systems, both the KL Divergence and the FIM can be written in simple forms:

$$D_{KL} = (\mu^1 - \mu^2)^T C^{-1} (\mu^1 - \mu^2) \tag{1.6}$$

$$J_{ij} = \frac{\partial \mu^T}{\partial s_i} C^{-1} \frac{\partial \mu}{\partial s_j} \tag{1.7}$$

If the mean is linearly dependent on the signal, the above approximation becomes exact. This will be the case for all of our work.

2. Memory in Kalman Filters

Kalman Filter Equations:

$$x_t = Vx_{t-1} + \nu, \quad \nu \in \mathcal{N}(0,Q) \tag{2.1}$$

$$y_t = Ux_t + \mu, \quad \mu \in \mathcal{N}(0,R) \tag{2.2}$$

$$\hat{x}_t^- = V\hat{x}_{t-1} \tag{2.3}$$

$$P_t^- = VP_{t-1}V^T + Q \tag{2.4}$$

$$K_t = P_t^- U^T(UP_t^- U^T + R)^{-1} \tag{2.5}$$

$$\hat{x}_t = \hat{x}_t^- + K_t(y_t - U\hat{x}_t^-) \tag{2.6}$$

$$P_t = (I - K_tU)P_t^- \tag{2.7}$$

Iterated equations and their derivations:

x_n:

$$x_n = V^n x_0 + \sum_{k=0}^{n-1} V^k \nu_{n-k} \tag{2.8}$$

$\hat{x}_n^-$:

$$\begin{aligned}
\hat{x}_n^- &= V\hat{x}_{n-1} \\
&= V(\hat{x}_{n-1}^- + K_{n-1}(y_{n-1} - U\hat{x}_{n-1}^-)) \\
&= (V - VK_{n-1}U)\hat{x}_{n-1}^- + VK_{n-1}y_{n-1} \\
&= (V - VK_{n-1}U)(V - VK_{n-2}U)\hat{x}_{n-2}^- + (V - VK_{n-1}U)VK_{n-2}y_{n-2} + VK_{n-1}y_{n-1} \\
&= \left(\prod_{k=1}^{n-1}(V - VK_{n-k}U)\right)Vx_0 + \sum_{i=1}^{n-1}\left(\prod_{k=1}^{i-1}(V - VK_{n-k}U)\right)VK_{n-i}y_{n-i}
\end{aligned}$$

P_n^-:

$$P_n^- = \left(\prod_{k=1}^{n-1} V(I - K_{n-k}U)\right)P_1^- V^{(n-1)T} + \sum_{i=0}^{n-2}\left(\prod_{k=1}^{i} V(I - K_{n-k}U)\right)QV^{iT} \tag{2.9}$$

$\hat{x}_n$:

$$\begin{aligned}
\hat{x}_n &= \hat{x}_n^- + K_n(y_n - U\hat{x}_n^-) \\
&= V\hat{x}_{n-1} + K_ny_n - K_nUV\hat{x}_{n-1} \\
&= (V - K_nUV)\hat{x}_{n-1} + K_ny_n \\
&= (V - K_nUV)(V - K_{n-1}UV)\hat{x}_{n-2} + (V - K_nUV)K_{n-1}y_{n-1} + K_ny_n \\
&= \left(\prod_{k=0}^{n-1}(V - K_{n-k}UV)\right)x_0 + \sum_{i=0}^{n-1}\left(\prod_{k=0}^{i-1}(V - K_{n-k}UV)\right)K_{n-i}y_{n-i}
\end{aligned}$$

P_n:

$$P_n = \left(\prod_{k=0}^{n-2}(I - K_{n-k}U)V\right) P_1 V^{(n-1)T} + \sum_{i=0}^{n-2}\left(\prod_{k=0}^{i}(I - K_{n-k}U)V\right) V^{-1}QV^{iT} \tag{2.10}$$

