A Stochastic Variational Inference Approach for Semiparametric Distributional Regression

Final Kolloquium MSc Applied Statistic

Sebastian Lorek

First supervisor: Prof. Dr. Thomas Kneib
Second supervisor: Gianmarco Callegher

2023-11-28

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Table of Contents I

Introduction

Statistical Infrence

Frequentist inference:

- ML estimation (Newton-Rhapson/Fisher-Scoring)
- Hyperparameter estimation
- REML for latent parameters (random-effects)
- SGD, Automatic differentiation in machine/deep learning

Bayesian inference:

- Conjugate models, full conditional conjugate models, non-conjugate models
- Gibbs sampling
- Rejection/Importance sampling
- MCMC (IWLS, HMC, NUTS)
- MAP estimation in combination with the Laplace approximation

Bayesian Inference

- Focal point of interest is the posterior distribution
- General posterior specification for Bayesian regression

$$
\begin{aligned}
p(\boldsymbol{\theta}|\mathbf{y},\mathcal{D}) &= \frac{p(\mathbf{y},\boldsymbol{\theta}|\ \mathcal{D})}{p(\mathbf{y}|\mathcal{D})} && (1)\\
&= \frac{p(\mathbf{y}|\ \boldsymbol{\theta},\mathcal{D})p(\boldsymbol{\theta})}{p(\mathbf{y}|\mathcal{D})} && (2)\\
&= \frac{p(\mathbf{y}|\ \boldsymbol{\theta},\mathcal{D})p(\boldsymbol{\theta})}{\int p(\mathbf{y}|\ \boldsymbol{\theta},\mathcal{D})p(\boldsymbol{\theta})\,d\boldsymbol{\theta}} && (3)\\
&\propto p(\mathbf{y}|\ \boldsymbol{\theta},\mathcal{D})p(\boldsymbol{\theta}) && (4)
\end{aligned}
$$

- Primary challenge is the calculation of the normalizing constant/evidence
- We need approximate methods that bypass the direct calculation of the normalizing constant

- MCMC methods have been **so far** the work horse for Bayesian inference in classical statistics
- Set up Markov chain (adhere to detailed balance and ergodicity) and sample . . .
- Enjoyes nice properties
  - Assurance of convergence to the true posterior
- But unfortunately does not scale well for modern applications
  - High dimensional parameter spaces (tausands of parameters)
  - Large dataset
  - Many latent and hyper-parameters
- Trade some **accuracy** for **scalability**
- Make use of SGD and automatic differentiation which works well for inference in machine/deep learning (frequentist inference)

Theory

Variational Inference

- VI is a method from machine learning to **approximate** probability densities (Jordan et al. 1999)
- Approximate posterior with a variational distribution $q(\boldsymbol{\theta})$ from a predefined (parametric) variational family $\mathcal{Q}$

$$q(\boldsymbol{\theta}) \in \mathcal{Q}$$

- Use optimization (SGD) to find a member that is as closely as possible to the true posterior
- What means close in terms of distributions ?

Kullback-Leibler divergence

- Divergence measure (Kullback and Leibler 1951) that quantifies the proximity between two probability distributions

$$
\begin{aligned}
\mathrm{D}_{\mathrm{KL}} &= \int q(\boldsymbol{\theta}) \ln \left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{y}, \mathcal{D})} \right) \, d\boldsymbol{\theta} \\
&= \mathrm{E}_{q(\boldsymbol{\theta})} \left[\ln \left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{y}, \mathcal{D})} \right) \right]
\end{aligned}
$$

- In short $\mathrm{D}_{\mathrm{KL}}\left(q||\, p\right)$
- It holds that $\mathrm{D}_{\mathrm{KL}}\left(q||\, p\right) \geq 0$
- Has some nice properties but also drawbacks (not a distance/metric)

