MCMC and Variational Inference for AutoEncoders

Achille Thin [1], Alain Durmus [2], Eric Moulines [1]

[1] Ecole Polytechnique, [2] ENS Paris-Saclay

September 9, 2020

Problem

Generative modelling objective

- Objective: Learn and sample from a model of the true underlying data distribution p^* given a dataset $\{x_1, \ldots, x_n\}$ where $x_i \in \mathbb{R}^P$, with $P \gg 1$.
- Two-steps
  - Specify a class of model $\{p_\theta \, , \, \theta \in \Theta\}$.
  - Find the best $\hat{\theta}^n$ by maximizing the likelihood

$$\hat{\theta}^n = \arg\max_\theta \sum_{i=1}^n \log p_\theta(x_i) \; .$$

Latent variable modelling

- Autoencoders assume the existence of a latent variable whose dimension D is much smaller than the dimension of the observation P.
- Attached to the latent variable $z \in \mathbb{R}^D$ is a prior distribution π from which we can sample from.
- The specification of the model is completed by specifying the conditional distribution of the observation x given the latent variable z:

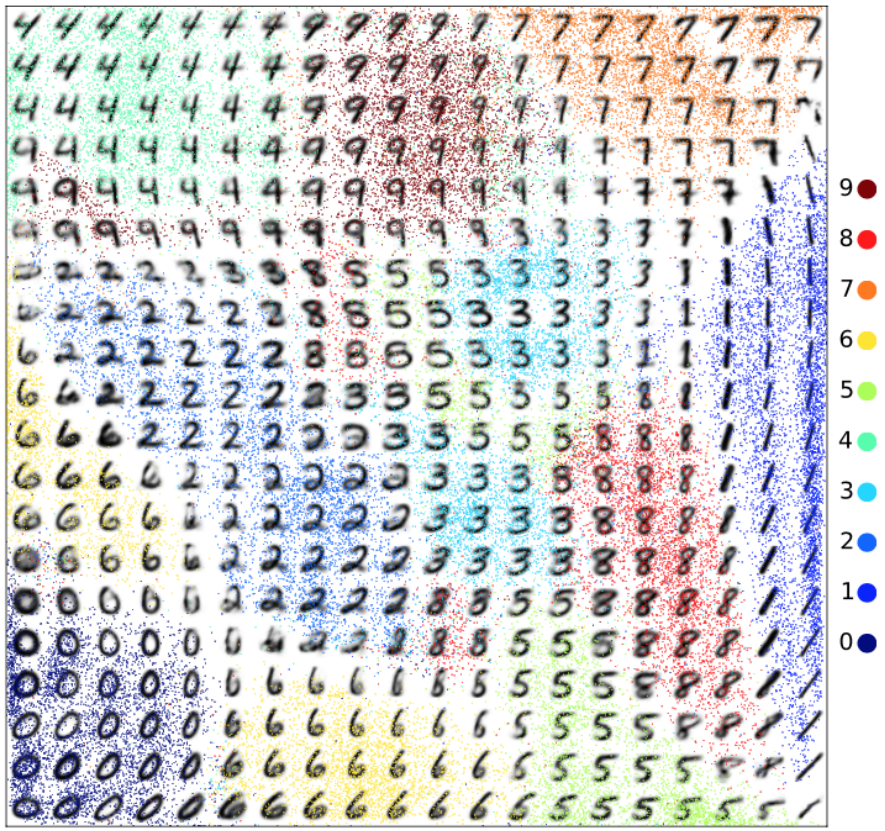
$$x \mid z \sim p_\theta(x \mid z)$$

- The marginal likelihood of the observations is obtained by computing first the joint distribution of the observation and the latent variable $p_\theta(x, z) = p_\theta(x \mid z)\pi(z)$ and then marginalizing w.r.t. the latent variable z:

$$p_\theta(x) = \int p_\theta(x \mid z)\pi(z)\mathrm{d}z \ .$$

Data Generation with Latent variables

- Draw latent variable $z \sim \pi$.
- Draw observation $x \mid z \sim p_\theta(x \mid z)$.
- Each region in the latent space is associated to a particular form of observation.



Optimisation of the model

- Estimation Perform maximum likelihood estimation with stochastic gradient techniques.
- Obtain unbiased estimators of the gradient of

$$p_\theta(x) = \int p_\theta(x \mid z)\pi(z)\mathrm{d}z \ .$$

- Usually untractable !!

Fisher's Identity

- Idea: take advantage of Fisher's identity:

$$\begin{aligned}\nabla_\theta \log p_\theta(x) &= \int \frac{\nabla_\theta p_\theta(x,z)}{p_\theta(x)} \mathrm{d}z \\ &= \int \nabla_\theta \log p_\theta(x,z) \frac{p_\theta(x,z)}{p_\theta(x)} \mathrm{d}z \\ &= \int \nabla_\theta \log p_\theta(x,z) p_\theta(z \mid x) \mathrm{d}z \ .\end{aligned}$$

- Gradient of incomplete likelihood of the observations is computed using the complete likelihood (which is tractable !)
- However, we need to sample from the posterior $p_\theta(z \mid x)$.

Markov Chain Monte Carlo

- Idea: Build an ergodic Markov chain whose invariant distribution is the target, known up to a normalization constant: $p_\theta(z \mid x) \propto \pi(z) p_\theta(x|z)$.
- Metropolis Hastings (MH) algorithms is an option
  - Draw a proposal z' from $q_\theta(z' \mid z, x)$
  - Accept / Reject the proposal with probability

$$\alpha_\theta(z, z') = 1 \wedge \frac{p_\theta(z'|x) q_\theta(z|z', x)}{p_\theta(z|x) q_\theta(z'|z, x)} .$$

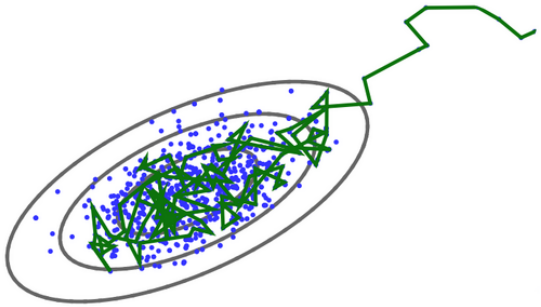


Figure: Markov chain targetting a correlated Gaussian distribution

Markov Chain Monte Carlo

- Many recent advances for efficient MCMC methods, using Langevin dynamics, Hamiltonian Monte Carlo.
- Pros: provide a theoretically sound framework to sample from

$$p_\theta(z \mid x) \propto p_\theta(x \mid z)\pi(z)$$

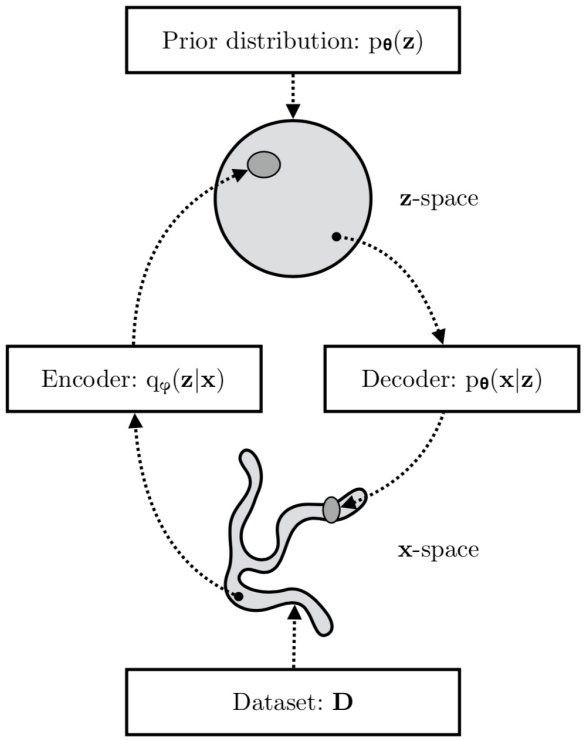
  (known up to a constant).
- Cons:
  - mixing times in high dimensions.
  - convergence assessment.
  - multimodality (metastability).
- But Cons do not always outweights the Pros, see [HM19]

Variational Inference

- Idea: Introduce a parametric family of probability distributions

$$\mathcal{Q} = \{q_\phi\,,\ \phi \in \Phi\}\ .$$

- Goal minimize a divergence between q_ϕ and the untractable posterior $p_\theta(\cdot \mid x)$.
- For each observation x: different target posterior $p_\theta(z \mid x)$.
- Idea: use amortized Variational Inference: $x \mapsto q_\phi(z \mid x)$.



