[Paper review 26]

Variational Inference with Normalizing Flows

(Danilo Jimenez Rezende, Shakir Mohamed, 2016)

[Contents]

1. Abstract

choice of approximate posterior distribution q in VI :

- had been simple families

  (ex. mean-filed or other simple structured approximations)
- these restrictions $\rightarrow$ not good performance

Introduce a new approach, "Normalizing Flow"

- flexible, complex, and scalable

2. Introduction

limitations of variational methods : choice of posterior approximation are often limited

$\rightarrow$ thus, richer approximation is needed

Methods for richer approximation

- ex1) structured mean field approximations that incorporate basic form of dependency within the approximate posterior
- ex2) mixture model (limit : potential scalability... have to compute each for the mixture component)

We will

- 1) review the current est practice (based on "amortized VI ")
- 2) make following contributions
  - a) propose a method using normalizing flow (NF)
  - b) show that NF admit infinitesimal flows

3. Amortized Variational Inference

current best practice in VI uses...

- 1) mini-batches
- 2) stochastic gradient descent (SGD)

$\rightarrow$ to deal with very large dataset

for successful variational approach, we need to...

- 1) efficient computation of the derivatives of the expected log-likelihood, $\nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p_\theta(\mathbf{x} \mid \mathbf{z})]$

  $\rightarrow$ solution 1) MC estimation

  $\rightarrow$ solution 2) inference networks

  (solution 1+2 = "amortized VI")

- 2) choosing the richest, computationally-feasible approximate posterior distribution, $q(\cdot)$

  $\rightarrow$ solution) Normalizing Flow!

3.1 Stochastic Backpropagation

compute $\nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p_\theta(\mathbf{x} \mid \mathbf{z})]$ (expected log likelihood) ... with MC estimation!

also called "doubly-stochastic estimation".. why double?

- 1) stochasticity from the mini-batch
- 2) stochasticity from the MC approximation of the expectation

"continuous latent variables" + "MC approximation"

= Stochastic Gradient Variational Bayes (SGVB)

SGVB involves 2 steps

- 1) Reparameterization

  $z \sim \mathcal{N}\left(z \mid \mu, \sigma^2\right) \Leftrightarrow z=\mu+\sigma \epsilon, \quad \epsilon \sim \mathcal{N}(0,1)$
- 2) Backprop with MC

  $\nabla_\phi \mathbb{E}_{q_\phi(z)}\left[f_\theta(z)\right] \Leftrightarrow \mathbb{E}_{\mathcal{N}(\epsilon \mid 0,1)}\left[\nabla_\phi f_\theta(\mu+\sigma \epsilon)\right]$

3.2 Inference Networks

Inference Network

- def) model that learns an INVERSE MAP from observation(x) to latent variables(z)
- $q_\phi(\cdot)$ is represented using Inference Networks!
- why Inference Network?
  - we avoid the need to compute per data point variational parameters, but can instead compute
    a set of global variational parameters ϕ valid for inference at both training and test time.
- simplest Inference Network : "DIAGONAL GAUSSIAN densities"

  $q_\phi(\mathbf{z} \mid \mathbf{x})=\mathcal{N}\left(\mathbf{z} \mid \boldsymbol{\mu}_\phi(\mathbf{x}), \operatorname{diag}\left(\boldsymbol{\sigma}_\phi^2(\mathbf{x})\right)\right)$

3.3 Deep Latent Gaussian Models (DLGM)

hierarchy of L layers of Gaussian latent variables z_l for layer l

$p\left(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_L\right)=p\left(\mathbf{x} \mid f_0\left(\mathbf{z}_1\right)\right) \prod_{l=1}^L p\left(\mathbf{z}_l \mid f_l\left(\mathbf{z}_{l+1}\right)\right)$

- prior over latent variables : $p\left(\mathbf{z}_l\right)=\mathcal{N}(\mathbf{0}, \mathbf{I})$
- observation likelihood : $p_\theta(\mathrm{x} \mid \mathrm{z})$ by NN

DLGMs

- use continuous latent variable
- model class perfectly suited to fast amortized VI (using ELBO & stochastic back-prop)
- end-to-end system of DLGM $\approx$ encoder-decoder architecture

4. Normalizing Flows (NF)

optimal variational distribution

- $\mathbb{D}_{\mathrm{KL}}[q \| p]=0$

  (=$q_\phi(\mathbf{z} \mid \mathbf{x})=p_\theta(\mathbf{z} \mid \mathbf{x})$)
- $q_\phi(\mathbf{z} \mid \mathbf{x})$ should be highly flexible

NF descrribes the transformation of probability density through "A SEQUENCE OF INVERTIBLE MAPPINGS"

4.1 Finite Flows

setting

- $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$, where $f^{-1} = g$
- $g \circ f(\mathbf{z}) = \mathbf{z}.$
- $\mathbf{z}' = f(\mathbf{z})$

variable transformation

-$q\left(\mathbf{z}'\right) = q(\mathbf{z})\left|\det \frac{\partial f^{-1}}{\partial \mathbf{z}'}\right| = q(\mathbf{z})\left|\det \frac{\partial f}{\partial \mathbf{z}}\right|^{-1}$

successive application

$\mathbf{z}_K = f_K \circ \ldots \circ f_2 \circ f_1\left(\mathbf{z}_0\right)$

$\ln q_K\left(\mathbf{z}_K\right) = \ln q_0\left(\mathbf{z}_0\right) - \sum_{k=1}^K \ln\left|\det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}}\right|$

expectation

$\mathbb{E}_{q_K}[h(\mathbf{z})] = \mathbb{E}_{q_0}\left[h\left(f_K \circ f_{K-1} \circ \ldots \circ f_1\left(\mathbf{z}_0\right)\right)\right]$

- does not depend on q_k

summary

- use simple factorized distribution (ex. independent Gaussian)
- apply NF of different lengths to get increasingly complex distribution

5. Inference with NFs

we must ...