Optimization objective

- Use optimization to find a variational distribution that is as close as possible in terms of the divergence measure to the true posterior

$$
\begin{aligned}
q^*(\boldsymbol{\theta}) &= \underset{q(\boldsymbol{\theta}) \in \mathcal{Q}}{\arg\min}\, \mathrm{D}_{\mathrm{KL}}\left(q(\boldsymbol{\theta}) || \; p(\boldsymbol{\theta}|\mathbf{y}, \mathcal{D})\right) \\
&= \underset{q(\boldsymbol{\theta}) \in \mathcal{Q}}{\arg\min} \int q(\boldsymbol{\theta}) \ln \left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{y}, \mathcal{D})} \right) d\boldsymbol{\theta} \\
&= \underset{q(\boldsymbol{\theta}) \in \mathcal{Q}}{\arg\min}\, \mathrm{E}_{q(\boldsymbol{\theta})} \left[\ln \left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{y}, \mathcal{D})} \right) \right].
\end{aligned}
$$

- Flexibility of $\mathcal{Q}$ significantly influences the optimization
  - Complex $\mathcal{Q} \rightarrow$ better approximation, but increased complexity during optimization
  - Simple $\mathcal{Q} \rightarrow$ worse approximation, but simpler optimization
- Objective offers theoretical insights but remains **infeasible** to compute, due to containing the posterior (evidence)

Evidence Lower Bound

- Think about a way to introduce a mathematical **equivalent** objective that does not depend on the evidence
- Lets start with the log-evidence

$$
\begin{aligned}
\ln(p(\mathbf{y}|\mathcal{D})) &= \int q(\boldsymbol{\theta}|\phi) \ln(p(\mathbf{y}|\mathcal{D}))\, d\boldsymbol{\theta} && (5) \\
&= \int q(\boldsymbol{\theta}|\phi) \ln\left(\frac{p(\mathbf{y}|\ \mathcal{D})p(\boldsymbol{\theta}|\mathbf{y},\mathcal{D})}{p(\boldsymbol{\theta}|\mathbf{y},\mathcal{D})}\right)\, d\boldsymbol{\theta} && (6) \\
&= \int q(\boldsymbol{\theta}|\phi) \ln\left(\frac{p(\mathbf{y},\boldsymbol{\theta}|\mathcal{D})}{p(\boldsymbol{\theta}|\mathbf{y},\mathcal{D})}\right)\, d\boldsymbol{\theta} && (7) \\
&= \int q(\boldsymbol{\theta}|\phi) \ln\left(\frac{p(\mathbf{y},\boldsymbol{\theta}|\mathcal{D})}{q(\boldsymbol{\theta}|\phi)}\frac{q(\boldsymbol{\theta}|\phi)}{p(\boldsymbol{\theta}|\mathbf{y},\mathcal{D})}\right)\, d\boldsymbol{\theta} && (8) \\
&= \int q(\boldsymbol{\theta}|\phi) \ln\left(\frac{p(\mathbf{y},\boldsymbol{\theta}|\mathcal{D})}{q(\boldsymbol{\theta}|\phi)}\right)\, d\boldsymbol{\theta} + \int q(\boldsymbol{\theta}|\phi) \ln\left(\frac{q(\boldsymbol{\theta}|\phi)}{p(\boldsymbol{\theta}|\mathbf{y},\mathcal{D})}\right)\, d\boldsymbol{\theta} && (9) \\
&= \mathrm{E}_{q(\boldsymbol{\theta}|\phi)}\left[\ln\left(\frac{p(\mathbf{y},\boldsymbol{\theta}|\mathbf{X})}{q(\boldsymbol{\theta}|\phi)}\right)\right] + \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{\theta}|\phi)||p(\boldsymbol{\theta}|\mathbf{y},\mathcal{D})) && (10)
\end{aligned}
$$

- Parameterize q with ϕ

Optimization objective revisited

- Rewrite 10 as

$$\ln(p(\mathbf{y}|\mathbf{X})) - \mathrm{D}_{\mathrm{KL}}(q||p) = \mathrm{E}_{q(\boldsymbol{\theta}|\phi)}\left[\ln\left(\frac{p(\mathbf{y},\boldsymbol{\theta}|\mathcal{D})}{q(\boldsymbol{\theta}|\phi)}\right)\right] \tag{11}$$

