Combining Variational Inference and Sequential MC methods

Tutorial for PGM Class Rika Antonova

Introduction and Motivation

The goal of this tutorial is to give an introduction to techniques that combine Variational Inference (VI) and Monte Carlo (MC) methods. **Prior familiarity with VI and SMC is required to complete this tutorial. Interested students could get some of the necessary background in [5] and [6]. But be warned that studying both of these would take a significant amount of time, if one is starting 'from scratch'**. This tutorial starts with a brief overview of general principles underlying VI and MC methods, pointing out their strengths and limitations. Variational Inference (VI) is a class of efficient methods for inference in Graphical Models. VI can be used effectively to quickly determine a crude approximation to a model's posterior. VI approaches first select a parametric family of distributions, then optimize its parameters. However, if the parametric family chosen for VI is too restrictive, the resulting approximation might fail to capture important aspects of the posterior (e.g. fail to express multimodality). Monte Carlo approaches in general, and Sequential Monte Carlo (SMC) in particular, can be used to propose more complex families of distributions for approximation. However, they tend to be computationally expensive. Recently, these approaches have been combined with VI to obtain flexible and efficient alternative to using either VI or MC alone. Three groups proposed combining VI and SMC, publishing almost simultaneously [1], [2], [3]. This tutorial focuses on the line of work from [1], which shows that SMC estimator of the marginal likelihood can serve as a variational objective. This tutorial gives an introduction to using this combined technique and analyzes the practical advantages offered by this combination of VI and SMC.

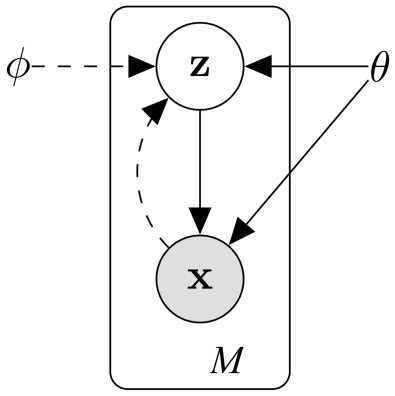
A Brief Introduction to Variational Inference

We consider dataset $\boldsymbol{x} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{M}$ that consist of M iid (independent and identically distributed) samples. This is our *observed* data, shown in a circle with gray background in Figure 1. We assume that the data is generated by a random process that involves an *unobserved* variable $\boldsymbol{z}$, illustrated by a circle with white background in Figure 1. The data generation process consists of 2 steps:

1. $\boldsymbol{z}^{(i)}$ is generated from a *prior* distribution $p_{\boldsymbol{\theta}_{true}}(\boldsymbol{z})$
2. $\boldsymbol{x}^{(i)}$ is generated from a conditional distribution $p_{\boldsymbol{\theta}_{true}}(\boldsymbol{x}|\boldsymbol{z})$, called *likelihood*

We assume that the distributions involved generating the data come from parametric families $p_{\boldsymbol{\theta}}(\boldsymbol{z})$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ that are differentiable w.r.t $\boldsymbol{\theta}, \boldsymbol{z}$. We assume that $\boldsymbol{\theta}_{true}$ parameters are not known, and latent variables $\boldsymbol{z}$ are not observed. In our setting, integrating *marginal likelihood* $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z}$ is intractable. Finding the exact *posterior* density $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}) / p_{\boldsymbol{\theta}}(\boldsymbol{x})$ is also intractable. Variational Inference (VI) makes use of a variational bound to allow learning parameters $\boldsymbol{\phi}$ of a *proposal* distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ that serves as approximation to $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. Right side of Figure 1 gives a summary of the notation for this setting.

Note: to avoid clutter we use $\boldsymbol{\theta}$ to denote parameters of various distributions, meaning $\boldsymbol{\theta}$ represents some kind of parameters (e.g. mean and variance vectors, or weights of a neural network, etc); however we do not mean to imply that the prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ and likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ literally share the same parameter values.



$$prior: p_{\boldsymbol{\theta}}(\boldsymbol{z})$$
$$likelihood: p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$$
$$marginal\ likelihood: p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z}$$
$$posterior: p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{x})}$$
$$proposal: q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$$
$$\big(\text{approximates intractable posterior } p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\big)$$

Figure 1: A generic model from [4] for illustrating VI principles (left) and a summary of the notation (right).

Maximum likelihood estimation (MLE) method provides an estimate for model parameters by maximizing observed data likelihood, in practice – data log likelihood: $\log p(\boldsymbol{x}) = \log \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}$. To achieve this in our setting, we can instead optimize evidence lower bound (ELBO):

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\Big[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\Big] \leq \log p(\boldsymbol{x}) \tag{1}$$

Exercise 1: Show that Equation 1 is a lower bound on the data log likelihood: $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}) \leq \log p(\boldsymbol{x})$.