Variational Inference

- Evidence Lower BOund (ELBO)

$$\begin{aligned}\mathrm{ELBO}(\theta, \phi; x) &= \int \log \left(\frac{p_\theta(x, z)}{q_\phi(z \mid x)} \right) q_\phi(z \mid x) \mathrm{d}z \\ &= \int \log \left(\frac{p_\theta(z \mid x) p_\theta(x)}{q_\phi(z \mid x)} \right) q_\phi(z \mid x) \\ &= \log p_\theta(x) - \mathrm{KL}(q_\phi(z \mid x) \| p_\theta(z \mid x)) \leq \log p_\theta(x) \ .\end{aligned}$$

- The ELBO is a lower bound of the incomplete data likelihood also referred to as the evidence.
  - the bound is tight if $\mathcal{Q}$ contains the true posterior $p_\theta(\cdot \mid x)$.
- The KL divergence measures the discrepancy when approximating the posterior with the variational distribution.
  - Can be replaced by f-divergence.
- The ELBO is tractable and can be easily optimized using the reparameterization trick, crucial for stochastic gradient descent.

Variational Auto Encoder

The Variational Auto Encoder (VAE) builds on the representational power of (Deep) Neural Networks to implement a very flexible class of encoders $q_\phi(z \mid x)$ and decoders $p_\theta(z \mid x)$.

- The encoder q_ϕ is parameterized by a deep neural network, which takes as input the observation x and outputs parameters for the distribution $q_\phi(\cdot \mid x)$.
- The decoder $p_\theta(z \mid x)$ is built symmetrically as a neural network which takes as input a latent variable z and outputs the parameters of the distribution $p_\theta(x \mid z)$.

"Classical" implementation

- In most examples, the dimension P of the observation x is large.
- The dimension of the latent space D is typically much smaller.
- The distribution of the latent variable denoted π is Gaussian.
- ... More sophisticated proposals can be considere: Gaussian mixture or hierarchical priors.
- In the vanilla implementation the variational distribution $q_\phi(\cdot \mid x)$ is

$$q_\phi(z \mid x) = \mathrm{N}(z; \mu_\phi(x), \sigma_\phi(x)\,\mathrm{Id})$$

where $\mu_\phi(x), \sigma_\phi(x)$ are the output of a neural network taking the observation x as input. This parameterization is often referred to as the mean field approximation.

Reparameterization trick

Optimization w.r.t. θ, ϕ of

$$\mathrm{ELBO}(\theta, \phi; x) = \int \log \left(\frac{p_\theta(x, z)}{q_\phi(z \mid x)} \right) q_\phi(z \mid x) \mathrm{d}z \ .$$

- The gradient of the function

$$\phi \mapsto \int h(x, z) q_\phi(z \mid x) \mathrm{d}z$$

  may be written as

$$\int h(x, z) \nabla \log q_\phi(z|x) q_\phi(z|x) \mathrm{d}z \ ,$$

- Monte Carlo estimation

$$M^{-1} \sum_{i=1}^{M} h(x, Z_i) \nabla \log q_\phi(Z_i \mid x) \ , \quad Z_i \sim q_\phi(\cdot \mid x) \ .$$

- Problem: the variance of the vanilla unbiased estimator of this quantity is generally very high !

Reparameterization trick

- Reparameterization trick Assume there exists a diffeomorphism $V_{\phi,x}$ and a distribution g easy to sample from such that

$$\epsilon \sim \mathrm{g}\,, \quad z = V_{\phi,x}(\epsilon) \sim q_\phi(\cdot \mid x)\,.$$

- Using the reparameterization, the ELBO writes

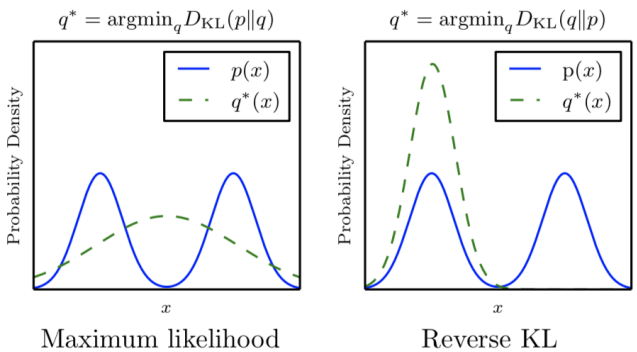
$$\mathrm{ELBO}(\theta, \phi; x) = \int \log\left(\frac{p_\theta(x, V_{\phi,x}(\epsilon))}{q_\phi(V_{\phi,x}(\epsilon) \mid x)}\right) \mathrm{g}(\epsilon)\mathrm{d}\epsilon\,.$$

- Gradient is computed using the chain rule.

Limitations of the VAE

The vanilla VAE suffers from some well known limitations.

- The mean-field approximation is usually believed to be too simple.
- Leads to overfitting or mode dropping (reverse KL used in Variational Inference).



- Moreover, we can re write the ELBO as

$$\text{ELBO}(\theta, \phi; x) = \mathbb{E}_{q_\phi(\cdot|x)} \left[\log p_\theta(x \mid z)\right] - \text{KL}\left(q_\phi(\cdot \mid x)||\pi\right)$$

This can lead to an uninformed posterior approximation. Introduction of β-VAE and Ladder Variational Autoencoders [HMP+16, SRM+16].

Enriching the variational approximation

- To address the first issue presented, [RM15] suggests to improve the variational mean-field using parameterized diffeomorphisms which increase the flexibility of the distribution.
- Those diffeomorphisms are referred to as **Normalizing Flows**.
- Thanks to the recent advances in MCMC methods, flows [CDS18] and other MCMC inspired methods come to enrich the variational distribution [SKW15, Hof17].
- However, none of these approaches thoroughly combine MCMC and Metropolis Hastings methods with Variational inference.

MetFlow variational family

Our objective: construct a family of variational distributions, based on the K-th marginal of a Markov chain with the following properties:

- The chain is initialized with the amortized variational mean-field approximation, whose density is denoted m_ϕ^0.
- The Markov chain has the true posterior $p_\theta(z \mid x)$ as invariant distribution.
- The Markov kernel depend on learnable parameters also denoted ϕ which can be adjusted.

We specify a framework in which the parameters of the Markov kernel and the initial distribution are all learnable.

Metropolis Hastings kernel

Denote by π the target distribution dependence in x and θ is implicit.

- innovation noise: $(U_k)_{k\in\mathbb{N}}$ an i.i.d. sequence of random vectors in $\mathbb{R}^{D_u}$, with density h.
- proposal mapping $T\colon \mathbb{R}^D \times \mathbb{R}^{D_u} \to \mathbb{R}^D$.
- Algorithm:
  - Propose a move $Y_{k+1} = T(Z_k, U_{k+1}) = T_{U_{k+1}}(Z_k)$.
  - accept the move $X_{k+1} = Y_{k+1}$ with probability $\alpha_{U_{k+1}}(Z_k)$.
  - Otherwise, set $X_{k+1} = X_k$.