- And take arg max w.r.t. ϕ

$$\begin{aligned}
\arg\max_{\phi} \ln(p(\mathbf{y}|\mathcal{D})) - \mathrm{D}_{\mathrm{KL}}(q||p) &= \arg\max_{\phi} \mathrm{E}_{q(\boldsymbol{\theta}|\phi)}\left[\ln\left(\frac{p(\mathbf{y},\boldsymbol{\theta}|\mathcal{D})}{q(\boldsymbol{\theta}|\phi)}\right)\right]\\
\arg\max_{\phi} -\mathrm{D}_{\mathrm{KL}}(q||p) &= \arg\max_{\phi} \mathrm{E}_{q(\boldsymbol{\theta}|\phi)}\left[\ln\left(\frac{p(\mathbf{y},\boldsymbol{\theta}|\mathcal{D})}{q(\boldsymbol{\theta}|\phi)}\right)\right]\\
\arg\min_{\phi} \mathrm{D}_{\mathrm{KL}}(q||p) &= \arg\max_{\phi} \mathrm{E}_{q(\boldsymbol{\theta}|\phi)}\left[\ln\left(\frac{p(\mathbf{y},\boldsymbol{\theta}|\mathcal{D})}{q(\boldsymbol{\theta}|\phi)}\right)\right]
\end{aligned}$$

- New optimization objective

$$\hat{\phi} = \arg\max_{\phi} \mathrm{E}_{q(\boldsymbol{\theta}|\phi)} \left[\ln \left(\frac{p(\mathbf{y}, \boldsymbol{\theta}|\mathcal{D})}{q(\boldsymbol{\theta}|\phi)} \right) \right]$$
$$= \arg\max_{\phi} \mathrm{ELBO}(\phi)$$

- Take a breath 😮‍💨

- What does the ELBO ?

$$
\begin{aligned}
\mathrm{ELBO}(\phi) &= \mathrm{E}_{q(\boldsymbol{\theta}|\phi)}\left[\ln\left(\frac{p(\mathbf{y},\boldsymbol{\theta}|\mathcal{D})}{q(\boldsymbol{\theta}|\phi)}\right)\right] && (12)\\
&= \mathrm{E}_{q(\boldsymbol{\theta}|\phi)}\left[\ln\left(\frac{p(\mathbf{y}|\mathcal{D},\boldsymbol{\theta})p(\boldsymbol{\theta})}{q(\boldsymbol{\theta}|\phi)}\right)\right] && (13)\\
&= \mathrm{E}_{q(\boldsymbol{\theta}|\phi)}\left[\ln(p(\mathbf{y}|\mathcal{D},\boldsymbol{\theta}))\right] + \mathrm{E}_{q(\boldsymbol{\theta}|\phi)}\left[\ln(p(\boldsymbol{\theta}))\right] - \mathrm{E}_{q(\boldsymbol{\theta}|\phi)}\left[\ln(q(\boldsymbol{\theta}|\phi))\right] && (14)\\
&= \mathrm{E}_{q(\boldsymbol{\theta}|\phi)}\left[\ln(p(\mathbf{y}|\mathcal{D},\boldsymbol{\theta}))\right] - \mathrm{D}_{\mathrm{KL}}\left(q(\boldsymbol{\theta}|\phi)||p(\boldsymbol{\theta})\right) && (15)
\end{aligned}
$$

- Actually something similar to MAP/ML 🤔

Variational family

- What is the $\mathcal{Q}$ and thus $q(\boldsymbol{\theta}|\phi)$?
- Start simple

$$q(\boldsymbol{\theta}|\phi) = \prod_{j=1}^{J} q_j(\theta_j|\ \phi_j)$$

- Known as mean-field variational family
- q_j factors in the variational distribution
  - Some parameteric distribution that respects the parameter space of $\boldsymbol{\theta}_j$
  - All model parameters are independent

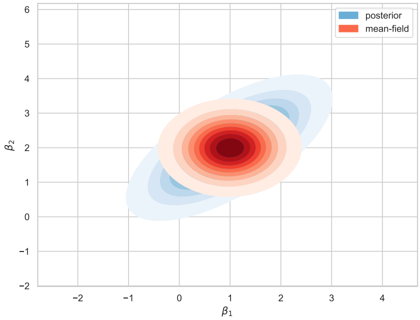


Figure 1: Mean-field approximation to a 2-D multivariate normal distribution, based on Blei et al. (ibid., p. 9, figure 1).

