Notes for Super Resolution Microscopy

Sky

Columbia University

1 Single Particle CVI

1.1 Simplified Problem Setup

We define our state-space model by the following distributions:

$$
\begin{align}
s_{1,1} &\sim \mathcal{N}(\mu_1, C) \tag{1}\\
s_{1,t}|s_{1,t-1} &\sim \mathcal{N}(As_{1,t-1}, Q) \ \forall \ t \geq 1 \tag{2}\\
y_{1,t} &\sim \text{Poisson}(s_{1,t}) \tag{3}
\end{align}
$$

with means and variances of dimension $\mathbb{R}^{2\times 1}$ and $\mathbb{R}^{2\times 2}$ respectively. We also have $\boldsymbol{S} = \boldsymbol{s}_{1:}$ and $\boldsymbol{Y} = \boldsymbol{y}_{1:}$.
Using the above definitions, the joint likelihood is:

$$
p(\boldsymbol{S}, \boldsymbol{Y}) = p(s_{1,1}) \prod_{t=2}^{T} p(s_{1,t}|s_{1,t-1}) \prod_{t=1}^{T} p(y_{1,t}|s_{1,t}) \tag{4}
$$

1.2 CVI for Simplified Problem

In this subsection, we demonstrate how to find the closest approximation with CVI. A possible approximation of posterior distribution $q(\boldsymbol{S})$ in state-space models is using a multivariate Gaussian distribution. Let us begin by breaking down the joint likelihood into conjugate and non-conjugate parts:

$$
p(\boldsymbol{S}, \boldsymbol{Y}) \propto \underbrace{p(s_{1,1}) \prod_{t=2}^{T} p(s_{1,t}|s_{1,t-1})}_{\text{conjugate}} \underbrace{\prod_{t=1}^{T} p(y_{1,t}|s_{1,t})}_{\text{non-conjugate}} \tag{5}
$$

We note that the first two components are Gaussian and the last is Poisson. In practice, the last term which we denote as the observation term, is approximated with a Gaussian to enable us to take advantage of its properties.

$$
P(y_{1,t}|s_{1,t}) \approx \mathcal{N}(y_{1,t}|\tilde{y}_{1,t}, \tilde{\sigma}^2_{1,t}) \tag{6}
$$

and its exponential form is $h(z)\exp\{\langle \boldsymbol{\lambda}, \boldsymbol{\phi(z)} \rangle - A(\boldsymbol{\lambda}\})$ with $\boldsymbol{\lambda} = [\lambda^{(1)}, \lambda^{(2)}]$ whose $\hat{\mu} = \hat{\sigma}^2\lambda^{(1)}$ and $\hat{\sigma^2} = -\frac{1}{2}{\lambda^{(2)}}^{-1}$.

First, we compute the joint Gaussian distribution $q_c(\boldsymbol{S})$ of the conjugate terms by expanding them and obtain its natural parameter:

$$q_c(\boldsymbol{S}) = p(s_{1,1}) \prod_{t=2}^{T} p(s_{1,t}|s_{1,t-1}) \tag{7}$$

$$\propto \exp\left\{-\frac{1}{2}\left((s_{1,1}-\mu_1)^T C^{-1}(s_{1,1}-\mu_1) + \sum_{t=2}^{T}(s_{1,t}-As_{1,t-1})^T Q^{-1}(s_{1,t}-As_{1,t-1})\right)\right\} \tag{8}$$

$$\begin{aligned}\propto \exp\Bigg\{-\frac{1}{2}\Big(&s_{1,1}^T C^{-1} s_{1,1} - 2s_{1,1}^T C^{-1}\mu_1 + \mu_1^T C^{-1}\mu_1 \\ &+\sum_{t=2}^{T} s_{1,t}^T Q^{-1} s_{1,t} - 2s_{1,t}^T Q^{-1} A s_{1,t-1} + (As_{1,t-1})^T Q^{-1} A s_{1,t-1}\Big)\Bigg\}\end{aligned} \tag{9}$$

$$\propto \exp\left\{-\frac{1}{2}\left(\boldsymbol{S^T H S} - 2\boldsymbol{S^T G}\right)\right\} \tag{10}$$

$$\propto \exp\left\{-\frac{1}{2}\left(\boldsymbol{S^T H S} - 2\boldsymbol{S^T H H^{-1} G} + \boldsymbol{(H^{-1}G)^T H H^{-1} G}\right)\right\} \tag{11}$$