Metropolis-Hastings kernel

- Q_u: the Markov kernel conditional to the innovation noise as

$$Q_u(z, \mathsf{A}) = \alpha_u(z)\delta_{T_u(z)}(\mathsf{A}) + \{1 - \alpha_u(z)\}\delta_z(\mathsf{A}) \ .$$

- The Metropolis-Hastings kernel M_h is obtained by marginalizing w.r.t. to the distribution of the innovation:

$$M_h(z, \mathsf{A}) = \int Q_u(z, \mathsf{A})h(u)\mathrm{d}u \ .$$

- The acceptance function α_u is chosen to satisfy the reversibility condition

$$\pi(\mathrm{d}z)M_h(z, \mathrm{d}z') = \pi(\mathrm{d}z')M_h(z', \mathrm{d}z) \ .$$

Random Walk Metropolis

- Here $D_u = D$, $h = \mathrm{N}(z; 0, \Sigma)$.
- Draw innovation $U_k \sim h$.
- Propose a point

$$Y_{k+1} = T_{U_k}^{\mathrm{RWM}}(Z_k) = Z_k + U_k \ .$$

- Accept with probability

$$\alpha_u^{\mathrm{RWM}}(z) = 1 \wedge \left(\pi(T_u^{\mathrm{RWM}}(z))/\pi(z)\right) \ .$$

- Very simple and straightforward.... Slow mixing in high dimensions.

Metropolis Adjusted Langevin Algorithm

- Idea: Inform MH proposal mapping with target distribution.
- Here $D_u = D$, $h = \mathrm{N}(z; 0, \mathrm{Id})$. Assume that $z \mapsto \log \pi(z)$ is differentiable and denote by $\nabla \log \pi(z)$ its gradient. At each step k,
  - Draw innovation $U_k \sim h$.
  - Propose

$$Y_{k+1} = T_{U_k}^{\mathrm{MALA}}(Z_k) = Z_k + \Sigma \nabla \log \pi(Z_k) + \sqrt{2}\Sigma^{1/2} U_k \ .$$

  - Accept with probability

$$\alpha_u^{\mathrm{MALA}}(z) = 1 \wedge \frac{\pi\big(T_u^{\mathrm{MALA}}(z)\big) g\big(T_u^{\mathrm{MALA}}(z), z\big)}{\pi(z) g\big(z, T_u^{\mathrm{MALA}}(z)\big)} \ ,$$

  where $g(z_1, z_2) = \mathrm{N}(z_2; T_0^{\mathrm{MALA}}(z_1), \Sigma)$ is the proposal kernel density.
- Mixing time is faster than RWM, but still the proposed moves are local

Hamiltonian Monte Carlo I

- Currently viewed as the state of the art MCMC algorithm.
- Uses a Data Augmentation approach: artificially extends the state space by adding a momentum variable. The extended target density is

$$\pi(z) = \pi_q(q)\mathrm{N}(p; 0, \mathrm{Id}_S) \ ,$$

  where π_q is the distribution of interest over the position q.
- The marginal distribution is $\int \pi(p, q)\mathrm{d}p = \pi_q(q)$...
  - it therefore suffices to sample the joint distribution and to discard the momentum variable.

Hamiltonian system

- The extended target

$$\pi(p, q) \propto \exp(-H(p, q))$$

where $H(p, q)$ is the Hamiltonian is the sum of the potential energy and kinetic energy:

$$H(p, q) = U(q) + K(p)\,, \quad U(q) = -\log \pi_q(q)\,, \quad K(p) = (1/2)|p|^2$$

- Hamiltonian equations :

$$\dot{q} = \nabla_p H(p, q) = p \quad \dot{p} = -\nabla_q H(p, q) = -\nabla_q U(q)\,.$$

- Hamilton's equations can be easily shown to be equivalent to Newton's equations.
- Because a system described by conservative forces conserves the total energy, the Hamilton's equations conserve the total Hamiltonian.

Leapfrog steps

- When an exact analytic solution of the Hamilton dynamics is available, we can use the proposed flow.
- however, there is no analytic solution for Hamilton's equations, and therefore, Hamilton's equations must be approximated by discretizing time.
- The leapfrog discretization integration, also called the Stormer-Verlet method, provides a good approximation for Hamiltonian dynamics: $\mathsf{LF}_\gamma(q_0, p_0) = (q_1, p_1)$ with

$$p_{1/2} = p_0 - \gamma/2 \nabla U(q_0)\,, \quad q_1 = q_0 + \gamma p_{1/2}\,, \quad p_1 = p_{1/2} - \gamma/2 \nabla U(q_1)\,.$$

Partial refresh

- Define the mappings, for the *partial refresh coefficient* $\kappa \in (0,1)$:

$$T_\gamma^{\mathsf{LF}} : (q,p) \to \mathsf{LF}_{\gamma,N}(q,-p) \ ,$$

$$\text{and} \quad T_{\kappa,u}^{\mathsf{ref}}(q,p) : (q,p) \to (q, \kappa p + \sqrt{1-\kappa^2}u) \ , \ u \in \mathbb{R}^P \ ,$$

where $\mathsf{LF}_{\gamma,N}$ is the N-th time composition of LF_γ.

Hamiltonian Monte Carlo II

- Set $D_u = P$, $h = \mathrm{N}(z; 0, \mathrm{Id}_P)$.
- Draw innovation $U_k \sim h$.
- Propose point

$$Y_{k+1} = T_\gamma^{\mathsf{LF}} \circ T_{\kappa, U_k}^{\mathsf{ref}}(Z_k) \ .$$

- Accept with probability

$$\alpha_u(q,p) = 1 \wedge \left[\pi\big(T_u(q,p)\big)/\pi(q,p)\right] \ .$$

- This is not a "classical" MH algorithm yet the resulting kernel is reversible w.r.t. π, see [Nea11, Section 3.2] and [BRJM18, Section 6].
- Proposals can be far from current points thanks to LF.

MetFlow variational family

- Let $M_{\phi,h}$ be a parameterized MH kernel and associated proposal mappings $T_{\phi,u}$, innovation noise density h and acceptance functions $\alpha_{\phi,u}$.
- Define the `MetFlow` variational family

$$\mathcal{Q} := \{\xi_\phi^K = \xi_\phi^0 M_{\phi,h}^K : \phi \in \Phi\} .$$

- $M_{\phi,h}^K$ is the K iterate of $M_{\phi,h}$ and thus ξ_ϕ^K is the distribution of the K-th iterate Z_K of the Markov chain $(Z_k)_{k\in\mathbb{N}}$ with $Z_0 \sim \xi_\phi^0$.
- Idea: Express the marginal distribution of the Markov chain after K iterations.

Flavour of the proof

To give an idea, we show here the expression after only 1 iteration. For a $\mathrm{C}^1(\mathbb{R}^D, \mathbb{R}^D)$ diffeomorphism ψ, define by $J_\psi(z)$ the absolute value of the Jacobian determinant at $z \in \mathbb{R}^D$.

Lemma

Let $(u, \phi) \in \mathbb{R}^{D_u} \times \Phi$. Assume that ξ_ϕ^0 admits a density m_ϕ^0 w.r.t. the Lebesgue measure. Assume in addition $T_{\phi,u}$ is a C^1 diffeomorphism. Then, the distribution $\xi_\phi^1(\cdot|u) = \int_{\mathbb{R}^d} m_\phi^0(z_0) Q_{\phi,u}(z_0, \cdot) \mathrm{d}z_0$ has a density w.r.t. the Lebesgue measure given by

$$m_\phi^1(z|u) = \alpha_{\phi,u}^1\left(T_{\phi,u}^{-1}(z)\right) m_\phi^0\left(T_{\phi,u}^{-1}(z)\right) J_{T_{\phi,u}^{-1}}(z) + \alpha_{\phi,u}^0(z) m_\phi^0(z) \ ,$$

with

$$\alpha_{\phi,u}^1(z) = \alpha_{\phi,u}(z) \quad \textit{and} \quad \alpha_{\phi,u}^0(z) = 1 - \alpha_{\phi,u}(z) \ .$$

The distribution ξ_ϕ^1 has a density given by

$$m_\phi^1(z) = \int m_\phi^1(z|u) h(u) \mu_{\mathbb{R}}^{D_u}(\mathrm{d}u) \ .$$

Flavour of the proof

Proof.

