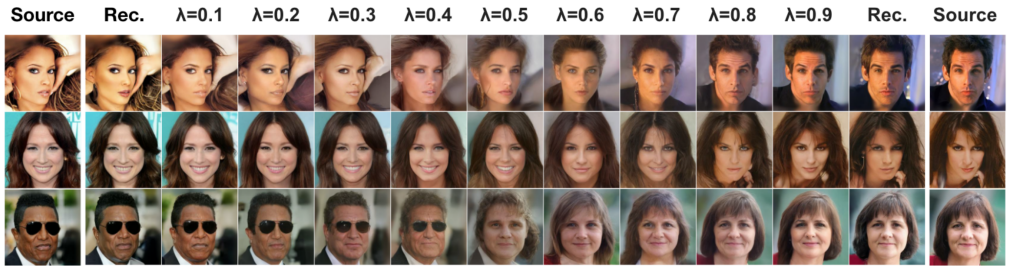
Advanced Bayesian Methodology

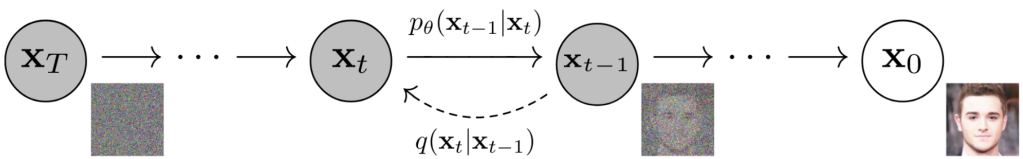
Part I: Introduction

Welcome

Recent advances in representation learning, generative AI...



...inherit ideas from Bayesian statistics and probabilistic modeling.

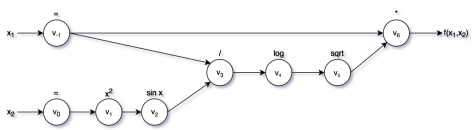


(Figures from "Denoising Diffusion Probabilistic Models" Ho et al. [2020].)

Recent advances in scalable and generic Bayesian inference...

Stan PYRO PyMC3

...are built on the machinery of the "deep learning revolution".



This course is about that **two-way street**.

Welcome

Goal: From classical variational inference to diffusion in 9 weeks!

Course description (revised):

> **This course will** ~~*survey advanced topics in model-building, model-fitting, and model-checking in Bayesian analysis. One part will focus on building models with stochastic processes, including point processes and Bayesian nonparametric priors. Another part*~~ **will focus on modern techniques for scalable posterior inference, including stochastic variational inference and variational auto-encoders.** ~~*A final part (time permitting) will focus on the theory and practice of model critiquing via predictive checks and its role within the full Bayesian workflow.*~~

Why the revisions?

- so much exciting stuff in recent generative AI
- it's not a big jump from variational auto-encoders (2014) to diffusion (2020)
- BNP will still be there when we get back

Statisticians have lots to gain from and contribute to contemporary ML/AI.

Let's dive in together!

Criteria

Prerequisites

- graduate course(s) in Bayesian stats (e.g., STAT 348)
- *passing* familiarity with core machine learning concepts (e.g., gradient descent, neural networks)
- some experience with Python

Attitude

- excitement about this research area
- desire to write a paper / work on a project in this domain
- game to be lively and involved in seminar

Also

- alignment with your research is terrific
- diverse and nontraditional students welcome

Format

Tone

- fun and fast paced
- technical and creative
- positive and inclusive

Seminar

- journal club-style: everybody reads, everybody participates
- RealDeal$^{(TM)}$ rules: no phones, no email, no text, etc; tablets ok

Reading

- 2(ish) papers per week (weeks are themed)
- work hard to understand; you likely have to find / read other materials
- 1 reader report per week (due Thursday before seminar)

Presentation

- you prepare and lead the seminar (slides, whiteboard, etc.)
- prepare hard but be transparent/open about uncertainty/questions
- optional (encouraged): pre-meetings with me to take me through it

Project...

Project

Purpose

- learn / understand concepts by applying them
- familiarize with modern code frameworks (e.g., PyTorch)
- contribute to your own research

Requirements (loose)

- project report with style / content of an ML conference paper
  (intro, lit review, model/method, implementation, results, figures, discussion...)
- meaningful software implementation
- *some* kind of novel contribution
  - new application (e.g., your own interesting data set)
  - new theory or theoretical perspective
  - new implementation (e.g., in a different language)
  - new tutorial / survey
  - new method / model (of course)

Structure

- solo or teams of two (expectations higher, contributions must be separate)
- meet with me periodically to discuss / refine project
- office hours on Thursdays 2-3pm (does that work?)