- Augment the variational distribution to blocks of parameters
- Structured mean-field variational inference (Wainwright and Jordan 2007)

$$q(\boldsymbol{\theta}|\phi) = \prod_{j=1}^{J} q_j(\boldsymbol{\theta}_j|\ \phi_j)$$

- Captures interdependencies for blocks of parameters

Coordinate ascent variational inference (side node)

- Traditional way of solving opt. objective is CAVI (Blei et al. 2017)
  - However CAVI does not scale well
  - Closely connected to Gibbs sampling
  - Only works for conditional conjugate models
  - If you search for VI CAVI is still all over the place, see f.e. wikipedia
- Of course we want to be able to also conduct inference in non-conjugate models ✗

Stochastic variational inference

- SGD to optimize the ELBO (Hoffman et al. 2012)
- 2 sources of stochasticity
  - We use a subset $\mathcal{I}$ of the data in each iteration (ELBO remains unbiased)
  - We need to evaluate the integral in the ELBO

$$\text{ELBO}(\phi)_{\mathcal{I}} = \int q(\boldsymbol{\theta}|\phi) \ln \left(\frac{p(\mathbf{y}_{\mathcal{I}}, \boldsymbol{\theta}|\mathcal{D}_{\mathcal{I}})}{q(\boldsymbol{\theta}|\phi)} \right) d\boldsymbol{\theta} \tag{16}$$

$$\nabla_{\phi}\text{ELBO}(\phi) = \nabla_{\phi} \int q(\boldsymbol{\theta}|\phi) \ln \left(\frac{p(\mathbf{y}_{\mathcal{I}}, \boldsymbol{\theta}|\mathcal{D}_{\mathcal{I}})}{q(\boldsymbol{\theta}|\phi)} \right) d\boldsymbol{\theta} \tag{17}$$

- Common method to solve this problem is Monte Carlo integration

$$\nabla_\phi \mathrm{ELBO}(\phi)_{\mathcal{I}} = \nabla_\phi \int q(\boldsymbol{\theta}|\phi) \ln\left(\frac{p(\mathbf{y}_{\mathcal{I}}, \boldsymbol{\theta}|\mathcal{D}_{\mathcal{I}})}{q(\boldsymbol{\theta}|\phi)}\right) d\boldsymbol{\theta} \tag{18}$$

$$\approx \nabla_\phi \frac{1}{S}\sum_{s=1}^{S} \ln\left(\frac{p(\mathbf{y}_{\mathcal{I}}, \boldsymbol{\theta}^s|\mathcal{D}_{\mathcal{I}})}{q(\boldsymbol{\theta}^s|\phi)}\right), \ \boldsymbol{\theta}^s \sim q(\boldsymbol{\theta}|\phi). \tag{19}$$

- But this does not work 😢
- If we change ϕ even infinitessimal the samples $\boldsymbol{\theta}^s$ are invalid, which we used to calculate $\nabla_\phi \mathrm{ELBO}(\phi)_{\mathcal{I}}$ in the first place

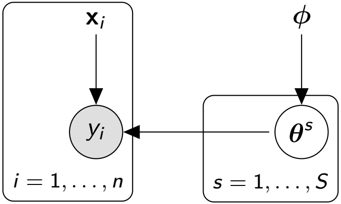


Figure 2: Plate notation of the dependence structure in a Bayesian regression model for VI, when sampling from the variational distribution.