Idea: Change of variable $z_1 = T_{\phi,u}(z_0)$:

$$\int f(z) m^0_\phi(z_0) Q_{\phi,u}(z_0, \mathrm{d}z) =$$

$$\int \Big[m^0_\phi(z_0) \big\{ \alpha^1_{\phi,u}(z_0) f\big(T_{\phi,u}(z_0)\big) + \alpha^0_{\phi,u}(z_0) f(z_0) \big\} \Big] \mathrm{d}z_0$$

$$= \int \Big[\big\{ \alpha^1_{\phi,u}\big(T^{-1}_{\phi,u}(z_1)\big) m^0_\phi(T^{-1}_{\phi,u}(z_1)) J_{T_u^{-1}}(z_1) + \alpha^0_{\phi,u}(z_1) m^0_\phi(z_1) \big\} f(z_1) \Big] \mathrm{d}z_1 \ .$$

□

- Different flows depending on the results of the accept/reject steps: the final distribution is a mixture of the push-forward distributions
- Increased complexity and ability to recover different modes (while keeping invariance of MCMC kernels guarantee that we do "better" each time)

Main Result

Define, for a family $\{T_i\}_{i=1}^K$ of mappings on $\mathbb{R}^D$ and $1 \leq i \leq k < K$, $\bigcirc_{j=i}^k T_j = T_i \circ \cdots \circ T_k$, for a family of vectors $\mathbf{v}_K = (v_1, \ldots, v_K)$. Set $h(\mathbf{u}_K) = \prod_{i=1}^K h(u_i)$. By convention, $T^0 = \mathrm{Id}$.

Proposition

Assume that for any $(u, \phi) \in \mathbb{R}^{D_u} \times \mathbf{\Phi}$, $T_{\phi,u}$ is a C^1 diffeomorphism and ξ_ϕ^0 admits a density m_ϕ^0 w.r.t. the Lebesgue measure. For any $\{u_i \in \mathbb{R}^{D_u}\}_{i=1}^K$ and $\phi \in \mathbf{\Phi}$, $\xi_\phi^K(\mathrm{d}z \mid \mathbf{u}_K) = \xi_\phi^0 Q_{\phi,u_1} \cdots Q_{\phi,u_K}(\mathrm{d}z)$ has a density given by

$$m_\phi^K(z|\mathbf{u}_K) = \sum_{\mathbf{a}_K \in \{0,1\}^K} m_\phi^K(z, \mathbf{a}_K|\mathbf{u}_K) \ ,$$

where

$$m_\phi^K(z, \mathbf{a}_K|\mathbf{u}_K) = m_\phi^0\left(\bigcirc_{j=1}^K T_{\phi,u_j}^{-a_j}(z)\right) J_{\bigcirc_{j=1}^K T_{\phi,u_j}^{-a_j}}(z) \prod_{i=1}^K \alpha_{\phi,u_i}^{a_i}\left(\bigcirc_{j=i}^K T_{\phi,u_j}^{-a_j}(z)\right)$$

In particular,

$$m_\phi^K(z) = \int m_\phi^K(z \mid \mathbf{u}_K) h(\mathbf{u}_K) \mathrm{d}\mathbf{u}_K \ .$$

A New ELBO

- Objective optimize the ELBO

$$\mathrm{ELBO}(\theta, \phi; x) = \int \log \left(\frac{p_\theta(x, z)}{m^K_{\theta,\phi}(z \mid x)} \right) m^K_\phi(z \mid x) \mathrm{d}z \ .$$

- Note that $m^K_{\theta,\phi}$ now also depends on θ as MCMC targets $p_\theta(\cdot \mid x)$.
- Problem: The distribution $m^K_{\theta,\phi}$ is untractable (a mixture of 2^K components) !!
- Idea: Define a new ELBO

$$\mathcal{L}(\theta, \phi; x) = \sum_{\mathbf{a}_K \in \{0,1\}^K} \int h(\mathbf{u}_K) m^K_{\theta,\phi}(z_K, \mathbf{a}_K | \mathbf{u}_K, x) s_{\theta,\phi}(x, z_K, \mathbf{a}_K, \mathbf{u}_K) \mathrm{d}z_K \mathrm{d}\mathbf{u}_K \ ,$$

where

$$s_{\theta,\phi}(x, z_K, \mathbf{a}_K, \mathbf{u}_K) = \log \left(2^{-K} p_\theta(x, z_K) / m^K_{\theta,\phi}(z_K, \mathbf{a}_K | \mathbf{u}_K, x) \right) \ .$$

A new ELBO

This is a proper evidence lower bound !! Jensen's inequality w.r.t. $m^K_{\theta,\phi}(z_K, \mathbf{a}_K|\mathbf{u}_K, x)$ indeed shows:

$$\sum_{\mathbf{a}_K \in \{0,1\}^K} \int m^K_{\theta,\phi}(z_K, \mathbf{a}_K|\mathbf{u}_K, x) \log \left(\frac{2^{-K} p_\theta(x, z_K)}{m^K_{\theta,\phi}(z_K, \mathbf{a}_K|\mathbf{u}_K, x)} \right) \mathrm{d}z_K \leq \log p_\theta(x) \ .$$

Further investigating the lower bound

- Define

$$m^K_{\theta,\phi}(z_K, \mathbf{a}_K, \mathbf{u}_K|x) = h(\mathbf{u}_K) m^K_{\theta,\phi}(z_K, \mathbf{a}_K|\mathbf{u}_K, x) \ ,$$
$$m^K_{\theta,\phi}(\mathbf{a}_K, \mathbf{u}_K|z_K, x) = m^K_{\theta,\phi}(z_K, \mathbf{a}_K, \mathbf{u}_K|x) / m^K_{\theta,\phi}(z_K|x) \ .$$

- Jensen's inequality w.r.t. $m^K_{\theta,\phi}(\mathbf{u}_K, \mathbf{a}_K|z_K, x)$

$$\mathcal{L}(\theta,\phi) = \sum_{\mathbf{a}_K \in \{0,1\}^K} \int m^K_{\theta,\phi}(z_K, \mathbf{a}_K, \mathbf{u}_K|x) \log \left(\frac{2^{-K} p_\theta(x, z_K)}{m^K_{\theta,\phi}(z_K, \mathbf{a}_K|\mathbf{u}_K, x)} \right) \mathrm{d}z_K \mathrm{d}\mathbf{u}_K$$
$$= \int m^K_{\theta,\phi}(z_K|x) \sum_{\mathbf{a}_K} \int m^K_{\theta,\phi}(\mathbf{a}_K, \mathbf{u}_K|z_K, x) \log \left(\frac{2^{-K} p_\theta(x, z_K)}{m^K_{\theta,\phi}(z_K, \mathbf{a}_K|\mathbf{u}_K, x)} \mathrm{d}\mathbf{u}_K \right) \mathrm{d}z_K$$
$$\leq \int m^K_{\theta,\phi}(z_K|x) \log \left(\sum_{\mathbf{a}_K} \int m^K_{\theta,\phi}(\mathbf{a}_K, \mathbf{u}_K|z_K, x) \frac{2^{-K} p_\theta(x, z_K)}{m^K_{\theta,\phi}(z_K, \mathbf{a}_K|\mathbf{u}_K, x)} \mathrm{d}\mathbf{u}_K \right) \mathrm{d}z_K$$

Further investigating the lower bound

- Define

$$m^K_{\theta,\phi}(z_K, \mathbf{a}_K, \mathbf{u}_K|x) = h(\mathbf{u}_K) m^K_{\theta,\phi}(z_K, \mathbf{a}_K|\mathbf{u}_K, x) \ ,$$
$$m^K_{\theta,\phi}(\mathbf{a}_K, \mathbf{u}_K|z_K, x) = m^K_{\theta,\phi}(z_K, \mathbf{a}_K, \mathbf{u}_K|x)/m^K_{\theta,\phi}(z_K|x) \ .$$

- Hence, we get

$$\mathcal{L}(\theta,\phi) \leq \int m^K_{\theta,\phi}(z_K|x) \log \left(\sum_{\mathbf{a}_K} \int m^K_{\theta,\phi}(\mathbf{a}_K, \mathbf{u}_K|z_K, x) \frac{2^{-K} p_\theta(x, z_K)}{m^K_{\theta,\phi}(z_K, \mathbf{a}_K|\mathbf{u}_K, x)} \mathrm{d}\mathbf{u}_K \right.$$
$$\leq \int m^K_{\theta,\phi}(z_K|x) \log \left(\sum_{\mathbf{a}_K} \int m^K_{\theta,\phi}(\mathbf{u}_K|z_K, x) \frac{2^{-K} p_\theta(x, z_K)}{m^K_{\theta,\phi}(z_K|\mathbf{u}_K, x)} \mathrm{d}\mathbf{u}_K \right) \mathrm{d}z_K$$