With this, we can maximize lower bound $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi})$ with gradient ascent, since we assumed that p is differentiable w.r.t $\boldsymbol{\theta}, \boldsymbol{z}$, and we can choose proposal q that is differentiable w.r.t $\boldsymbol{\phi}, \boldsymbol{z}$. This yields MLE estimates for $\boldsymbol{\phi}$, allowing us to draw samples from $q_{\boldsymbol{\phi}}$ that approximate posterior $p_{\boldsymbol{\theta}}$. Further extensive introduction to VI can be found in [5].

Sequential Monte Carlo Methods

There are many settings when data generating process is known to have sequential structure. For example: handwriting, speech, music, dynamical systems representing physical processes. In such cases the graphical model can be unrolled into a sequence of observed variables $\boldsymbol{x}_1, ..., \boldsymbol{x}_T$ generated by latent variables $\boldsymbol{z}_1, ..., \boldsymbol{z}_T$. A commonly used assumption is that this sequential latent variable model can be factored as a series of tractable conditionals:

$$p(x_{1:T}, z_{1:T}) = p_1(x_1, z_1) \prod_{t=2}^{T} p_t(x_t, z_t | x_{1:t\text{-}1}, z_{1:t\text{-}1})$$

Sequential Monte Carlo (SMC), also known as particle filtering, propagates N weighted particles for T steps using a combination of importance sampling and resampling. Each particle represents a likely path of length t that is extended at each step, until step T is reached. As a result, the algorithm provides a way to generate sample paths of length T from posterior, without exact analytic computations.

The input to the algorithm is: the number of particles N, model distribution p, observations $x_{1:T}$. SMC maintains population $\{w_{t\text{-}1}^{(i)}, z_{1:t\text{-}1}^{(i)}\}_{i=1}^{N}$ of particles $z_{1:t\text{-}1}^{(i)}$ with weights $w_{t\text{-}1}^{(i)}$. After step t, the algorithm proposes extension $z_t^{(i)} \sim q_t(z_t | x_{1:t}, z_{1:t\text{-}1}^{(i)})$ to each particle trajectory $z_{1:t\text{-}1}^{(i)}$. Weights are multiplied by importance weights $\alpha\big(z_{1:t}^{(i)}\big)$ using:

$$\alpha\big(z_{1:t}^{(i)}\big) = \frac{p_t(x_t, z_t^{(i)} | x_{1:t\text{-}1}, z_{1:t\text{-}1}^{(i)})}{q_t(x_{1:t}, z_{1:t\text{-}1}^{(i)})}$$

Then, N particles/trajectories $z_{1:t}^{(i)}$ are renormalized and resampled in proportion with renormalized weights, with replacement. Figure 2 birefly illustrates the outline of the algorithm. Please see [6] for background, intuition and further details on SMC.

$SMC\text{-}FIVO\big(x_{1:T}, p, q, N\big)$:

$\{w_0^i\}_{i=1}^N = \{1/N\}_{i=1}^N$

for $t \in \{1..T\}$:

for $i \in \{1..N\}$

$z_t^i \sim q_t(z_t | x_{1:t}, z_{1:t\text{-}1}^i)$

$z_{1:t}^i = CONCAT(z_{1:t\text{-}1}^i, z_t^i)$

we

$$\hat{p}_t = \left(\sum_{i=1}^{N} w_{t\text{-}1}^i \alpha(z_{1:t}^i) \right)$$

$\hat{p}_N(x_{1:t}) = \hat{p}_N(x_{1:t\text{-}1})\hat{p}_t$

$\{w_t^i\}_{i=1}^N = \{w_{t\text{-}1}^i \alpha_t(z_{1:t}^i)/\hat{p}_t\}_{i=1}^N$

$\{w_t^i \, z_{1:t}^i\}_{i=1}^N = RESAMP\big(\{w_t^i \, z_{1:t}^i\}_{i=1}^N\big)$

return $\log \hat{p}_N(x_{1:T})$

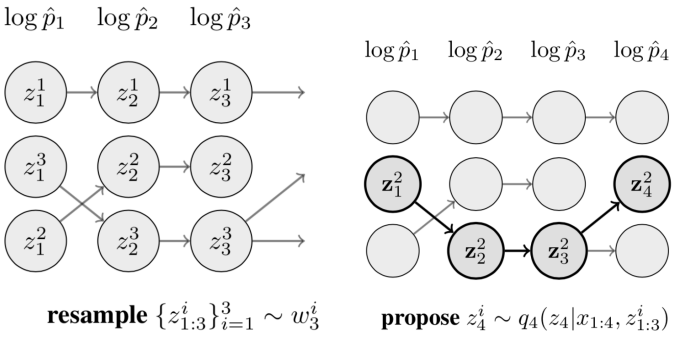


Figure 2: SMC algorithm, also known as a particle filter and illustrations from [1]

Filtering Variational Objectives

As a result of SMC we obtain a useful sample of $\log \hat{p}_N(x_{1:T})$. [1] defines a family of Filtering Variational Objectives (FIVO): $\mathcal{L}_N(x, p) = \mathbb{E}\big[\log \hat{p}_N(x)\big]$. Proposition 1 shows three qualities of FIVO:

$$(Bound)\ \mathcal{L}_N(x,p) \leq \log p(x) \tag{2}$$

$$(Consistency)\ \mathcal{L}_N(x,p) \to \log p(x) \text{ as } N \to \infty \tag{3}$$

$$\begin{gathered}(Asymptotic\ Bias)\ \log p(x) - \mathcal{L}_N(x,p) = \tfrac{1}{2} Var\Big(\frac{\hat{p}(x)}{p(x)}\Big) + O\big(\sqrt{g(N)}\big) \\ g(N) = \mathbb{E}[(\hat{p}_N(x) - p(x))^6]\end{gathered} \tag{4}$$

These properties help argue that FIVO gives a useful variational bound.