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Reparameterization gradient estimator

- Reparapemeterize $\boldsymbol{\theta} = \mathbf{g}_{\phi}(\boldsymbol{\epsilon})$, with a bijective function $\mathbf{g}_{\phi}$ such that we can use SGD with Monte Carlo integration (Kingma and Welling 2013; Kucukelbir et al. 2016; Rezende et al. 2014)
- Amounts to using the (multivariate) change of variable theorem for probability density functions

$$\mathbf{g}_{\phi} : \mathbb{R}^d \to \mathbb{R}^d;\ \boldsymbol{\theta}, \boldsymbol{\epsilon} \in \mathbb{R}^d$$

$$\boldsymbol{\theta} = \mathbf{g}_{\phi}(\boldsymbol{\epsilon}),\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$q(\boldsymbol{\theta}|\phi) = \begin{cases} p_{\boldsymbol{\epsilon}}(\mathbf{g}_{\phi}^{-1}(\boldsymbol{\theta})) \left|\det(\mathbf{J}_{\mathbf{g}_{\phi}^{-1}})\right|, & \text{if } \boldsymbol{\theta} \text{ is in the codomain of } \mathbf{g}_{\phi} \\ 0, & \text{else.} \end{cases}$$

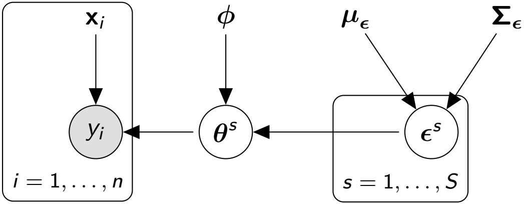


Figure 3: Plate notation of the dependence structure in a Bayesian regression model in VI, using the "reparameterization-trick".

- This “trick” allows us to pull the gradient operator inside of the monte carlo integral and use the chain rule

$$
\begin{aligned}
\nabla_{\phi}\mathrm{ELBO}(\phi)_{\mathcal{I}} &\approx \frac{1}{S}\sum_{s=1}^{S}\nabla_{\phi}\ln\left(\frac{p(\mathbf{y}_{\mathcal{I}},\boldsymbol{\theta}=\mathbf{g}_{\phi}(\epsilon)|\mathcal{D}_{\mathcal{I}})}{q(\boldsymbol{\theta}=\mathbf{g}_{\phi}(\epsilon)|\phi)}\right)\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{s}} \\
&\approx \frac{1}{S}\sum_{s=1}^{S}\nabla_{\boldsymbol{\theta}}\ln\left(\frac{p(\mathbf{y}_{\mathcal{I}},\boldsymbol{\theta}=\mathbf{g}_{\phi}(\epsilon)|\mathcal{D}_{\mathcal{I}})}{q(\boldsymbol{\theta}=\mathbf{g}_{\phi}(\epsilon)|\phi)}\right)\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{s}}\nabla_{\phi}\mathbf{g}_{\phi}(\epsilon^{s}),
\end{aligned}
$$

$$
\text{with } \boldsymbol{\theta}^{s}_{\phi}=\mathbf{g}_{\phi}(\epsilon^{s}),\ \epsilon^{s}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\ s=1,\dots,S
$$

- Opens the door for backpropagation and thus automatic diff. 💡

“Black-box” variational inference

- Using SVI with the reparameterization gradient estimator
- Researcher only formulates a probabilistic model and provides a dataset (Kucukelbir et al. 2016), inference algo. is model “agnostic”
- What is $\mathbf{g}_\phi$?
  - Linear and non-linear choices
  - We consider linear choice

$$\boldsymbol{\theta}_j = \mathbf{L}_j \boldsymbol{\epsilon}_j + \boldsymbol{\mu}_j, \boldsymbol{\epsilon}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$
$$\boldsymbol{\theta}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{L}_j \mathbf{L}_j^{\mathrm{T}})$$

- For positive restricted parameters we need to chain another transformation layer via exp transformation
- Chaining variable transformations
  - Normalizing flows (Rezende et al. 2014)
  - Allow for expressive $\mathcal{Q}$s but are more difficult to optimize