Further investigating the lower bound

- Define

$$m^K_{\theta,\phi}(z_K, \mathbf{a}_K, \mathbf{u}_K|x) = h(\mathbf{u}_K) m^K_{\theta,\phi}(z_K, \mathbf{a}_K|\mathbf{u}_K, x) \ ,$$
$$m^K_{\theta,\phi}(\mathbf{a}_K, \mathbf{u}_K|z_K, x) = m^K_{\theta,\phi}(z_K, \mathbf{a}_K, \mathbf{u}_K|x) / m^K_{\theta,\phi}(z_K|x) \ .$$

- Finally,

$$\mathcal{L}(\theta,\phi) \leq \int m^K_{\theta,\phi}(z_K|x) \log \left(\sum_{\mathbf{a}_K} \int m^K_{\theta,\phi}(\mathbf{u}_K|z_K,x) \frac{2^{-K} p_\theta(x, z_K)}{m^K_{\theta,\phi}(z_K|\mathbf{u}_K, x)} \mathrm{d}\mathbf{u}_K \right) \mathrm{d}z_K$$
$$= \int m^K_{\theta,\phi}(z_K|x) \log \left(p_\theta(x, z_K) / m^K_{\theta,\phi}(z_K|x) \right) \mathrm{d}z_K$$

Other methods for MCMC & VI: [Hof17]

- Simple method to improve a variational approximation with MCMC steps.
- First optimize variational mean-field distribution m_ϕ using classical ELBO.
- Sample $Z_0 \sim m_\phi$.
- Perform K MCMC steps (typically HMC) targetting $p_\theta(\cdot \mid x)$ to obtain sample Z_K.
- Use sample Z_K of "improved" variational distribution to update θ.
- Pros: Very straightforward to implement and understand.
- Cons: Compared to MetFlow ELBO, no feedback between MCMC steps and variational approximation !! Does not fix mode dropping in most cases as MCMC struggles to mix in a few iterations.

Improving [Hof17] with Normalizing Flows

- Method in [Hof17] is simple: easily improved.
- Idea: NeutraHMC [HSD$^+$19] improves HMC with a Normalizing flow.
- Optimize first a flow f_ϕ to minimize the KL between $\#_{f_\phi} q(z) = q(f_\phi^{-1}(z)) J_{f_\phi^{-1}}(z)$ and π the target.
- Perform HMC initialized from q with target $\#_{f_\phi^{-1}} \pi$ (in the original space, target "unwarped" by the flow).
- Push samples obtained through flow f_ϕ.
- Pros: Simplify the space on which HMC is performed, improves efficiency and flexibility.
- Cons: Additional parameters and optimization, does not necessarily correct unbiasedness of VI.

Numerical example

- Target: mixture of 8 well separated $2D$ Gaussian distributions.
- HMC kernels $L = 1$ leapfrog step, learnable stepsize and learnable mean field initialization for our `HMC-MetFlow`.
- Comparison of [Hof17] plain method, and [Hof17] method improved with a Neural Autoregressive Flow (NAF) - NeutraHMC [HSD+19].

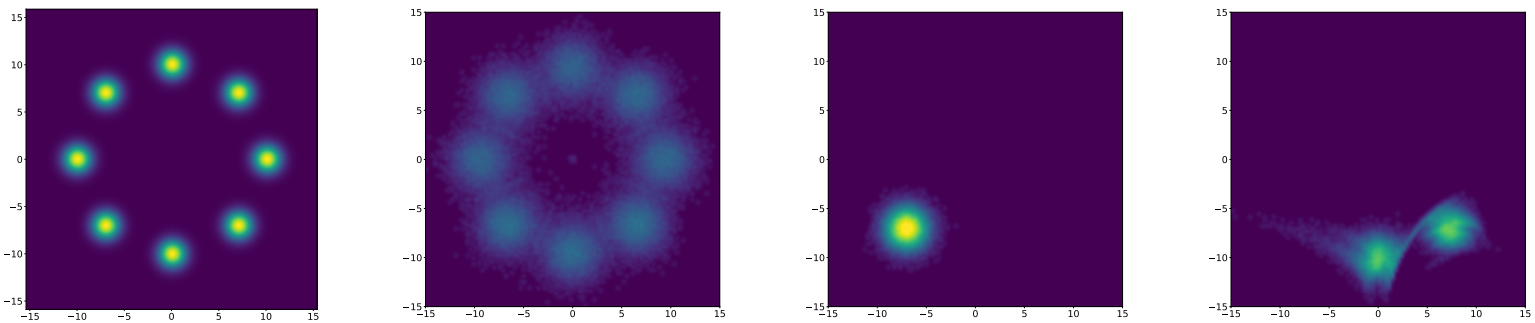


Figure: Left to right: target distribution, `HMC-MetFlow` with 2 HMC transitions, Hoffman's method [Hof17], and NeutraHMC.

MCMC with Normalizing flows

- Let $T_\phi : \mathbb{R}^D \to \mathbb{R}^D$ be a learnable invertible flow parameterized by $\phi \in \Phi$. T_ϕ should design a C^1-diffeomorphism.
- Denote by π the target distribution parameters are implicit.
- Idea: construct a Markov kernel, reversible w.r.t π based on T_ϕ.
- T_ϕ kernel: At each step k,
  - Draw a direction $V_{k+1} \in \{-1, +1\}$ with probability $1-p, p$.
  - Define a proposal $Y_{k+1} = T_\phi^{V_{k+1}}(Z_k)$.
  - Accept with probability $\alpha_{\phi, V_{k+1}}(Z_k)$ where

$$\begin{cases} \alpha_{\phi,1}(z) = 1 \wedge \frac{1-p}{p} \frac{\pi(T_\phi(z))}{\pi(z)} J_{T_\phi}(z) \ , \\ \alpha_{\phi,-1}(z) = 1 \wedge \frac{p}{1-p} \frac{\pi(T_\phi^{-1}(z))}{\pi(z)} J_{T_\phi}^{-1}(z) \ . \end{cases}$$

- The next value is proposed using either the forward or the backward mapping.

MetFlow with Normalizing flows

- Closely related to the "classical" MCMC framework.. taking the direction (V_k) as the innovation noise with distribution ν over $\{-1, +1\}$: $\nu(1) = p$, $\nu(-1) = 1 - p$.
- In this setting, the conditional Markov kernel is given by

$$Q_{\phi,v}(z, \mathsf{A}) = \alpha^1_{\phi,v}(z)\delta_{T^v_\phi(z)}(\mathsf{A}) + \alpha^0_{\phi,v}(z)\delta_z(\mathsf{A}) \ ,$$

  where we denote again $\alpha^1_{\phi,v}(z) = \alpha_{\phi,v}(z)$ and $\alpha^0_{\phi,v}(z) = 1 - \alpha_{\phi,v}(z)$.
- The integrated Markov kernel $M_{\phi,\nu}$ is defined by

$$M_{\phi,\nu}(z, \mathsf{A}) = \sum_{v \in \{-1,1\}} \nu(v) \left\{ \alpha^1_{\phi,v}(z)\delta_{T^v_\phi(z)}(\mathsf{A}) + \alpha^0_{\phi,v}(z)\delta_z(\mathsf{A}) \right\} \ .$$

- Problem: the integration is over a discrete distribution ...the proposal distribution does not have a density ! Cannot apply directly classical Metropolis-Hastings argument.

