Extended Kalman Filter using Normalizing Flow and Variational Inference

علی ایزدی

تیر ۱۴۰۰

فهرست مطالب

فهرست مطالب **2**

۱- مقدمه **3**

۲- مرور VAE و variational inference with normalizing flow **4**

۱-۱- variational autoencoder 4

۱-۲- variational inference with normalizing flow 5

۳- Deep kalman filter **6**

۳-۱- Encoder or inference model or recognition model 6

۳-۲- Decoder or generative model 7

۴- روش پیشنهادی **8**

۵- مروری بر کارهای گذشته **9**

مراجع **11**

۱- مقدمه

در قسمت ۲ ابتدا به طور مختصر به بررسی مقالات variational autoencoder و سپس variational inference with normalizing flow خواهیم پرداخت.

در قسمت ۳ نیز به توضیح deep kalman filter که روشی برای extended kalman filter بوده و از VAE برای حل این مسئله استفاده کرده است، خواهیم پرداخت.

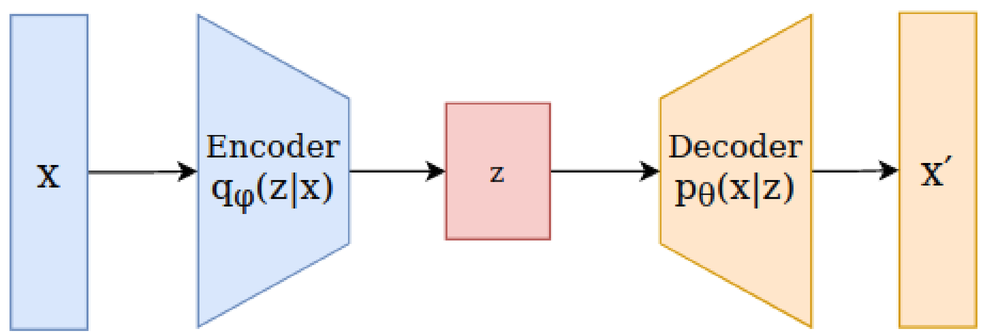
همان گونه که مقاله normalizing flow توانسته است برای توزیع احتمال پسین در ساختار VAE از normalizing flow استفاده کند و توزیع احتمال پسین دقیق تر برسد، می توان از این روش استفاده کرد و برای حل مسئله extended kalman filter از normalizing flow در ساختار variational inference استفاده کرد. در قسمت ۴ به این موضوع خواهیم پرداخت.

در قسمت ۵ نیز مقاله normalizing kalman filter با این روش مقایسه شده است.

۲- مرور VAE و variational inference with normalizing flow

۱-۱- variational autoencoder

همان طور که میدانیم که ساختار آن مطابق شکل ۱ است.



شکل ۱- variational autoencoder

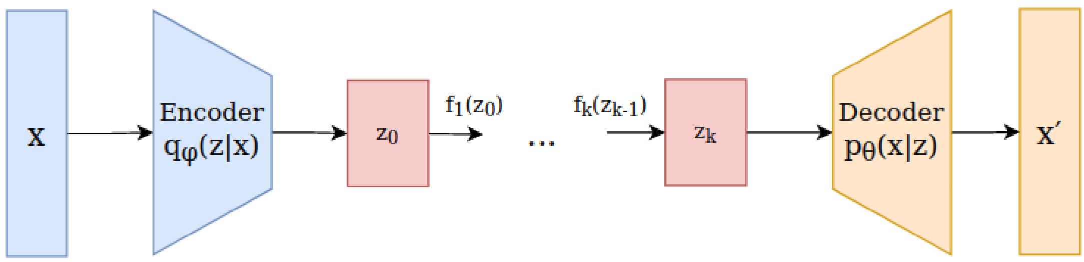
در VAE هدف بهینه سازی رابطه ۱ که یک lower bound برای likelihood بوده، است.