Optimization

- We use Adam (Kingma and Ba 2014)

$$\hat{\phi}^{t} = \hat{\phi}^{t-1} + \rho_t \nabla_{\phi} \mathrm{ELBO}(\phi)_{\mathcal{I}^t} \Big|_{\phi = \hat{\phi}^{t-1}}$$

Full Algorithm

Data: $\mathcal{D}_{\text{train}}$; $\mathcal{D}_{\text{val}}$

Require: Learning rate α; stopping threshold ε; mini-batch size M; share train w; num. var. samples S; num. epochs: E

Initialize $\hat{\boldsymbol{\phi}}^0$; set $t = 1$

for $e = 1$ **to** E **do**

  $n_{\text{train}} = |\mathcal{D}_{\text{train}}| * w$

  create the mini-batches, $\mathcal{B} = \{\dots, \mathcal{I}^k, \dots\}$, $k = 1, \dots, n_{\text{train}}//M\ (+1)$

  for $k = 1$ **to** $n_{train}//M\ (+1)$ **do**

    sample noise, $\boldsymbol{\epsilon}_j^s \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $s = 1, \dots, S$, $\forall j$

    calculate approx. gradient, $\nabla_{\boldsymbol{\phi}} \text{ELBO}(\boldsymbol{\phi})_{\mathcal{I}^k}\Big|_{\boldsymbol{\phi}=\hat{\boldsymbol{\phi}}^{t-1}}$

    update variational parameters, $\hat{\boldsymbol{\phi}}^t = \hat{\boldsymbol{\phi}}^{t-1} + \rho_t \nabla_{\boldsymbol{\phi}} \text{ELBO}(\boldsymbol{\phi})_{\mathcal{I}^k}\Big|_{\boldsymbol{\phi}=\hat{\boldsymbol{\phi}}^{t-1}}$

    calculate approx. ELBO, $\text{ELBO}(\hat{\boldsymbol{\phi}}^t)_{\mathcal{D}_{\text{val}}}$

    $t = t + 1$

  end

  if $t > 200$ **then**

    $\Delta\text{ELBO} = |\text{ELBO}(\hat{\boldsymbol{\phi}}^t)_{\mathcal{D}_{\text{val}}} - \text{ELBO}(\hat{\boldsymbol{\phi}}^{t-200})_{\mathcal{D}_{\text{val}}}|$

  else

    $\Delta\text{ELBO} = \infty$

  end

  if $\Delta ELBO < \varepsilon$ **then**

    break

  end

end

Result: $\hat{\boldsymbol{\phi}}$; $\text{ELBO}(\hat{\boldsymbol{\phi}})_{\mathcal{D}_{\text{val}}}$

Algorithm 1: BBVI algorithm.

Impact Analysis

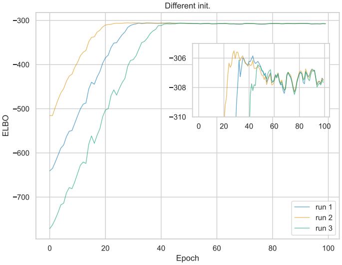


Figure 4: ELBO traces for 3 different SGD runs, using different initializations but the same seed. We use a batch size of 128, 64 samples from the variational distribution and a learning rate of 1e-2.

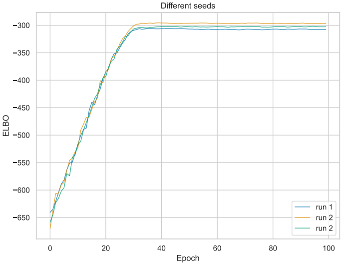
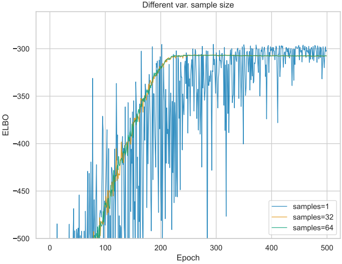


Figure 5: ELBO traces for 3 different SGD runs, using different seeds but the same initialization. Otherwise same configuration as in Figure 4.



