Adaptive Latent Modeling and Optimization via Neural Networks and Langevin Diffusion

Yixuan Qiu[1] Xiao Wang[2]

The 36th QPRC, 2019

[1]Department of Statistics, Carnegie Mellon University

[2]Department of Statistics, Purdue University

Adaptive **L**atent **M**odeling and **O**ptimization via Neural **N**etworks and Langevin **D**iffusion

(Image: http://dreamicus.com/data/almond/almond-05.jpg)

Outline

Motivation

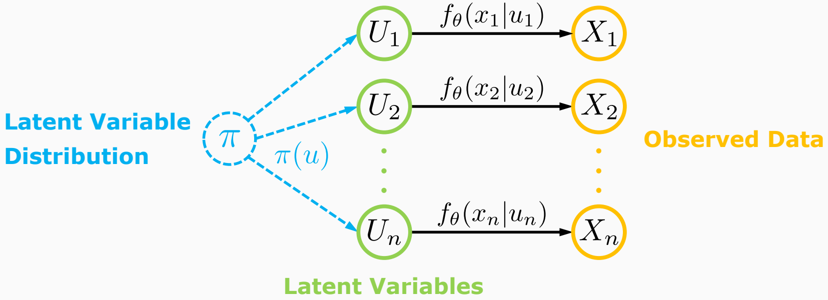
Latent Variable Model

- A general and powerful way to modeling complicated data distribution

$$f(x_1, \ldots, x_n) = \prod_{i=1}^{n} f(x_i), \quad f(x_i) = \int f(x_i|u_i)\pi(u_i)\mathrm{d}u_i$$

- Observed data points $X_i \in \mathbb{R}^p$
- Unobserved latent variables $U_i \in \mathbb{R}^d$
- Marginal latent distribution $\pi(u)$
- Latent-to-data distribution $f(x|u)$

Diagram for Latent Variable Model



Some Well-Known Examples

Hierarchical Bayesian models

$$X_i|\mu_i \sim N(\mu_i, \sigma_i^2), \quad \mu_i \overset{iid}{\sim} N(\mu_0, \tau_0^2)$$

(Generalized) Linear mixed models

$$Y_i = W_i\beta + Z_i b_i + \varepsilon_i \;\Leftrightarrow\; Y_i|b_i \sim N(W_i\beta + Z_i b_i, \Sigma_i)$$

$$b_i \overset{iid}{\sim} N(0, D)$$

Gaussian mixture models

$$X_i|\{U_i = c\} \sim N(\mu_c, \Sigma_c), \quad P(U_i = c) = \pi_c$$

Related Methods

- Bayesian inference (Gelman et al., 2014)
  - Based on the posterior distribution $p(u_i|x_i) = f(x_i|u_i)\pi(u_i)/f(x_i)$
  - Typically computed using Markov chain Monte Carlo (MCMC, Gilks, Richardson, and Spiegelhalter, 1995)
  - **Pro**: Widely used in real applications
  - **Pro**: Elegant and well-developed statistical properties
  - **Con**: Requires fully known $\pi(u)$ and $f(x|u)$
  - **Con**: High computational cost with MCMC; nontrivial to scale to large data sets

Related Methods cont.

- The expectation-maximization algorithm (EM, Dempster, Laird, and Rubin, 1977)
    - Latent variables as “missing data”
    - Computes the maximum likelihood estimator (MLE) for θ
    - **Pro**: Allows for unknown parameters in $\pi(u)$ and $f(x|u)$, thus bringing more flexibility in modeling
    - **Con**: Mostly used for point estimation
    - **Con**: E-step does not have closed form for complicated models
    - **Con**: M-step is also challenging for big data

Related Methods cont.

- Variational inference (Jordan et al., 1999; Blei, Kucukelbir, and McAuliffe, 2017)
    - An alternative approach for large-scale Bayesian inference
    - Approximates the true posterior using a simpler distribution
    - **Pro**: Very efficient in computation
    - **Pro**: Easy to scale to large data sets
    - **Con**: Lack of accuracy in the inference result

A (Subjective) Summary

	Ease of Modeling	Efficiency of Computation	Accuracy of Inference
Bayesian Inference	★★☆	★★☆	★★★
EM Algorithm	★★★	★☆☆ / ★★½ Highly depends on the model	★★★
Variational Inference	★★★	★★★	★★☆