Exercise 2: It has been shown in filtering literature as well as in Appendix of [1] that $\hat{p}_N(x_{1:T})$ is an unbiased positive estimator of $p(\boldsymbol{x})$. Using this fact, prove *Bound* part of Proposition 1 above.

As before, we assume p and q are parameterized by $\boldsymbol{\theta}, \boldsymbol{\phi}$ and differentiable with respect to them. With this, FIVO can be optimized analogously to ELBO, using the gradient given by:

$$\begin{aligned}\nabla_{(\boldsymbol{\theta},\boldsymbol{\phi})} \mathcal{L}_N^{FIVO} = \mathbb{E}\Big[& \nabla_{(\boldsymbol{\theta},\boldsymbol{\phi})} \log \hat{p}_N(x_{1:T}) \\ & + \sum_{t=1}^{T} \sum_{i=1}^{N} \log \frac{\hat{p}_N(x_{1:T})}{\hat{p}_N(x_{1:t\text{-}1})} \nabla_{\boldsymbol{\phi}} \log q_{t,\boldsymbol{\phi}}\big(z_t^i | x_{1:t}, z_{1:t\text{-}1}^i\big) \\ & + \sum_{t=1}^{T} \sum_{i=1}^{N} \mathbb{I}_t^{rsmpl} \log \frac{\hat{p}_N(x_{1:T})}{\hat{p}(x_{1:t})} \nabla_{(\boldsymbol{\theta},\boldsymbol{\phi})} \log w_t^i \Big]\end{aligned}$$

The three groups of terms in the gradient above are:

1. gradients of $\log \hat{p}_N(x_{1:T})$ w.r.t. parameters conditional on the latent states
2. gradients of the densities q_t w.r.t. their parameters
3. gradients of the resampling probabilities w.r.t parameters

For the empirical results demonstrated in [1], the authors use reparameterizable qs. This removes the second term from the gradient. q is called reparameterizable when it can be written in terms of a noise variable ϵ with parameter-free density $d(\epsilon)$ and a deterministic function f_ϕ s.t. $z = f_\phi(x, \epsilon)$. The authors also choose to omit the third group of terms that concerns resampling. This makes the overall gradient

biased, but is reported to give an empirical benefit. The overall gradient used for experiments then becomes:

$$\nabla_{(\theta,\phi)}\mathcal{L}_N^{FIVO} = \mathbb{E}\Big[\nabla_{(\theta,\phi)} \log \hat{p}_N(x_{1:T})\Big]$$

Empirical Results for VI+SMC

In the case of using VI in combination with SMC, the focus in on the potential to improve performance on sequential data. [1] report improvements on polyphonic music and speech datasets; [2] study performance on linear gaussian state space models (LGSSM) and video sequences; [3] report results on analytic datasets (LGSSM, multivariate stochastic volatility model) and on real dataset of motor cortex neuron recordings. For parametric models of p, q [3] uses Recurrent Neural Networks (RNNs) while [1], [2] use more recent Variational RNNs [7]. Most experiments report log marginal likelihood, comparing with optimizing original ELBO bound, and with an improved importance-weighted autoencoder (IWAE) bound. These comparisons show significant improvements on some datasets. In the coding/practical part of this tutorial (see attached notebook with instructions and code) we use code and datasets from [1] to examine the practical benefits of using FIVO bounds.

Conclusion

In this tutorial, we examined approaches to combining VI and sequential MC. We took a close look at the line of work from [1] that introduced the family of filtering variational objectives (FIVO): a class of lower bounds on the log marginal likelihood that extend the evidence lower bound commonly used for VI. Conceptually, [1] introduced a view of marginal likelihood estimators as objectives instead of algorithms for inference. These objectives are suited for MLE in latent variable models, and yield tighter VI bounds for some sequential models. Exercise 1 and Exercise 2 of this tutorial gave an opportunity to practice with theoretical components, while further practical exercises utilized code from [1] to examine the practical relevance of VI+SMC combo.

References

[1] Chris J Maddison, John Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Teh. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pages 6576–6586, 2017.

[2] Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential monte carlo. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[3] Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In *International Conference on Artificial Intelligence and Statistics*, pages 968–977, 2018.

[4] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[5] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[6] Thomas B. Schön, Fredrik Lindsten, Johan Dahlin, Johan Wagberg, Christian A. Naesseth, Andreas Svensson, and Liang Dai. Sequential Monte Carlo methods for system identification. *17th IFAC Symposium on System Identification (SYSID)*, 2015.

[7] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988, 2015.