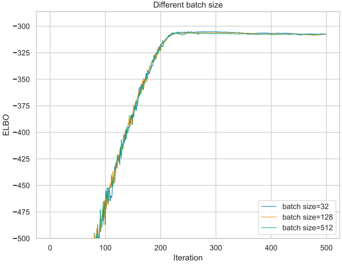


Figure 6: ELBO traces for 3 different SGD runs, using different variational sample sizes. We use batch VI with a learning rate of 1e-2 and the same seed as in Figure 4.

Figure 7: ELBO traces for 3 different SGD runs, using different batch sizes. We use a variational sample size of 64 with a learning rate of 1e-2 and the same seed as in Figure 4.

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Semiparametric distributional regression

- Not only normally distributed responses
- Linear predictors with inverse link function
- Structured addtive linear predictors so fixed and smooth effects
  - B-spline basis functions
- Augmented with priors
  - Bayesian P-splines

Application

Implementation

- Developed a small `python` package `tigerpy`, which consists of two libararies
- A model building library `tigerpy.model`
  - Construct the model
  - Uses the idea of probabilistic graphical models
- An inference library `tigerpy.bbvi`
  - Runs the inference algorithm
- Aligned with concepts found in `liesel` (Riebl et al. 2022)

Technology

When walking about the countryside of Italy, the people will not hesitate to tell you that JAX has "una anima di pura programmazione funzionale". (`JAX` docs, Bradbury et al. (2018))

- There are great things about `JAX` 😍
  - Uses a `numpy` flawored API
  - Closely follows the math (`jax.grad`)
  - Is fast (if you follow `JAX`s principles)
- There are things that cause headaches 🥴
  - Pure functions
  - Tracing
  - Efficiency considerations in JIT-compiled code

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

- `tigerpy.model` constructs under the hood a DAG
- Employs the `networkx` package for constructing, traversing and visualizing the DAG

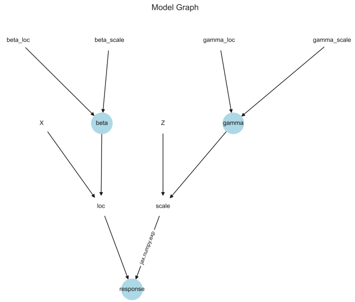


Figure 8: The DAG visualization for location-scale regression from the method `.visualize_graph()`.

Simulation Studies

Open Problems

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe (Apr. 2017). "Variational Inference: A Review for Statisticians". In: *Journal of the American Statistical Association* 112.518, pp. 859–877.

Bradbury, James, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang (2018). *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. URL: http://github.com/google/jax.

Hoffman, Matt, David M. Blei, Chong Wang, and John Paisley (2012). *Stochastic Variational Inference*.

Jordan, Michael I., Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul (1999). "An Introduction to Variational Methods for Graphical Models". In: *Machine Learning* 37.2, pp. 183–233.

Kingma, Diederik P and Max Welling (2013). *Auto-Encoding Variational Bayes*.

Kingma, Diederik P. and Jimmy Ba (2014). *Adam: A Method for Stochastic Optimization*.

Kucukelbir, Alp, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M. Blei (2016). *Automatic Differentiation Variational Inference*.

Kullback, S. and R. A. Leibler (Mar. 1951). “On Information and Sufficiency”. In: *The Annals of Mathematical Statistics* 22.1, pp. 79–86.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra (2014). *Stochastic Backpropagation and Approximate Inference in Deep Generative Models*.

Riebl, Hannes, Paul F. V. Wiemann, and Thomas Kneib (2022). *Liesel: A Probabilistic Programming Framework for Developing Semi-Parametric Regression Models and Custom Bayesian Inference Algorithms*.

Wainwright, Martin J. and Michael I. Jordan (2007). “Graphical Models, Exponential Families, and Variational Inference”. In: *Foundations and Trends® in Machine Learning* 1.1–2, pp. 1–305.