$$\propto \exp\left\{-\frac{1}{2}(\boldsymbol{S} - \boldsymbol{H^{-1}G})^T \boldsymbol{H}(\boldsymbol{S} - \boldsymbol{H^{-1}G})\right\} \tag{12}$$

where $\boldsymbol{S}$ is the $T \times 1$ vector containing $s_{1,1}$ to $s_{1,T}$, $\boldsymbol{H}$ the $T \times T$ block diagonal matrix of form

$$\begin{bmatrix} C^{-1}+A^TQ^{-1}A & -Q^{-1}A & 0 & \dots & \dots & 0 \\ -Q^{-1}A & Q^{-1}+A^TQ^{-1}A & -Q^{-1}A & \dots & \dots & \vdots \\ 0 & -Q^{-1}A & Q^{-1}+A^TQ^{-1}A & \dots & \dots & \vdots \\ 0 & 0 & 0 & \ddots & \dots & \vdots \\ 0 & 0 & 0 & \ddots & Q^{-1}+A^TQ^{-1}A & -Q^{-1}A \\ 0 & 0 & 0 & \dots & -Q^{-1}A & Q^{-1} \end{bmatrix} \tag{13}$$

and $\boldsymbol{G}$ is the $T \times 1$ vector $[C^{-1}\mu_1, 0, 0, \dots 0]$. Interrogating $q_c(\boldsymbol{S})$, we can see that its natural parameters $\boldsymbol{\lambda} = [\boldsymbol{G}, -\frac{1}{2}\boldsymbol{H}]$ and variational parameters $\boldsymbol{\theta} = [\boldsymbol{H^{-1}G}, \boldsymbol{H^{-1}}]$. In practice, $\boldsymbol{H}$ is of size $2T \times 2T$ with each entry being a 2×2 block. To ensure efficient computation of $\boldsymbol{m}$, we flatten $\boldsymbol{G}$ to a $2T \times 1$ matrix before multiplying with $\boldsymbol{H^{-1}}$ and reshape the resulting matrix to $2 \times T$ and taking its transpose.

Second, we apply the CVI update rule for our Gaussian approximation of the non-conjugate term:

$$\tilde{\lambda}_k^{(i)} = (1-\beta_k)\tilde{\lambda}_{k-1}^{(i)} + \beta_k \hat{\nabla}_{\mu^{(i)}} \mathbb{E}_q[\log P(\boldsymbol{Y}|\boldsymbol{S})]\big|_{\mu=\mu_k^{(i)}} \tag{14}$$

with k the number of gradient descent steps taken and $\mu_k^{(i)}$ the i^{th} mean of the overall distribution $q(\boldsymbol{S})$ respectively.
We use Monte Carlo integration to compute $\hat{\nabla}_{\mu^{(i)}}\mathbb{E}_q[\log P(\boldsymbol{Y}|\boldsymbol{S})]|_{\mu=\mu_k^{(i)}}$. Following [KL17], we set $f_n = \mathbb{E}_q[\log P(y_{1,t}|s_{1,t})]|$ with $q(s_{1,t}) = \mathcal{N}(z_{1,t}|m_{1,t}, V_{(1,t),(1,t)})$ and express the mean parameters $\mu_n^{(1)} = m_n$ and $\mu_n^{(1)} = V_{nn} + m_n^2$. By applying the chain rule, we can write the gradients in terms of their variational parameters:

$$\frac{\partial f_n}{\partial \mu_n^{(1)}} = \frac{\partial f_n}{\partial m_n}\frac{\partial m_n}{\partial \mu_n^{(1)}} + \frac{\partial f_n}{\partial V_{nn}}\frac{\partial V_{nn}}{\partial \mu_n^{(1)}} = \frac{\partial f_n}{\partial m_n} - 2\frac{\partial f_n}{\partial V_{nn}}m \tag{15}$$

$$\frac{\partial f_n}{\partial \mu_n^{(2)}} = \frac{\partial f_n}{\partial m_n}\frac{\partial m_n}{\partial \mu_n^{(2)}} + \frac{\partial f_n}{\partial V_{nn}}\frac{\partial V_{nn}}{\partial \mu_n^{(2)}} = \frac{\partial f_n}{\partial V_{nn}} \tag{16}$$

We observe from [OA09] that fitting a local Laplace approximation is equivalent to rewriting the gradients of the expectation as:

$$\frac{\partial f_n}{\partial m_n} = \mathbb{E}_q\left[\nabla_{s_{1,t}} f_n\right], \ \frac{\partial f_n}{\partial V_{nn}} = \frac{1}{2}\mathbb{E}_q\left[\nabla_{s_{1,t}}\nabla_{s_{1,t}} f_n\right] \tag{17}$$