Expectation and Covariance Calculations:
E($\hat{x}_n^-$):

$$E(\hat{x}_n^-) = \left(\prod_{k=1}^{n-1}(V - VK_{n-k}U)\right) Vx_0 + \sum_{i=1}^{n-1}\left(\prod_{k=1}^{i-1}(V - VK_{n-k}U)\right) VK_{n-i}UV^{n-i}x_0 \tag{2.11}$$

E($\hat{x}_n$):

$$E(\hat{x}_n) = \left(\prod_{k=0}^{n-1}(V - K_{n-k}UV)\right) x_0 + \sum_{i=1}^{n-1}\left(\prod_{k=0}^{i-1}(V - K_{n-k}UV)\right) K_{n-i}UV^{n-i}x_0 + K_nUV^nx_0 \tag{2.12}$$

Cov($\hat{x}_n^-$):

$$\begin{aligned} Cov(\hat{x}_n^-) &= \sum_{i=1}^{n-1}(B_iQB_i^T + A_iRA_i^T) \\ A_i &= \left(\prod_{k=1}^{i-1}(V - VK_{n-k}U)\right) VK_{n-i} \\ B_i &= \sum_{j=1}^{i} A_jUV^{i-j} \end{aligned} \tag{2.13}$$

Cov($\hat{x}_n$)

$$\begin{aligned} Cov(\hat{x}_n) &= \sum_{i=0}^{n-1}(H_iQH_i^T + G_iRG_i^T) \\ G_i &= \left(\prod_{k=0}^{i-1}(V - K_{n-k}UV)\right) K_{n-i} \\ H_i &= \sum_{j=0}^{i} G_jUV^{i-j} \end{aligned} \tag{2.14}$$

A few interesting things to note are that the expectation depends linearly on x_n and that the covariance will increase with increasing n. This second fact will become more important when looking at the FIM.

Fisher Information Matrices:

$$J^-_{rs} = \frac{\partial \mu^{-T}}{\partial y^{n-r}} C^{--1} \frac{\partial \mu^-}{\partial y^{n-s}}$$

$$\frac{\partial \mu^-}{\partial y^{n-r}} = \left(\prod_{k=1}^{r-1} (V - VK_{n-k}U) \right) VK_{n-r} = A_r \tag{2.15}$$

$$J^-_{rs} = A_r^T \left(\sum_{i=1}^{n-1} (B_i Q B_i^T + A_i R A_i^T) \right)^{-1} A_s$$

$$J_{rs} = \frac{\partial \mu^T}{\partial y_{n-r}} C^{-1} \frac{\partial \mu}{\partial y_{n-s}}$$

$$\frac{\partial \mu}{\partial y^{n-r}} = \left(\prod_{k=0}^{r-1} (V - K_{n-k}UV) \right) K_{n-r} = G_r \tag{2.16}$$

$$J_{rs} = G_r^T \left(\sum_{i=0}^{n-1} (H_i Q H_i^T + G_i R G_i^T) \right)^{-1} G_s$$

In the simple 1-D case, all variables $\in \mathbb{R}^{1x1}$, we can compute the decay of J^-_{rr} and J_{rr} as $r \to n$. For J^-_{rr},

$$\begin{aligned} J^-_{rr} &= A_r C^{--1} A_r \\ &= \frac{A_r^2}{C^-} \\ &= \frac{\left(\left(\prod_{k=1}^{r-1} (V - VK_{n-k}U) \right) VK_{n-r} \right)^2}{C^-} \\ &\approx \frac{((V - V\mathbf{K}U)^{r-1}\mathbf{K})^2}{C^-} \\ &\approx \alpha^- \cdot (\lambda^-)^{r-1} \end{aligned} \tag{2.17}$$

where $\mathbf{K}$ is the steady state approximation of K_{n-i}, $\alpha^- = \frac{V^2\mathbf{K}^2}{C^-}$ and $\lambda^- = (V - V\mathbf{K}U)^2$. Similarly, J_{rr} can be expressed as

$$J_{rr} \approx \alpha \cdot (\lambda)^r \tag{2.18}$$

with $\alpha = \frac{\mathbf{K}^2}{C}$ and $\lambda = (V - \mathbf{K}UV)^2$.
Taking the sum of J^-_{rr} from 1 to $n-1$ yields

$$J^-_{tot} = \sum_{r=1}^{n-1} J^-_{rr} = \frac{\alpha^-}{\lambda^-} \frac{\lambda^- - (\lambda^-)^n}{1 - \lambda^-} \tag{2.19}$$