The ALMOND Framework

Overview

- A flexible and data-driven specification of the latent variable distribution $\pi(u)$ via neural networks
- The latent-to-data distribution $f_\theta(x|u)$ can also contain unknown parameters θ
- An efficient computational method based on:
    - Stochastic gradient methods (Robbins and Monro, 1951; Bottou et al., 2018)
    - The Langevin sampling algorithm (Roberts et al., 1996; Roberts and Stramer, 2002; Dalalyan, 2017)
- Theoretical guarantees on the convergence of the algorithm

Inference Objective

- Input
  - Observed data points $X_1, X_2, \ldots, X_n$
  - Latent-to-data distribution $f_\theta(x|u)$ up to an unknown parameter vector θ
- Output
  - Estimated latent variable distribution $\hat{\pi}(u)$
  - Estimate of θ: $\hat{\theta}$
  - Conditional distribution of the latent variable given the data $p(u_i|x_i)$

Modeling: Adaptive Latent Variable Distribution

- $\pi(u)$ controls the expressive power of the marginal data distribution $f(x) = \int f(x|u)\pi(u)\mathrm{d}u$
- We specify an adaptive $\pi(u)$ through a probability transformation $U_i = h_\eta(Z_i)$
- $Z_i \in \mathbb{R}^r$ follows a known distribution, e.g. $N(0, I_r)$
- $h_\eta : \mathbb{R}^r \mapsto \mathbb{R}^d$ is represented by a deep neural network (DNN), where η contains the network parameters
- $\hat{\pi}(u) \Leftrightarrow h_{\hat{\eta}}$

Computation: Challenges and Solutions

- η and θ can be estimated by maximizing the log-likelihood function $\ell(\theta, \eta; x) \equiv \log[f(x)]$
- However, $f(x) = \int f(x|u)\pi(u)\mathrm{d}u$ involves a potentially high-dimensional integration
- A direct optimization over η and θ is intractable
- Our method
  - First, obtain a rudimentary estimation for unknown quantities using the efficient variational autoencoder framework (VAE, Kingma and Welling, 2013)
  - Then proceeds with a bias correction procedure to achieve a high accuracy of the inference results
  - Combines the efficiency of VAE and the accuracy of EM algorithm

A Bit of Background Knowledge

- For **any** distribution $q(z|x)$,

$$\ell(\beta; x) \geq \mathcal{L}(\beta; q, x) := \mathbb{E}_{z\sim q(z|x)}\left[\log f_\beta(x|z)\right] - \mathcal{D}\left[q(z|x) \| \pi_0(z)\right]$$

- $f_\beta(x|z) := f_\theta(x|h_\eta(z))$, $\beta = (\theta, \eta)$
- $\mathcal{D}[q\|p]$ is the Kullback–Leibler divergence from p to q
- Instead of maximizing $\ell(x)$, VAE does the following
  - Choose $q(z|x)$ to be $N(\mu_\phi(x), \mathrm{diag}(\sigma^2_\phi(x)))$
  - $\mu_\phi(\cdot)$ and $\sigma^2_\phi(\cdot)$ are DNNs with parameter ϕ
  - Optimizes $\mathcal{L}(\beta; q_\phi, x)$ over the parameters β and ϕ

The New Method

- VAE is fast, but biased, even with an infinite sample size
- It has the wrong target: a lower bound instead of $\ell(\beta; x)$
- We propose a new method that targets on the true $\ell(\beta; x)$
- Define

$$\mathcal{L}(\beta, \tilde{\beta}; x) = \int \log \left[\frac{f_\beta(x|z)\pi_0(z)}{p_{\tilde{\beta}}(z|x)} \right] p_{\tilde{\beta}}(z|x) \mathrm{d}z$$

- When $\tilde{\beta} = \beta$, we have $\mathcal{L}(\beta, \beta; x) = \ell(\beta; x)$
- The quantity $g(\beta, \tilde{\beta}; x) = \partial \mathcal{L}(\beta, \tilde{\beta}; x)/\partial \beta$ is similar to a gradient when $\tilde{\beta} = \beta$
- We iteratively update the parameter estimate β_t:

$$\beta_{t+1} = \beta_t + \alpha_t \cdot \tilde{g}(\beta_t; x, W_t)$$

- $\tilde{g}(\beta_t; x, W_t)$ is a stochastic approximation to $g(\beta_t, \beta_t; x)$

The Langevin Algorithm

- Define $G(\beta; x, z) = \partial \log[f_\beta(x|z)]/\partial\beta$, then $g(\beta_t, \beta_t; x) = \mathbb{E}_{z\sim p_{\beta_t}(z|x)} G(\beta_t; x, z)$
- We want to obtain a sequence of random vectors $W_t^{(1)}, \ldots, W_t^{(M_t)}$ such that

$$\tilde{g}(\beta_t; x, W_t) = \frac{1}{M_t}\sum_{i=1}^{M_t} G(\beta_t; x, W_t^{(i)}) \approx g(\beta_t, \beta_t; x)$$

