EinstStein VI: General and Integrated Stein Variational Inference in NumPyro
Ahmad Salim Al-Sibahi (UCPH), Ola Rønning (UCPH), Christophe Ley (UGent), Thomas Hamelryck (UCPH)

Introduction and User Interface

EinstStein VI is a NumPyro (Phan 2019) library that integrates the latest developments of Stein VI (Liu and Wang 2016):

- Compositional library supporting various loss functions (ELBO, Renyi ELBO, custom loss, etc.), automatic marginalization of discrete variables (Obermeyer et al. 2019), and deep learning.
- Inference based on ELBO with Stein core (Nalisnick 2017) with support for non-linear optimization (Wang and Liu 2019), a wealth of kernels that include matrix-valued ones (Wang 2019) for graphical models and higher-order optimization.
- Learnable transforms on the parameter space, including triangular transforms (Parno and Marzouk 2018) and neural transforms (Hoffman et al. 2018).

Integrated Stein VI Theory

Bayesian Variational Inference

- Goal: Given prior p(z) over model parameters z and likelihood p(y|x, z) over data x infer posterior: p(z|x) = Z p(z)p(y|x, z)
- Problem: Normalization constant Z = ∫ p(z)p(y|x, z) dz which can be done without explicitly calculating Z.
- Solution: Variational Inference
  - Post: easy to evaluate approximate distribution q(z)
  - Minimize the Kullback-Leibler (KL) divergence DKL(q∥p) which can be done without explicitly calculating Z.

Stein VI: VI (Liu and Wang 2016)

Use particles (z1, z2) as approximation, so q(z) = DKL diverge by inverting Stein forces f(z).

Two Forces of EinstStein VI
- Assumption: We have a negative loss function L(θ) we would like to maximize.
- Example: Evidence Lower Bound (ELBO; Kingma and Welling, 2014).
- Desiderata: We would like to have flexibility to capture multi-modal properties of parameters θ and quantify uncertainty on it.
- Idea: Use ELBO within Stein inference (Nalisnick 2017) which can be factored into an attractive force: S(θ) = Expw(K(θ))v(θ) and a repulsive force: S(θ) = Expw(K(θ))v(θ) which k = RwRw → R is a statistical kernel like RBF.

Matric-Valued Kernels
- Update: Change the attractive force: S(θ) = Expw(K(θ)v(θ)) and repulsive force: S(θ) = Expw(K(θ)v(θ)) which Advantages: Allow pre-conditioning using second-order matrices like the Hessian or Fisher information. Allow factorization of K into local kernels k(θi)j based on independent forces given by a graphical model.

Non-linear Stein

Implementation Challenges

Algorithm

Input: Classical parameters φ and φ′, Stein parameters {θi}, model pθ,h(x, z, y), guide qφ,φ′(z, loss), kernel inference KL param
Output: Parameter changes based on classical VI (Δφ, Δφ′) and Stein VI forces (ωi, Δωi).

Problem: Stein VI is more flexible to capture multi-modal properties of parameters θ.

Result: Stein VI can be seen to perform well in the above figure!

Experiment and Results

Double Moon Model
- Problem: Multi-modal distribution which can be hard to fit using traditional VI.
- Result: EinstStein VI with neural transform (3 autoregressive flows of hidden dimensions (2,2,2)) works well.

Stein Mixture Latent Dirichlet Allocation (SM-LDA)
- Goal: Infer topics θ from document represented as bags of words w. Each document is generated according to the following process:
  - D ~ Dir(α)
  - zj ~ Cat(θ) wj~ Cat(φjε) n ∈ {1...N}
- Problem: Discrete latent variable z makes model non-differentiable.
- Implementation: MLP (100 hidden dimensions), 20 topics, 5 Stein particles
- Result: EinstStein VI works well with automatic marginalization provided by NumPyro (Obermeyer et al. 2019).

Stein Mixture Deep Markov Model (SMD-MM)

References