Begin by converting the Poisson distribution to its exponential form

$$\text{Poisson}(y_{1,t}|s_{1,t}) = s_{1,t}^{y_{1,t}}\frac{e^{-s_{1,t}}}{y_{1,t}!} \tag{18}$$

$$= \frac{1}{y_{1,t}!}\exp\left\{y_{1,t}\log s_{1,t} - s_{1,t}\right\} \tag{19}$$

Next, we proceed to compute the functions within the expectations in 17:

$$\nabla_{s_{1,t}} \log P(y_{1,t}|s_{1,t}) = \nabla_{s_{1,t}}\left(-\log y_{1,t}! + y_{1,t}\log s_{1,t} - s_{1,t}\right) \tag{20}$$

$$= \frac{y_{1,t}}{s_{1,t}} - 1 \tag{21}$$

$$\nabla_{s_{1,t}}\nabla_{s_{1,t}} \log P(y_{1,t}|s_{1,t})] = \nabla_{s_{1,t}}\frac{y_{1,t}}{s_{1,t}} - 1 \tag{22}$$

$$= -\frac{y_{1,t}}{s_{1,t}^2} \tag{23}$$

Lastly, we take the average of a set number as an approximate to the expectations. With the above in place, we have all the pieces we need for CVI.
Here we include a brief note on our efforts in obtaining a closed-form solution to the gradients of the expectation and why we were unsuccessful. We began by looking at [Kha12] after the authors of [KL17] pointed out that the expectation of a Gaussian approximation to the Poisson distribution could be expressed analytically. However in the paper, the analytical form of such an expectation was in the natural parameter space instead of the variational parameter space we desired. We tried brute force integration of the Poisson log-likelihood and its gradients but was stymied by having to compute the expectation with respect to

a Gaussian of $\log s_{1,t}$, $\frac{1}{s_{1,t}}$ and $\frac{1}{s_{1,t}^2}$ respectively. We later found a Taylor series approximation for the expectation of $\log s_{1,t}$ in [TNW06] which worked well for higher values of $y_{1,t}$ but performed very poorly for values close to 0, Upon further scrutiny, we realized that the approximation had a caveat: the approximation is only very accurate when $\mathbb{E}_q[s_{1,t}] >> 0$. Although [TNW06] noted that the approximation works very well even when $s_{1,t}$ is small, in our case, our $s_{1,t}$ is too small for it to work. Hence we had to resort to Monte Carlo integration.

1.3 Realistic Problem Setup

Moving to the more realistic problem setup, we have:

$$s_{1,1} \sim \mathcal{N}(\mu_1, C) \tag{24}$$

$$s_{1,t}|s_{1,t-1} \sim \mathcal{N}(As_{1,t-1}, Q)\ \forall\ t \geq 1 \tag{25}$$

$$P(y_{1,t}|\boldsymbol{s_{1,t}}) = P(y_{1,t}|I_t) = \prod_{x=1}^{L}\prod_{y=1}^{L} e^{-[BI_t]_{xy}} \frac{[BI_t]_{xy}^{Y_{t,xy}}}{Y_{t,xy}!} \tag{26}$$

where B is a point spread function and I_t the high resolution image at time t.

1.4 CVI for Single Particle Tracking

As the above setup is identical to the simplified problem setup apart from the Poisson likelihood function, we only have to compute the derivative of the new likelihood function with respect to $s_{1,t}$. Following the previous section, we first write the likelihood in its exponential form:

$$P(y_{1,t}|\boldsymbol{s_{1,t}}) = \prod_{x=1}^{L}\prod_{y=1}^{L} \frac{1}{Y_{t,xy}!} \exp\left\{Y_{t,xy} \log[B * I_t]_{xy} - [B * I_t]_{xy}\right\} \tag{27}$$