[۱]

$$F(x) = E_{q_\phi(z \mid x)}[\log p_\theta(x \mid z)] + D_{KL}[q_\phi(x \mid z)||p(z)]$$

۱-۲- variational inference with normalizing flow

همان طور که میدانیم ساختار آن مشابه با variational inference با استفاده از VAE است با این تفاوت که توزیع احتمال پسین q توسط normalizing flow مدل میشود. ساختار این مدل مشابه با شکل ۲ است.



.

شکل ۲- variational inference with normalizing flow

همچنین در این حالت lower bound مطابق با عبارت ۲ به دست می آید.

[۲]

$$
\begin{aligned}
F(x) &= E_{q_\phi(z\,|\,x)}[\log p_\theta(x\,|\,z)] + D_{KL}[q_\phi(x\,|\,z)||p(z)] \\
&= E_{q_\phi(z\,|\,x)}[\ \log q_\phi(x\,|\,z) - \ \log p_\theta(x,\,z)] \\
&= E_{q_0(z_0)}[\ \log q_k(z_k) - \ \log p_\theta(x,\,z_k)] \\
&= E_{q_0(z_0)}[\ \log q_0(z_0)] - \ E_{q_0(z_0)}[\ \log p_\theta(x,\,z_k)] - E_{q_0(z_0)}\left[\sum_{i=1}^{k} \log \left|\det \frac{\delta f_i}{\delta z_{i-1}}\right|\right] \\
&= E_{q_0(z_0)}[\ \log p_\theta(x\,|\,z_k)] + E_{q_0(z_0)}[\ \log q_0(z_0) - \ \log p_\theta(z_k)] - E_{q_0(z_0)}\left[\sum_{i=1}^{k} \log \left|\det \frac{\delta f_i}{\delta z_{i-1}}\right|\right]
\end{aligned}
$$

Deep kalman filter -۳

مقاله deep kalman filter با استفاده از ایده VAE روشی برای extended kalman filter ارائه داده است.

روابط Gaussian state space model در زیر آورده شده است.

[۳]

$$
\begin{aligned}
z_t &\sim \mathcal{N}(G_\alpha(z_{t-1}, \Delta_t), S_\beta(z_{t-1}, \Delta_t)) && \textit{(Transition)} && (1) \\
x_t &\sim \Pi(F_\kappa(z_t)) && \textit{(Emission)} && (2)
\end{aligned}
$$

چون توزیع احتمال پسین p(z|x) در این حالت محاسبه آن intractable بوده مقاله از روش variational inference برای تقریب این توزیع با استفاده از ساختار VAE و تخمین q(z|x) به عنوان Encoder استفاده کرده است.

در ادامه ساختار Encoder و Decoder این مدل توضیح داده شده است.

Encoder or inference model or recognition model -۱-۳

Encoder میانگین و کواریانس z را در q(z|x) تخمین میزند(که به خودی خود تخمین خوبی برای p(z|x) نیست پس همان طور که در قسمت های بعدی توضیح داده خواهد شد میتوان از normalizing flow برای q(z|x) استفاده کرد).

چون هدف حل مسئله state space است، پس باید این میانگین و کواریانس برای توزیع q(z|x) وابسته به زمان و z(t-1) باشد.

روابط محاسبه این میانگین و کورایانس در زیر آمده است. در رابطه های پایین h(t) با استفاده از RNN به دست می آید.

$$h_{\text{combined}} = \frac{1}{2}(\tanh(W z_{t-1} + b) + h_t^{\text{right}})$$

$$\textit{(Posterior Means and Covariances)}$$

$$\mu_t = W_\mu h_{\text{combined}} + b_\mu$$

$$\sigma_t^2 = \text{softplus}(W_{\sigma^2} h_{\text{combined}} + b_{\sigma^2})$$

۳-۲- **Decoder** or generative model

همان طور که در روابط state space model مشاهده میشود مدل generative شامل توابع G و S و F که G و S مربوط به state transition و F مربوط به emission بین state و observation است. این توابع را میتوان با استفاده از MLP پیاده سازی کرد.