MCMC with Normalizing Flows

- Marginalizing w.r.t. the direction $v \in \{-1, +1\}$, the T_ϕ kernel defines a Markov kernel

$$M_\phi(z, \mathsf{A}) = p\alpha^1_{\phi,1}(z)\delta_{T_\phi(z)}(\mathsf{A}) + (1-p)\alpha^1_{\phi,-1}(z)\delta_{T_\phi^{-1}(z)}(\mathsf{A}) + \left\{p\alpha^0_{\phi,1}(z) + (1-p)\alpha^0_{\phi,-1}(z)\right\}\delta_z(\mathsf{A}) \ .$$

- [DB14] has shown that M_ϕ is reversible w.r.t. the target π...
- The reversibility is guaranteed because either $T_\phi(z)$ and $T_\phi^{-1}(z)$ are proposed (see next slides).

Reversibility

Let f, g be positive functions

$$\begin{aligned}\int\int \pi(\mathrm{d}z) M_\phi(z, \mathrm{d}z') f(z) g(z') = &\int \pi(z) f(z) g(T_\phi(z)) p \alpha^1_{\phi,1}(z) \mathrm{d}z \\ &+ \int \pi(z) f(z) g(T_\phi^{-1}(z)) (1-p) \alpha^1_{\phi,-1}(z) \mathrm{d}z \\ &+ \int \pi(z) f(z) g(z) \left\{ p \alpha^0_{\phi,1}(z) + (1-p) \alpha^0_{\phi,-1}(z) \right\} \mathrm{d}z\end{aligned}$$

It checks out !

Reversibility

Change of variable

$$\begin{aligned}\int\int \pi(\mathrm{d}z) M_\phi(z, \mathrm{d}z') f(z) g(z') = &\int \pi(T_\phi^{-1}(\tilde{z})) f(T_\phi^{-1}(\tilde{z})) g(\tilde{z}) p \alpha^1_{\phi,1}(T_\phi^{-1}(\tilde{z})) J_{T_\phi^{-1}}(\tilde{z}) \mathrm{d}\tilde{z} \\ &+ \int \pi(T_\phi(\tilde{z})) f(T_\phi(\tilde{z})) g(\tilde{z}) (1-p) \alpha^1_{\phi,-1}(T_\phi(\tilde{z})) J_{T_\phi}(\tilde{z}) \mathrm{d}\tilde{z} \\ &+ \int \pi(\mathrm{d}\tilde{z}) f(\tilde{z}) g(\tilde{z}) \left\{ p \alpha^0_{\phi,1}(\tilde{z}) + (1-p) \alpha^0_{\phi,-1}(\tilde{z}) \right\} \mathrm{d}\tilde{z}\end{aligned}$$

It checks out !

Reversibility

Change of variable

$$\begin{aligned}\int\int \pi(\mathrm{d}z) M_\phi(z, \mathrm{d}z') f(z) g(z') = &\int \pi(T_\phi^{-1}(\tilde{z})) f(T_\phi^{-1}(\tilde{z})) g(\tilde{z}) p \alpha^1_{\phi,1}(T_\phi^{-1}(\tilde{z})) J_{T_\phi^{-1}}(\tilde{z}) \mathrm{d}\tilde{z} \\ &+ \int \pi(T_\phi(\tilde{z})) f(T_\phi(\tilde{z})) g(\tilde{z}) (1-p) \alpha^1_{\phi,-1}(T_\phi(\tilde{z})) J_{T_\phi}(\tilde{z}) \mathrm{d}\tilde{z} \\ &+ \int \pi(\mathrm{d}\tilde{z}) f(\tilde{z}) g(\tilde{z}) \left\{ p \alpha^0_{\phi,1}(\tilde{z}) + (1-p) \alpha^0_{\phi,-1}(\tilde{z}) \right\} \mathrm{d}\tilde{z}\end{aligned}$$

Reversibility

$$p \alpha^1_{\phi,1}(T_\phi^{-1}(z)) J_{T_\phi^{-1}}(z) \pi(T_\phi^{-1}(z)) = (1-p) \alpha^1_{\phi,-1}(z) \pi(z)$$

$$(1-p) \alpha^1_{\phi,-1}(T_\phi(z)) J_{T_\phi}(z) \pi(T_\phi(z)) = p \alpha^1_{\phi,1}(z) \pi(z)$$

. It checks out !

MetFlow with Normalizing Flows

- Because the innovation is discrete distribution ...the proposal distribution does not have a density and we cannot apply directly classical Metropolis-Hastings argument to establish that $M_{\phi,\nu}$ is reversible w.r.t. π does no longer hold.
- But... most of the results derived above still hold or can be readily adapted !
- In particular, the definition of our new ELBO is still valid... enabling to learn the parameters θ, ϕ to get a full VAE.

MetFlow with Normalizing Flows

- Assumptions: a sequence $(T_{\phi,i})_{i=1}^K$ of C^1 diffeomorphisms.
- Idea: transform an initial distribution with density m_ϕ^0 by applying successively the Markov kernels

$$M_{\phi,\nu,i}(z,\mathsf{A}) = \sum_{v\in\{-1,1\}} \nu(v)\left\{\alpha^1_{\phi,v,i}(z)\delta_{T^v_{\phi,i}(z)}(\mathsf{A}) + \alpha^0_{\phi,v,i}(z)\delta_z(\mathsf{A})\right\} .$$

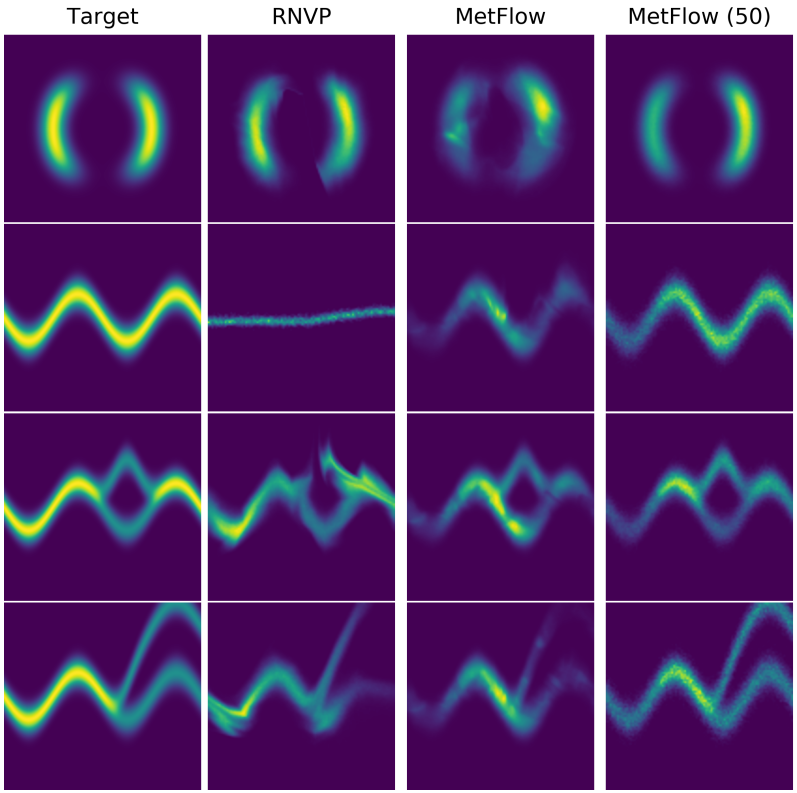
- After K steps, the marginal distribution has a density given by $m_\phi^K(z) = \sum_{\mathbf{a}_K\in\{0,1\}^K}\sum_{\mathbf{v}_K\in\{-1,1\}^K} m_\phi^K(z,\mathbf{a}_K|\mathbf{v}_K)\nu(\mathbf{v}_K)$ where

$$m_\phi^K(z,\mathbf{a}_K|\mathbf{v}_K)$$
$$= m_\phi^0\left(\bigcirc_{j=1}^K T_{\phi,j}^{-v_j a_j}(z)\right) J_{\bigcirc_{j=1}^K T_{\phi,j}^{-v_j a_j}}(z) \prod_{i=1}^K \alpha^{a_i}_{\phi,v_i,i}\left(\bigcirc_{j=i}^K T_{\phi,j}^{-v_j a_j}(z)\right) .$$

- Mixture of forward and backward transforms !
- Possible optimization using MetFlow ELBO.
- Idea: Possible to train MetFlow kernels with Normalizing Flows and repeat them after training complete – only refining final distribution at a low computational cost (no additional gradient computation)

Toy distributions

- Target: Distributions proposed by [RM15].
- Comparison Real Non Volume Preserving (Real-NVP) flows [DSDB16], and our `Real-NVP-MetFlow` with Normalizing flows. `Real-NVP-MetFlow (50)` is a specific instance of MetFlow in which more MetFlow kernels are applied after training the original 5.