Similarly,

$$J_{tot} = \sum_{r=0}^{n-1} J_{rr} \approx \alpha \frac{1 - \lambda^n}{1 - \lambda} \tag{2.20}$$

3. Desirable and Undesirable Memory

The Kalman filter has built in it two types of memory, which we have termed desirable and undesirable. The desirable memory can be seen as memory of the noise ν in the internal process, and the undesirable memory can be seen as memory of the noise μ in the observations. We propose that the FMC of the desirable memory should decay slowly, and the FMC of the undesirable should decay quickly, perhaps exponentially. I will briefly go over the derivation of the FMC for both types of memory, which will be of the form of equation 1.7.

Firstly, $\hat{x}_n$ should be rewritten.

$$\begin{aligned}
\hat{x}_n &= \left(\prod_{k=0}^{n-1}(V - K_{n-k}UV)\right)x_0 + \sum_{i=0}^{n-1}\left(\prod_{k=0}^{i-1}(V - K_{n-k}UV)\right)K_{n-i}y_{n-i} \\
&= Ax_0 + \sum_{i=0}^{n-1} B_i y_{n-i} \\
&= Ax_0 + \sum_{i=0}^{n-1} B_i(Ux_{n-i} + \mu_{n-i}) \\
&= Ax_0 + \sum_{i=0}^{n-1} B_i U(V^{n-i}x_0 + \sum_{j=0}^{n-i-1} V^j \nu_{n-i-j}) + \sum_{i=0}^{n-1} B_i \mu_{n-i} \\
&= (A + \sum_{i=0}^{n-1} B_i UV^{n-i})x_0 + \sum_{i=0}^{n-1} B_i U \sum_{j=0}^{n-i-1} V^j \nu_{n-i-j} + \sum_{i=0}^{n-1} B_i \mu_{n-i} \\
&= Cx_0 + \sum_{i=0}^{n-1}\sum_{j=0}^{i} B_j UV^{i-j}\nu_{n-i} + \sum_{i=0}^{n-1} B_i \mu_{n-i}
\end{aligned}$$

Written out, this equation yields:

$$\begin{aligned}
\hat{x}_n = {} & \left(\left(\prod_{k=0}^{n-1}(V - K_{n-k}UV)\right) + \sum_{i=0}^{n-1}\left(\prod_{k=0}^{i-1}(V - K_{n-k}UV)\right)K_{n-i}UV^{i-j}\right)x_0 + \\
& \sum_{i=0}^{n-1}\sum_{j=0}^{i}\left(\prod_{k=0}^{j}(V - K_{n-k})\right)K_{n-j}UV^{i-j}\nu_{n-i} + \sum_{i=0}^{n-1}\left(\prod_{k=0}^{i}(V - K_{n-k}UV)\right)K_{n-i}\mu_{n-i}
\end{aligned} \tag{3.1}$$

Using this we can calculate our desired quantities.