روابط G و S که برای محاسبه میانگین و کواریانس z(t) در روابط state space model استفاده میشود در زیر آمده است.
نکته مهم در روابط زیر وابستگی z(t) به z(t-1) به صورت غیر خطی است.

$$g_t = \text{MLP}(z_{t-1}, \text{ReLU}, \text{sigmoid}) \quad \textit{(Gating Unit)}$$

$$h_t = \text{MLP}(z_{t-1}, \text{ReLU}, \mathbb{I}) \quad \textit{(Proposed mean)}$$

$$\textit{(Transition Mean } G_\alpha \textit{ and } S_\beta\textit{)}$$

$$\mu_t(z_{t-1}) = (1 - g_t) \odot (W_{\mu_p} z_{t-1} + b_{\mu_p}) + g_t \odot h_t$$

$$\sigma_t^2(z_{t-1}) = \text{softplus}(W_{\sigma_p^2} \text{ReLU}(h_t) + b_{\sigma_p^2})$$

حال زمانی که z(t) با استفاده از روابط بالا به دست آمد میتوان x(t) را با استفاده از تابع F به دست آورد. تابع F نیز با استفاده از یک MLP مطابق با روابط زیر پیاده سازی شده است.

$$F_\kappa(z_t) \;=\; (W_{\text{emission}}\text{MLP}(z_t, \text{ReLU}, \text{ReLU}) + b_{\text{emission}})$$

در نتیجه با مشخص شدن ساختار encoder و decoder مدل بالا مطابق با ساختار variational autoencoder در شکل ۱ و هم چنین بهینه سازی lower bound مطابق با رابطه ۱ به مسئله روشی برای extended kalman filter ارائه میدهد.

مشکل روش بالا در ساده بودن توزیع احتمال پسین q(z|x) است که تخمین دقیقی از توزیع احتمال پسین p(z|x) نیست به خصوص که در این مقاله ماتریس کواریانس به صورت قطری در نظر گرفته شده است.

۴- روش پیشنهادی

همان طور که در قسمت قبل توضیح داده شد توزیع احتمال پسین q(z|x) با در نظر گرفتن یک توزیع گوسی ساده z تخمین دقیقی برای‌p(z|x) نیست.

یک روش برای حل این مشکل همان گونه که در قسمت ۲ توضیح داده شد. استفاده از مقاله variational inference with normalizing flow برای تخمین دقیق تر این توزیع احتمال پسین با استفاده از normalizing flow است.

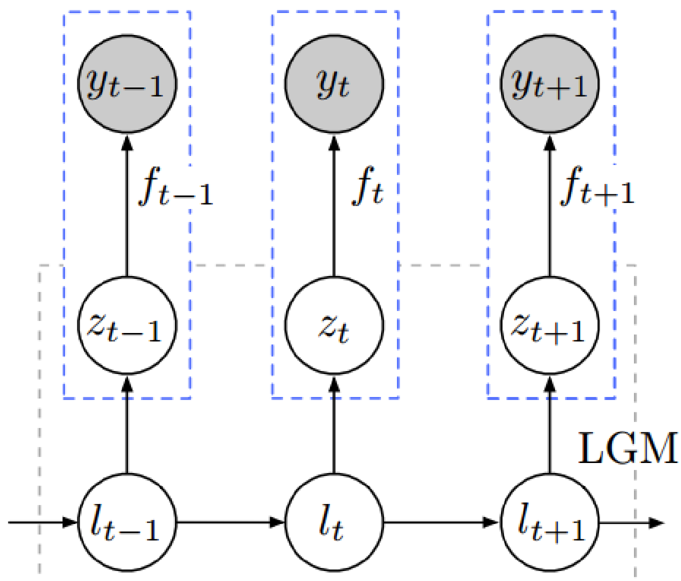
بنابراین روش پیشنهادی جدید شامل اضافه کردن normalizing flow به مدل deep kalman filter است.