- The Langevin algorithm is simple and easy to compute:

$$W_t^{(k)} = W_t^{(k-1)} + \gamma_t \cdot v_t(W_t^{(k-1)}) + \sqrt{2\gamma_t} \cdot \xi_t^{(k)}$$

where γ_t is the step size, $v_t(z) = \partial \log[f_{\beta_t}(x|z)\pi_0(z)]/\partial z$, and $\xi_t^{(k)} \overset{iid}{\sim} N(0, I_r)$

Theoretical Properties

Theorem
Under regularity conditions, for every $t \in \mathbb{N}$ and any $0 < \varepsilon_t < 1$, there exists a constant $C_t > 0$ such that whenever $\gamma_t \leq C_t \varepsilon_t$ and $M_t \geq \gamma_t^{-2}$, we have

$$\|\mathbb{E}_{W_t}[\tilde{g}(\beta_t; x, W_t)] - g(\beta_t, \beta_t; x)\| \leq \varepsilon_t$$
$$\mathbb{E}_{W_t}\left[\|\tilde{g}(\beta_t; x, W_t)] - g(\beta_t, \beta_t; x)\|^2\right] \leq \varepsilon_t$$

- It shows that $\tilde{g}(\beta_t; x, W_t)$ is a biased estimator for $g(\beta_t, \beta_t; x)$
- But we can control its bias to any small number ε_t

Convergence Analysis

Theorem
Under regularity conditions, let $\{\alpha_t\}$ *and* $\{\varepsilon_t\}$ *be two positive and decreasing sequences such that* $\sum_{t=1}^{\infty} \alpha_t = \infty$, $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$, *and* $\sum_{t=1}^{\infty} \alpha_t \varepsilon_t^2 < \infty$, *then we have*

$$\liminf_{t \to \infty} E\left[\|g(\beta_t, \beta_t; x)\|^2\right] = 0.$$

In particular, the above conditions hold if $\alpha_t \asymp O(t^{-1})$ *and* $\varepsilon_t = O(t^{-c})$ *for any* $c > 0$.

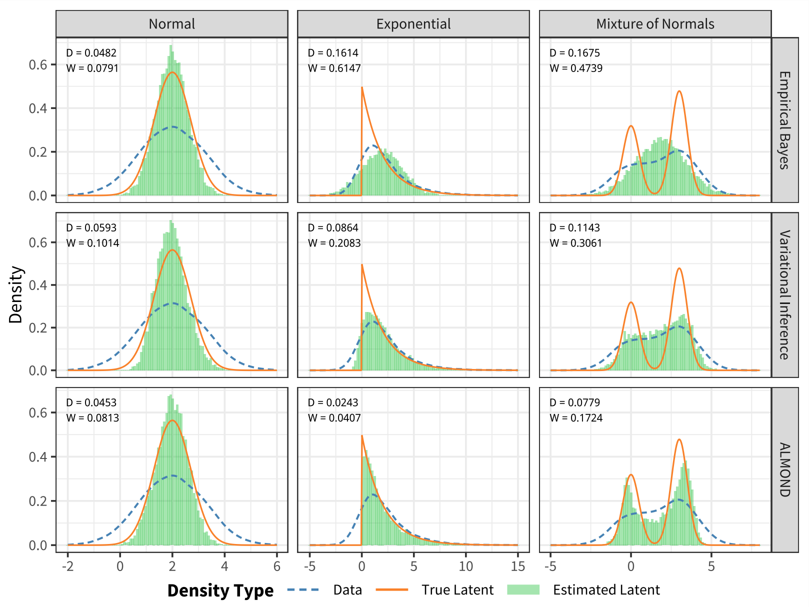
Moreover, if there exists a β^* *such that* $\|g(\beta^*, \beta^*; x)\| = 0$, *then* $\partial \ell(\beta; x)/\partial \beta|_{\beta=\beta^*} = 0$.