It will help to specify the variable B_i as

$$B_i = \left(\prod_{k=0}^{i-1}(V - K_{n-k}UV)\right)K_{n-i}$$

3.1. **Desirable Memory.**

The expected value of $\hat{x}$ is derived as follows:

$$\begin{aligned}E(\hat{x}_n|\nu) &= E\left(Cx_0 + \sum_{i=0}^{n-1}\sum_{j=0}^{i} B_jUV^{i-j}\nu_{n-i} + \sum_{i=0}^{n-1} B_i\mu_{n-i}\right)\\ &= Cx_0 + \sum_{i=0}^{n-1}\sum_{j=0}^{i} B_jUV^{i-j}\nu_{n-i} + E\left(\sum_{i=0}^{n-1} B_i\mu_{n-i}\right)\\ &= Cx_0 + \sum_{i=0}^{n-1}\sum_{j=0}^{i} B_jUV^{i-j}\nu_{n-i}\end{aligned}$$

For the FIM we are interested in the derivative of the expected value with respect an individual time point in the signal's history. Thus,

$$\frac{\partial E(\hat{x}_n|\nu)}{\partial \nu_{n-r}} = \sum_{j=0}^{r} B_jUV^{r-j}$$

The covariance of $\hat{x}_n$ is also derived as follows:

$$\begin{aligned}C(\hat{x}_n|\nu) = Cov(\hat{x}_n|\nu) &= Cov\left(Cx_0 + \sum_{i=0}^{n-1}\sum_{j=0}^{i} B_jUV^{i-j}\nu_{n-i} + \sum_{i=0}^{n-1} B_i\mu_{n-i}\right)\\ &= Cov(Cx_0) + Cov\left(\sum_{i=0}^{n-1}\sum_{j=0}^{i} B_jUV^{i-j}\nu_{n-i}\right) + Cov\left(\sum_{i=0}^{n-1} B_i\mu_{n-i}\right)\\ &= Cov\left(\sum_{i=0}^{n-1} B_i\mu_{n-i}\right)\\ &= \sum_{i=0}^{n-1} Cov(B_i\mu_{n-i})\\ &= \sum_{i=0}^{n-1} B_iRB_i^T\end{aligned}$$

Using the derivative of the expectation and the covariance calculations, we can write

$$J_{rs}(\hat{x}_n|\nu) = \left(\sum_{j=0}^{r} B_jUV^{r-j}\right)^T \left(\sum_{i=0}^{n-1} B_iRB_i^T\right)^{-1} \left(\sum_{j=0}^{s} B_jUV^{s-j}\right) \tag{3.2}$$

3.2. **Undesirable Memory.**

The expected value is calculated similar to above, and yields

$$E(\hat{x}_n|\mu) = Cx_0 + \sum_{i=0}^{n-1} B_i\mu_{n-i}$$

We are again interested in the derivative of this equation with respect to a previous μ term.

$$\frac{\partial E(\hat{x}_n|\mu)}{\partial \mu_{n-r}} = B_r$$

The covariance can be derived in a similar fashion as before and shown to be

$$C(\hat{x}_n|\mu) = \sum_{i=0}^{n-1} \left(\sum_{j=0}^{i} B_j U V^{i-j} \right) Q \left(\sum_{j=0}^{i} B_j U V^{i-j} \right)^T$$

Putting these equations together, we get an expression for the FIM.

$$J_{rs}(\hat{x}_n|\mu) = B_r^T \left(\sum_{i=0}^{n-1} \left(\sum_{j=0}^{i} B_j U V^{i-j} \right) Q \left(\sum_{j=0}^{i} B_j U V^{i-j} \right)^T \right)^{-1} B_r \tag{3.3}$$

4. Connection to Surya's Paper

Surya's update equation was written as $x_t = Wx_{t-1} + vs_t + z_t$. We can make a library translating our variables from the Kalman filter to match Surya's equation. In our case, the update equation is $\hat{x}_t = (V - K_t UV)\hat{x}_{t-1} + K_t U x_t + K_t \mu_t$.

Thus, in our case, the network state is the updated prediction, and the variables can be related as follows:

$x_t \rightarrow \hat{x}_t$
$W \rightarrow (V - K_t UV)$
$v \rightarrow K_t U$
$s_t \rightarrow x_t$
$z_t \rightarrow K_t \mu_t$

A notable difference between the two equations is, however, that W now becomes the time dependent W_t. The same occurs with v, becoming v_t. Since the time dependence arises from the time dependence of the Kalman gain, we can approximate the Kalman gain with the steady-state Kalman gain to simplify the problem and allow for a more direct translation.