- 1) specify a class of invertible transformations
- 2) efficient mechansim for computing the determinant of Jacobian

Therefore we require NF that allow for low-cost computation of the determinant, or where Jacobian is not needed!

5.1 Invertible Linear-time Transformations

linear time transformation

= we can compute the log det-Jacobian term in $O(D)$ time

5.1.1 Planar Flows

form : $f(\mathbf{z})=\mathbf{z}+\mathbf{u} h\left(\mathbf{w}^{\top} \mathbf{z}+b\right)$

- $\lambda=\left\{\mathbf{w} \in \mathbb{R}^{D}, \mathbf{u} \in \mathbb{R}^{D}, b \in \mathbb{R}\right\}$
- $h(\cdot)$: smooth element-wise non-line with derivative $h^{\prime}(\cdot)$
- $\left|\det \frac{\partial f}{\partial \mathbf{Z}}\right|=\left|\det\left(\mathbf{I}+\mathbf{u} \psi(\mathbf{z})^{\top}\right)\right|=\left|1+\mathbf{u}^{\top} \psi(\mathbf{z})\right|$

  (where $\psi(\mathbf{z})=h^{\prime}\left(\mathbf{w}^{\top} \mathbf{z}+b\right) \mathbf{w}$)

$\mathbf{z}_{K}=f_{K} \circ \ldots \circ f_{2} \circ f_{1}\left(\mathbf{z}_{0}\right)$

- before) $\ln q_{K}\left(\mathbf{z}_{K}\right)=\ln q_{0}\left(\mathbf{z}_{0}\right)-\sum_{k=1}^{K} \ln \left|\det \frac{\partial f_{k}}{\partial \mathbf{z}_{k-1}}\right|$
- after) $\ln q_{K}\left(\mathbf{z}_{K}\right)=\ln q_{0}(\mathbf{z})-\sum_{k=1}^{K} \ln \left|1+\mathbf{u}_{k}^{\top} \psi_{k}\left(\mathbf{z}_{k-1}\right)\right|$

5.1.2 Radial Flows

form : $f(\mathbf{z})=\mathbf{z}+\beta h(\alpha, r)\left(\mathbf{z}-\mathbf{z}_{0}\right)$

- $\left|\det \frac{\partial f}{\partial \mathbf{z}}\right|=[1+\beta h(\alpha, r)]^{d-1}\left[1+\beta h(\alpha, r)+\beta h^{\prime}(\alpha, r) r)\right]$

under certain conditions...

5.1.1) Planar flows and 5.1.2) Radial Flows can be invertible!

5.2 Flow-Based Free Energy Bound

approximate our posterior distribution, with a flow of length K

$q_{\phi}(\mathbf{z} \mid \mathbf{x}):=q_{K}\left(\mathbf{z}_{K}\right)$

$$
\begin{aligned}
\mathcal{F}(\mathrm{x}) &=\mathbb{E}_{q_{\phi}(z \mid x)}\left[\log q_{\phi}(\mathrm{z} \mid \mathrm{x})-\log p(\mathrm{x}, \mathrm{z})\right] \\
&=\mathbb{E}_{q_{0}\left(z_{0}\right)}\left[\ln q_{K}\left(\mathbf{z}_{K}\right)-\log p\left(\mathrm{x}, \mathbf{z}_{K}\right)\right] \\
&=\mathbb{E}_{q_{0}\left(z_{0}\right)}\left[\ln q_{0}\left(\mathbf{z}_{0}\right)\right]-\mathbb{E}_{q_{0}\left(z_{0}\right)}\left[\log p\left(\mathbf{x}, \mathbf{z}_{K}\right)\right]-\mathbb{E}_{q_{0}\left(z_{0}\right)}\left[\sum_{k=1}^{K} \ln \mid 1+\mathbf{u}_{k}^{\top} \psi_{k}\left(\mathbf{z}_{k-1}\right)\right]
\end{aligned}
$$

- do not need $q_{k}(\cdot)$, only need $q_{0}(\cdot)$

5.3 Algorithm Summary

Algorithm 1 Variational Inf. with Normalizing Flows

Parameters: ϕ variational, $\boldsymbol{\theta}$ generative

while not converged **do**

$\mathbf{x} \leftarrow \{\text{Get mini-batch}\}$

$\mathbf{z}_0 \sim q_0(\bullet|\mathbf{x})$

$\mathbf{z}_K \leftarrow f_K \circ f_{K-1} \circ \ldots \circ f_1(\mathbf{z}_0)$

$\mathcal{F}(\mathbf{x}) \approx \mathcal{F}(\mathbf{x}, \mathbf{z}_K)$

$\Delta\boldsymbol{\theta} \propto -\nabla_{\boldsymbol{\theta}}\mathcal{F}(\mathbf{x})$

$\Delta\boldsymbol{\phi} \propto -\nabla_{\boldsymbol{\phi}}\mathcal{F}(\mathbf{x})$

end while