۵- مروری بر کارهای گذشته

مقاله normalizing kalman filter برای تابع F یعنی تابع emission روش استفاده از normalizing flow را ارائه داده است.

$$\text{(NKF model)}\quad \begin{aligned} &\mathbf{l}_1 \sim \mathcal{N}(\mu_1, \Sigma_1) && && \text{(initial state)} \\ &\mathbf{l}_t = F_t \mathbf{l}_{t-1} + \boldsymbol{\epsilon}_t, && \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \Sigma_t), && \text{(transition dynamics)} \\ &\mathbf{y}_t = f_t(A_t^T \mathbf{l}_t + \boldsymbol{\varepsilon}_t), && \boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, \Gamma_t). && \text{(observation model)} \end{aligned}$$

در نتیجه ساختار آن مشابه با شکل ۳ است.



شکل ۳- observationهای y با استفاده از معکوس تابع f توسط یک normalizing flow به z تبدیل شده اند.

در این روش اثبات میشود که

$$p(l_t|y_{1:t}; \Theta, \Lambda) = p_{\text{LGM}}\left(l_t|z_{1:t}; \Theta\right)$$

$$\ell(\Theta, \Lambda) = p(y_{1:T}; \Theta, \Lambda) = \prod_{t=1}^{T} p_{\text{LGM}}(z_t|z_{1:t-1}; \Theta)\, |\det[\text{Jac}_{z_t}(f_t)]|^{-1}$$

در نتیجه برای تخمین پارامترها با استفاده از likelihood و حل مسئله فیلترینگ فرم بسته وجود دارد. همان طور که در kalman filter خطی نیز فرم بسته وجود دارد.

در نتیجه در مقایسه با روش پیشنهادی ما در موارد زیر تفاوت وجود دارد:

- تابع state transition در روش پیشنهادی ما غیر خطی بوده در صورتی که در این مقاله خطی بوده و تنها ماتریس های تابع خطی آن توسط یک RNN تخمین زده میشود که همچنان state transition خطی میماند.

- روش پیشنهادی ما lower bound را بهینه میکند در صورتی که این مقاله likelihood را به صورت مستقیم بهینه میکند. البته در روش پیشنهادی ما چون از normalizing flow استفاده میشود یک lower bound را tight میکند.

- اگر به صورت دقیق تر ساختار VAE مدل پیشنهادی ما و ساختار generative این مقاله را مقایسه کنیم:

در مدل پیشنهادی ما normalizing flow به q(z|x) اعمال میشود و سپس decoder از این توزیع، p(x|z) یا observationها را تولید میکند.

اما در مقاله بالا normalizing flow مستقیما به p(x|z) اعمال میشود.

پس آیا میتوان نتیجه گرفت normalizing flow در هر دو مدل به پیچیده تر کردن تابع emission یا p(x|z) کمک میکند؟

اما همچنان تفاوت روش پیشنهادی در غیر خطی کردن تابع state transition با روش این مقاله وجود دارد که میتواند باعث بهبود مدل شود.

مراجع

[1] Kingma, D.P. and Welling, M., 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[2] Rezende, D. and Mohamed, S., 2015, June. Variational inference with normalizing flows. In *International Conference on Machine Learning* (pp. 1530-1538). PMLR.

[3] Krishnan, R., Shalit, U. and Sontag, D., 2017, February. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 31, No. 1).

[4] de Bézenac, E., Rangapuram, S.S., Benidis, K., Bohlke-Schneider, M., Kurle, R., Stella, L., Hasson, H., Gallinari, P. and Januschowski, T., 2020. Normalizing Kalman Filters for Multivariate Time Series Analysis. *Advances in Neural Information Processing Systems*, *33*.