Collaborative Filtering

- Collaborative filtering predicts what items a user will prefer by discovering and exploiting the similarity patterns across users and items.
- Latent factor models still largely dominate the collaborative filtering research literature due to their simplicity and effectiveness.
  - However, these models are inherently linear, which limits their modeling capacity.
  - Previous work has demonstrated that adding carefully crafted non-linear features into the linear latent factor models can significantly boost recommendation performance.
  - Recently, a growing body of work involves applying neural networks to the collaborative filtering setting with promising results
- VAE generalize linear latent-factor models
  - enable us to explore non-linear probabilistic latent-variable models, powered by neural networks, on large-scale recommendation datasets

Collaborative Filtering

- Data: Matrix user-items of incomplete interactions
- Tasks: Given binary interactions user-item, predict for each user a "complete" set of items to interact with.
  - We use $u_i n\{1, \dots, U\}$ to index users and $i \in \{1, \dots, I\}$ to index items.
  - The user-by-item interaction matrix is the interaction matrix $X \in \mathbb{N}^{U \times I}$. $x_u = [x_{u,1}, \dots, x_{u,I}]^T \in \mathbb{N}^I$ is a binary vector : $x_{u,i} = 1$ if user u had an interaction with item i.

Generative model

- For each user u, the model starts by sampling a D-dimensional latent representation z_u from a standard Gaussian prior
- The latent representation z_u is transformed via a non-linear function g_θ to produce a probability distribution $\pi_\theta(z_u)$ over I items. Here we set

$$\pi_\theta(z) = \mathrm{softmax}(g_\theta(z))$$

- Given the total number of interactions $N_u = \sum_i x_{u,i}$, x_u is assumed to be sampled from

$$x_u \mid z_u, N \sim \mathrm{Mult}\left(N_u, \pi_\theta(z_u)\right)$$

- The non-linear function $g_\theta(\cdot)$ is a multilayer perceptron with parameters θ
- The log-likelihood for user u conditioned on the latent representation is

$$\log p_\theta(x_u \mid z_u) = \sum_{i=1}^{I} x_{u,i} \log \pi_{\theta,i}(z_u) \ .$$

Evaluation of the models

- Need to have access to number of items chosen by the user for the generative model.
- To assess performance, use top-K metrics.
- Complete the items selected by an user and compare it to all of the selections using

$$\text{Recall}\,@n = \frac{|\text{relevant items} \cap \text{recommended items}|}{|\text{recommended items}|}\,,$$

$$\text{nDCG}\,@n = \frac{\text{DCG}\,@n}{\text{IDCG}\,@n}\,,$$

where

$$\text{DCG}\,@n = \sum_{i=1}^{n} \text{rel}(i)/\log_2(i+1) \text{ and } \text{IDCG}\,@n = \sum_{i=1}^{|R_n|} 1/\log_2(i+1)\,.$$

R_n: set of the n relevant items
$\text{rel}(i)$: relevance function of the i-th recommended item of the list, equal to 1 if the item ranked at i is relevant, and 0 else.

Datasets & Competitors

- Three real world datasets: Foursquare [YCM+13], Gowalla [CML11], MovieLens.
- Preprocess to binarize them to fit CF task [LKHJ18].
- Competitors
  - MultiVAE [LKHJ18] a VAE for CF.
  - WRMF [HKV08] a weighted regularized matrix factorization for implicit feedback datasets.
  - BPR [RFGST09] a Bayesian ranking method.
  - GlbAvg, a generic naive baseline (recommends the most popular items among all users).

Results

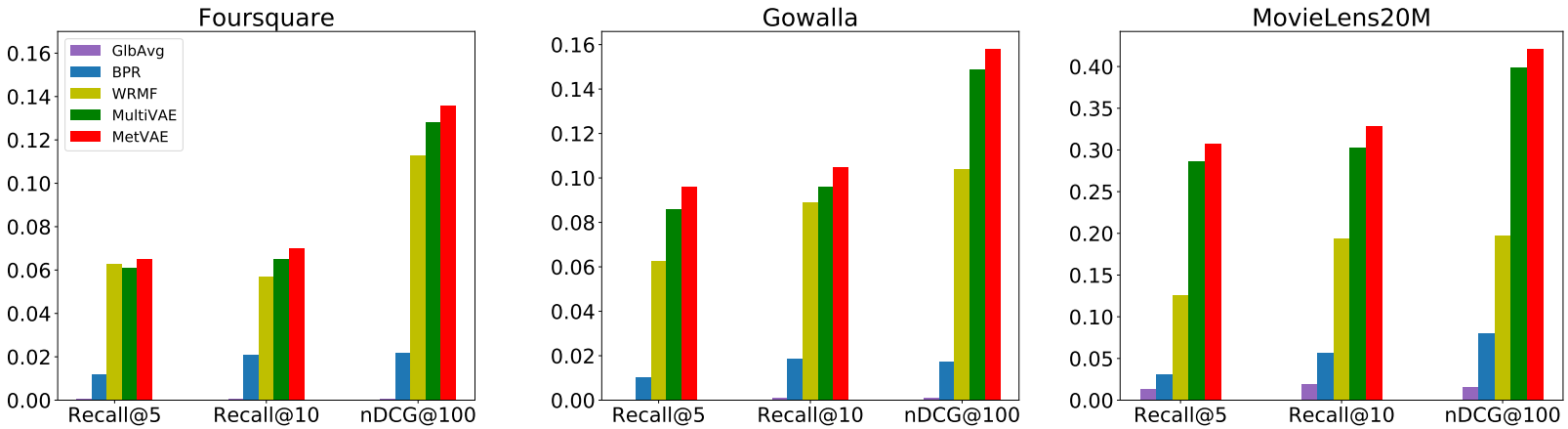


Figure: Recommendation scores in terms of $\mathrm{Recall}@5$, $\mathrm{Recall}@10$ and $\mathrm{nDCG}@100$ of the considered methods on Foursquare, Gowalla and MovieLens datasets. `MetVAE` shows consistently better results compared to other methods.

MNIST dataset and experiments

- MNIST dataset.
- Fix a generative model p_θ achieving SOTA results.
- First experiment: Consider L fixed observations.
- Approximate the posterior $p_\theta(z|(x_i)_{i=1}^L)$.
- Comparison between a NAF (SOTA Normalizing Flow) and MetFlow with 5 Real-NVP flows.
- Similar computational complexity.