In this case, since the particle in $s_{1,t}$ only exist at a fixed coordinate within the image I_t we only obtain a non-zero derivative at that position via the chain rule as shown below:

$$\nabla_{s_{1,t}} \log P(y_{1,t}|s_{1,t}) = \frac{\partial}{\partial I_t}\frac{\partial}{\partial s_{1,t}} \sum_{x=1}^{L}\sum_{y=1}^{L} - \log Y_{t,xy}! + Y_{t,xy} \log[B * I_t]_{xy} - [B * I_t]_{xy} \tag{28}$$

$$= \frac{\partial}{\partial s_{1,t}} \sum_{x=1}^{L}\sum_{y=1}^{L} \frac{Y_{t,xy}}{[B * I_t]_{xy}} - B_{xy} \tag{29}$$

$$= \sum_{x=1}^{L}\sum_{y=1}^{L} B_{s_{1,t}} \frac{Y_{t,s_{1,t}}}{[B * I_t]_{s_{1,t}}} - B_{s_{1,t}} \tag{30}$$

$$\nabla_{s_{1,t}}\nabla_{s_{1,t}} \log P(y_{1,t}|s_{1,t}) = \sum_{x=1}^{L}\sum_{y=1}^{L} -B_{s_{1,t}}^2 \frac{Y_{t,s_{1,t}}}{([B * I_t]_{s_{1,t}})^2} \tag{31}$$

Algorithm 1 CVI for Kalman Filter with non-conjugate likelihood

```
Initialize $\tilde{\boldsymbol{\lambda}}_0 = 0$ and $\boldsymbol{\lambda}_1 = \theta = [\boldsymbol{G}, -\frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{H}]$
for k = 1, 2, 3, ... till convergence do
    $\boldsymbol{\Sigma_k} = -\frac{1}{2}\left(\lambda_k^{(2)}\right)^{-1}$
    $\boldsymbol{\mu_k} = \boldsymbol{\Sigma_t \lambda_k^{(1)}}$
    $\tilde{\lambda}_k = (1-\beta_k)\tilde{\lambda}_{k-1} + \beta_k \hat{\nabla}_\mu \mathbb{E}_q[\log P(\boldsymbol{Y}|\boldsymbol{S})]|_{\mu=\mu_k}$
    $\boldsymbol{\lambda}_{k+1} = \tilde{\boldsymbol{\lambda}}_k + \boldsymbol{\theta}$
```

Using λ_K, we can calculate the mean and the variance of our approximating Gaussian $\hat{\mu} = \hat{\sigma}^2\lambda_K^{(1)}$ and $\hat{\sigma^2} = -\frac{1}{2\lambda_K^{(2)}}$.

2 Multiple Particle CVI

We use the exact same setup in the multiple particle case where each particle is modelled using its own individual state-space model. Following the previous setup, we model each particle as a vector of length T with each entry containing the x and y coordinates of the particle. The vectors are then stacked as a $J \times T \times 1$ tensor where J is the number of particles in each frame. To make the problem easier, we make the assumption that the number of particles per frame stays fixed.

3 To Do

1. Get model working for single particle images
2. Visualise and understand multiple particle data
3. Figure out model for multiple particle data

4 Done

– Understand how gradient descent is optimising a quadratic approximation to $\mathcal{L}$ [1] [2]
– Revise exponential families (FoGM), HMMs (SML, MLEE)
– Read state-space paper, Kalman Filters (PML)
– Visualise and understand single particle data
– Learn to use TrackMate and compute results for tracking performance for single particles
– Understand Khan and Lin 2017
– Understand how Kalman filters with non-Gaussian likelihoods can be approximated with a multivariate Gaussian (MLPP 654) and how it can be expressed in exponential form

[1] `https://homes.cs.washington.edu/~marcotcr/blog/gradient-descent/`

[2] `https://www.cs.cmu.edu/~ggordon/10725-F12/slides/05-gd-revisited.pdf`

- Write algorithm for single particle CVI
- Revise GLM for poisson likelihoods
- Understand the use of MC integration in Khan and Lin
- Fix η computation
- Generate data with poisson likelihood
- Check that derivatives obtained from GPML are right
- Revise VI, stochastic approximation (BMML 86)

5 Ideas

References

[TNW06] Yee Whye Teh, David Newman, and Max Welling. “A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation”. In: *NIPS*. Vol. 6. 2006, pp. 1378–1385.

[OA09] Manfred Opper and Cédric Archambeau. “The Variational Gaussian Approximation Revisited”. In: *Neural computation* 21.3 (2009), pp. 786–792.

[Kha12] Mohammad Khan. “Variational learning for latent Gaussian model of discrete data”. In: (2012).

[KL17] Mohammad Emtiyaz Khan and Wu Lin. “Conjugate-Computation Variational Inference: Converting Variational Inference in Non-Conjugate Models to Inferences in Conjugate Models”. In: *arXiv preprint arXiv:1703.04265* (2017).