Numerical Experiments

Many-Normal-Means Problem

- $U_i \overset{iid}{\sim} \pi(u)$, $X_i | \{U_i = u\} \sim N(\mu, 1)$, $i = 1, 2, \ldots, 1000$
- Three true latent distributions
    - $\pi = N(1, 0.5^2)$
    - $\pi = Exp(2)$, mean $= 2$
    - $\pi = 0.4 \cdot N(0, 0.5^2) + 0.6 \cdot N(3, 0.5^2)$
- Compare empirical Bayes, variational inferene, and ALMOND

Result

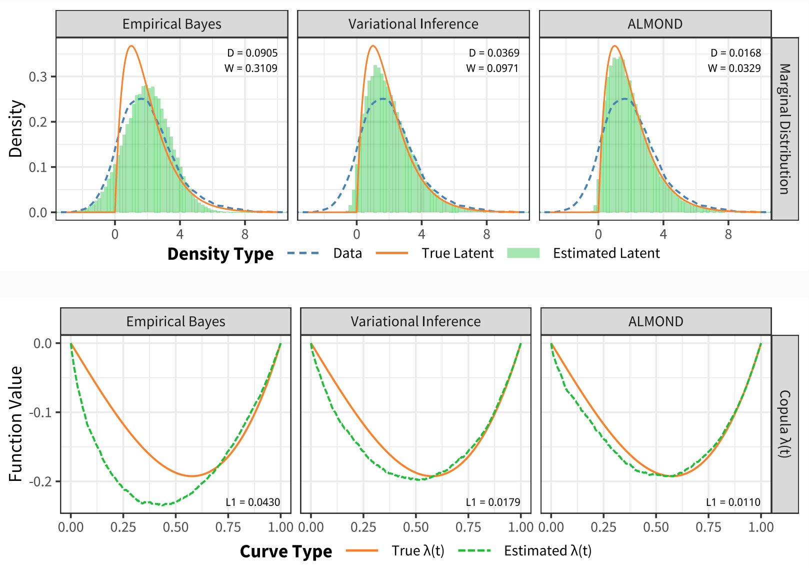


Normal
Exponential
Mixture of Normals
Empirical Bayes
Variational Inference
ALMOND
Density
D = 0.0482
W = 0.0791
D = 0.1614
W = 0.6147
D = 0.1675
W = 0.4739
D = 0.0593
W = 0.1014
D = 0.0864
W = 0.2083
D = 0.1143
W = 0.3061
D = 0.0453
W = 0.0813
D = 0.0243
W = 0.0407
D = 0.0779
W = 0.1724
Density Type
Data
True Latent
Estimated Latent

Multivariate Copula Model

- $P(U_1 \leq u_1, \ldots, U_{10} \leq u_{10}) = C(F(u_1), \ldots, F(u_{10}))$,
  $X | \{U = u\} \sim N(u, I_{10})$
- $F(u)$ is the c.d.f. of $Gamma(2)$
- $C(u_1, \ldots, u_{10}) = \varphi^{-1}(\varphi(u_1) + \cdots + \varphi(u_{10}))$, $\varphi = t^{-2} - 1$
- Study the estimates of $F(u)$ and $\lambda(t) = \varphi(t)/\varphi'(t)$
- Compare empirical Bayes, variational inferene, and ALMOND

Result

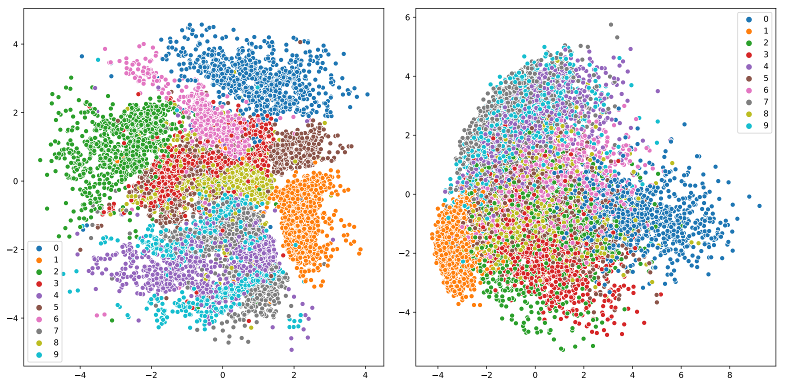

Empirical Bayes
Variational Inference
ALMOND
D = 0.0905
W = 0.3109
D = 0.0369
W = 0.0971
D = 0.0168
W = 0.0329
Density
Marginal Distribution
Density Type
Data
True Latent
Estimated Latent
L1 = 0.0430
L1 = 0.0179
L1 = 0.0110
Function Value
Copula λ(t)
Curve Type
True λ(t)
Estimated λ(t)

MNIST Data

- The well-known MNIST handwritten digits data
- Use $Z \sim N(0, I_2)$ to represent the low-dimensional latent space
- Compute the latent coordinates $\mathbb{E}(Z|X = x)$ for nonlinear dimensionality reduction

Result

- Left: Dimensionality reduction by ALMOND
- Right: Dimensionality reduction by PCA



THANK YOU!