Mixture of 3 on MNIST

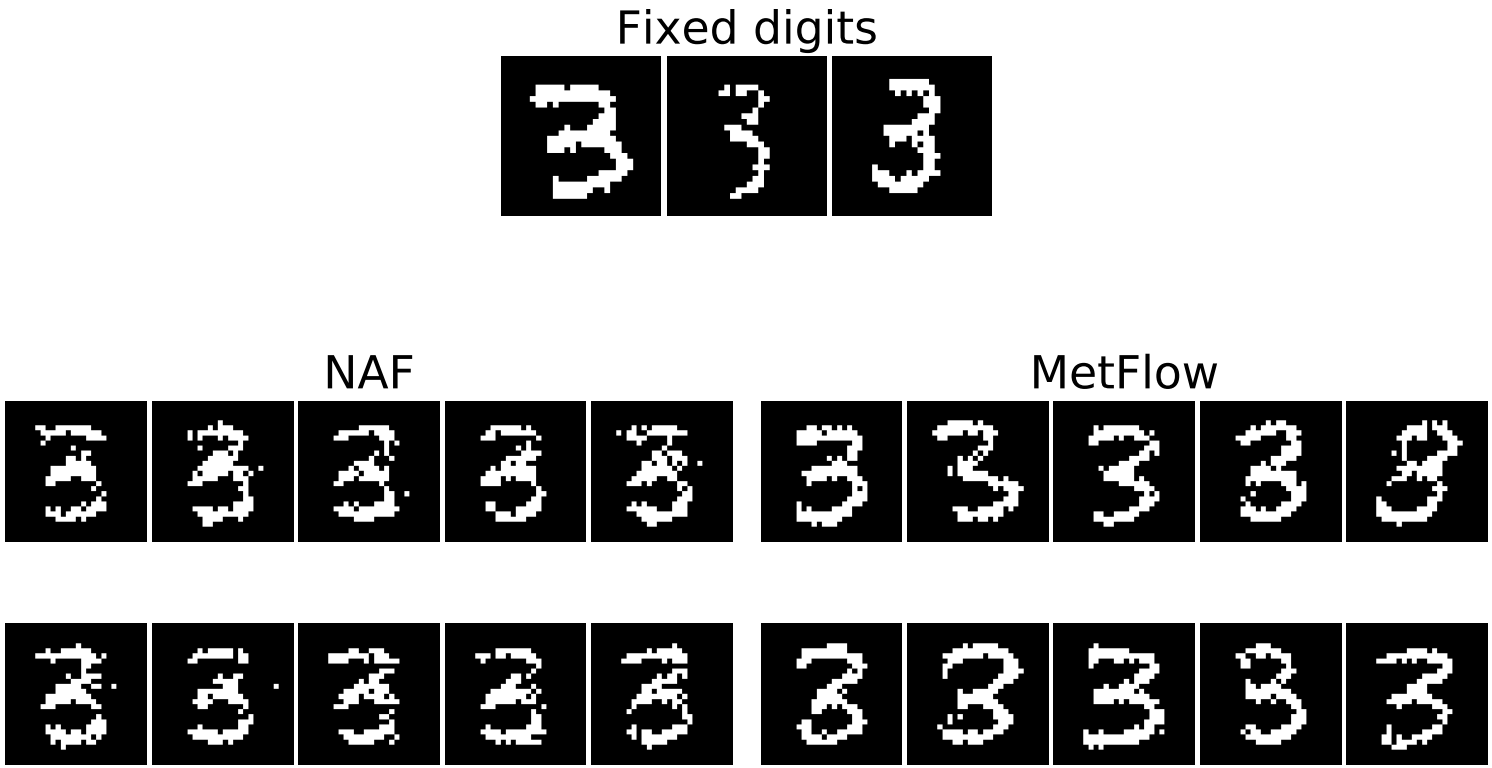


Figure: Mixture of 3 on MNIST

Inpainting on MNIST

- In-painting set-up introduced in [LHSD17].
- In-paint the top of an image using Block Gibbs sampling: Given an image x, we denote x^t, x^b the top and the bottom half pixels.
- Start from x_0.
- At each step, sample $z_t \sim p_\theta(z \mid x_t)$ and then $\tilde{x}_t \sim p_\theta(x \mid z_t)$.
- Set $x_{t+1} = (\tilde{x}_t^t, x_0^b)$.
- Use two variational approximations for $p_\theta(z \mid x)$: a mean-field approximation, a mean-field with a NAF push-forward, and MetFlow initialized at the mean-field.

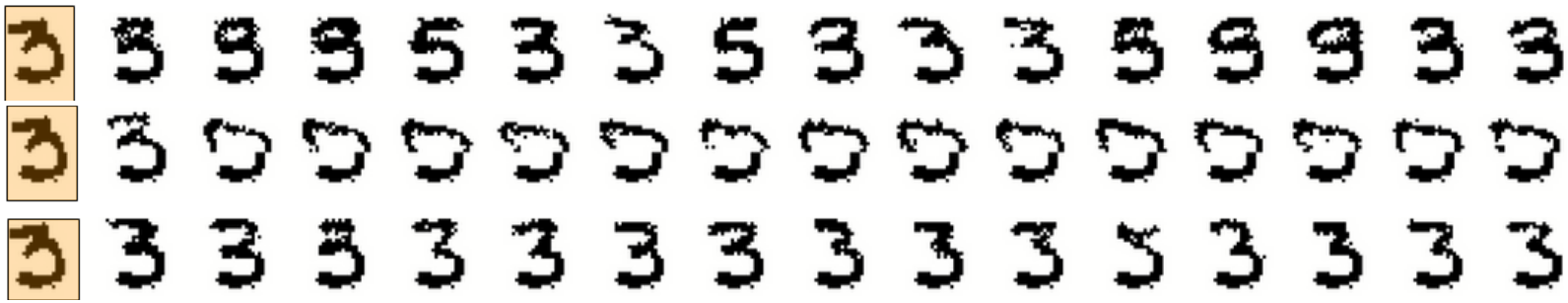


Figure: Top to bottom: Mean-Field approximation and MetFlow, Mean-Field approximation, Mean-Field Approximation and NAF. Orange samples on the left represent the initialization image.

Bibliography I


N. Bou-Rabee and S.-S. Jesús María, *Geometric integrators and the Hamiltonian Monte Carlo method*, Acta Numerica (2018), 1–92.


Anthony L Caterini, Arnaud Doucet, and Dino Sejdinovic, *Hamiltonian variational auto-encoder*, Advances in Neural Information Processing Systems, 2018, pp. 8167–8177.


Eunjoon Cho, Seth A. Myers, and Jure Leskovec, *Friendship and mobility: User movement in location-based social networks*, KDD '11, 2011.


Somak Dutta and Sourabh Bhattacharya, *Markov chain Monte Carlo based on deterministic transformations*, Statistical Methodology **16** (2014), 100–116.


Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio, *Density estimation using real NVP*, arXiv preprint arXiv:1605.08803 (2016).

Bibliography II


Yifan Hu, Yehuda Koren, and Chris Volinsky, *Collaborative filtering for implicit feedback datasets*, Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (USA), ICDM '08, IEEE Computer Society, 2008, p. 263–272.


Matthew D Hoffman and Yian Ma, *Langevin dynamics as nonparametric variational inference*.


Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner, *beta-vae: Learning basic visual concepts with a constrained variational framework*.


Matthew D. Hoffman, *Learning deep latent Gaussian models with Markov chain Monte Carlo*, Proceedings of the 34th International Conference on Machine Learning (International Convention Centre, Sydney, Australia) (Doina Precup and Yee Whye Teh, eds.), Proceedings of Machine Learning Research, vol. 70, PMLR, 06–11 Aug 2017, pp. 1510–1519.

Bibliography III


Matthew Hoffman, Pavel Sountsov, Joshua V Dillon, Ian Langmore, Dustin Tran, and Srinivas Vasudevan, *Neutra-lizing bad geometry in Hamiltonian Monte Carlo using neural transport*, arXiv preprint arXiv:1903.03704 (2019).


Daniel Levy, Matthew D Hoffman, and Jascha Sohl-Dickstein, *Generalizing Hamiltonian Monte Carlo with neural networks*, arXiv preprint arXiv:1711.09268 (2017).


Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara, *Variational autoencoders for collaborative filtering*, Proceedings of the 2018 World Wide Web Conference (Republic and Canton of Geneva, CHE), WWW '18, International World Wide Web Conferences Steering Committee, 2018, p. 689–698.


R. M. Neal, *MCMC using Hamiltonian dynamics*, Handbook of Markov Chain Monte Carlo (2011), 113–162.

Bibliography IV


Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme, *Bpr: Bayesian personalized ranking from implicit feedback*, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (Arlington, Virginia, USA), UAI '09, AUAI Press, 2009, p. 452–461.


Danilo Rezende and Shakir Mohamed, *Variational inference with normalizing flows*, International Conference on Machine Learning, 2015, pp. 1530–1538.


Tim Salimans, Diederik Kingma, and Max Welling, *Markov chain Monte Carlo and variational inference: Bridging the gap*, International Conference on Machine Learning, 2015, pp. 1218–1226.


Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther, *Ladder variational autoencoders*, Advances in neural information processing systems, 2016, pp. 3738–3746.

Bibliography V

Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann, *Time-aware point-of-interest recommendation*, Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), SIGIR '13, ACM, 2013, pp. 363–372.