Using this simplification,the covariance of $\hat{x}_t$ can be calculated directly from Surya's equation for the covariance of x_t, namely

$$C_n = \epsilon \sum_{k=0}^{\infty} W^k W^{kT}. \tag{4.1}$$

Plugging in for W and substituting $Kcov(\mu_t)K^T$ for the covariance of z_t, we get

$$cov(\hat{x}_n) = \sum_{k=0}^{n-1} (V - KUV)KRK^T(V - KUV)^T. \tag{4.2}$$

Using equation (4) from Surya's supplemental paper, we can now calculate J(k).

(4.3)

$$J(k) = (KU)^T(V-KUV)^{kT}\left(\sum_{j=0}^{n-1}(V-KUV)KRK^T(V-KUV)^T\right)^{-1}(V-KUV)(KU)$$

5. Stuff For Poster

$P(\hat{x}_n|y+\delta y)$

$P(\hat{x}_n|y)$

$P(x_n)$

$x_n^1\ x_n^2\ \hat{x}_n^1\ \hat{x}_n^2$

$D_{KL}(P(\hat{x}_n|y)||P(\hat{x}_n|y+\delta y)) = \frac{1}{2}\delta y^T J\delta y$

$$J_{rs} = \left\langle \frac{\partial^2}{\partial y_{n-r}\partial y_{n-s}} \log P(\hat{x}_n|y)\right\rangle \tag{5.1}$$

$$J_{rs} = \frac{\partial \mu^T}{\partial y_{n-r}} C^{-1} \frac{\partial \mu}{\partial y_{n-s}} \tag{5.2}$$

$$J_{rs} = G_r^T\left(\sum_{i=0}^{n-1}(H_iQH_i^T + G_iRG_i^T)\right)^{-1} G_s \tag{5.3}$$

$$G_i = \left(\prod_{k=0}^{i-1}(V - K_{n-k}UV)\right)K_{n-i} \tag{5.4}$$

$$H_i = \sum_{j=0}^{i} G_iUV^{i-j} \tag{5.5}$$

$x_{t+1}\ Vx_t + \nu\ Ux_{t-1} + \mu\ Ux_{t+1} + \mu$
$\hat{x}_{t-1}^-\ \hat{x}_{t+1}^-\ y_{t-1}\ y_{t+1}$
$\hat{x}_{t+1}\ \hat{x}_{t-1}^- + K_{t-1}(y_{t-1} - U\hat{x}_{t-1}^-)$
$\hat{x}_{t+1}^- + K_{t+1}(y_{t+1} - U\hat{x}_{t+1}^-)$

In the standard Kalman filter, x_t is a hidden variable that is of interest. In the discrete time case, an observation y_t is generated at each timestep. The Kalman filter is designed such that the variable $\hat{x}_t$ provides the optimal estimate for x_t. $\hat{x}_t^-$ represents the predicted value of x_t prior to the observation at time t. $\hat{x}_t$ is then generated by adjusting the predicted value by a weighted average of y_t and $\hat{x}_t^-$. Since y_t is corrupted by noise, this weighted average helps to denoise the signal to provide an accurate estimate for x_t.

The ability to accurately estimate x_t is affected by y. An infinitesimally small perturbation in y shifts $\hat{x}_t$ by an amount directly proportional to J,the FIM.

where μ is the expected value of $\hat{x}_n$ and C is the covariance of $\hat{x}_n$. Following a good deal of manipulations, this can be expressed in terms of the Kalman filter parameters as

References

[1] M. Schmidt, H. Lipson, *Distilling free-form natural laws from experimental data*, Science, **324** (2009), 81–85.

[2] M. Schmidt, H. Lipson, *Distilling free-form natural laws from experimental data*, Supplementary online materials.