Topics & (preliminary) schedule

Week	Theme	Day	Reading(s)	Content
1	Preliminaries	Tue 10/26	n/a	Introduction, review, overview
		Thu 10/28	Blei et al. [2017]	Classical VI
2	Black box VI	Tue 10/3	Ranganath et al. [2014]	Black box VI
		Thu 10/5	Mnih and Gregor [2014]	Neural VI
3	VAEs	Tue 10/10	Kingma and Welling [2014]	Variational autoencoders
		Thu 10/12	Rezende et al. [2014]	Variational autoencoders
4	Dying units	Tue 10/17	Burda et al. [2016]	Importance-weighted VAEs
		Thu 10/19	Tomczak and Welling [2018]	VAEs with a VampPrior
5	Structure in Q	Tue 10/24	Rezende and Mohamed [2015]	Normalizing flows
		Thu 10/26	Kingma et al. [2016]	Normalizing flows
6		Tue 10/31	Sønderby et al. [2016]	Ladder networks
		Thu 11/2	Ranganath et al. [2016]	Hierarchical variational models
7	Amortization gap	Tue 11/7	Cremer et al. [2018]	Amortization gap
		Thu 11/9	Kim et al. [2018]	Semi-amortized VAEs
8	Diffusion	Tue 11/14	Ho et al. [2020]	Diffusion
		Thu 11/16	Kingma et al. [2021]	Diffusion
9	TBD	Tue 11/28	TBD	TBD
		Thu 11/30	TBD	TBD

- it's okay if these terms are unfamiliar right now...
- presentations start next week (volunteers!)
- signup form: https://tinyurl.com/stat451signup

Part II: Review

Bayesian modeling

A probabilistic model is a joint distribution

$$P(\boldsymbol{x}, \boldsymbol{z}) = P(\boldsymbol{x} \mid \boldsymbol{z})\, P(\boldsymbol{z}) \tag{1}$$

that relates latent variables $\boldsymbol{z} = z_{1:m}$ to data $\boldsymbol{x} = x_{1:n}$.

The posterior distribution characterizes our uncertainty about the latent variables

$$P(\boldsymbol{z} \mid \boldsymbol{x}) = \frac{\overbrace{P(\boldsymbol{x} \mid \boldsymbol{z})}^{\text{likelihood}}\ \overbrace{P(\boldsymbol{z})}^{\text{prior}}}{\underbrace{P(\boldsymbol{x})}_{\text{evidence}}} \tag{2}$$

This is the target of Bayesian inference.

Bayesian modeling

Usually we use our knowledge of the data to *prescribe* a likelihood and prior... e.g.,

$$P(\boldsymbol{z}) = \prod_{j=1}^{m} \mathcal{N}(z_j; \cdots) \qquad P(\boldsymbol{x} \mid \boldsymbol{z}) = \prod_{i=1}^{n} \mathcal{N}(x_i; f(\boldsymbol{z}), \cdots) \tag{3}$$

The likelihood and prior are both known and analytically *tractable*...

...however the posterior may or may not be tractable...

$$P(\boldsymbol{z} \mid \boldsymbol{x}) = \frac{\overbrace{P(\boldsymbol{x} \mid \boldsymbol{z})}^{\text{likelihood}} \overbrace{P(\boldsymbol{z})}^{\text{prior}}}{\underbrace{\int dz_1 \cdots \int dz_m \, P(\boldsymbol{x} \mid \boldsymbol{z}) \, P(\boldsymbol{z})}_{= P(\boldsymbol{x}) \text{ evidence}}} \tag{4}$$

...because the evidence is a (potentially high-dimensional) integral (or sum).

Posterior inference is often approximate, as a result.

Conjugate models

Conjugate models are some of the (few) special cases where the posterior is tractable.

Example: beta-Bernoulli

$$z \sim \text{Beta}(a_0, b_0) \tag{5}$$
$$x_i \sim \text{Bern}(p) \qquad \text{for } i = 1 \dots n \tag{6}$$

The posterior is tractable:

$$P(z \mid \boldsymbol{x}) = \text{Beta}\left(a_0 + \sum_{i=1}^{n} x_i,\ b_0 + \sum_{i=1}^{n} (1 - x_i)\right) \tag{7}$$

Why? Exponential families are special.
(Other examples: gamma–Poisson, Dirichlet–multinomial, normal–normal, . . .)

Conjugate models are necessarily simple; most models are not conjugate.

Sidebar: Sampling notation

This is sampling notation:

$$z \sim \text{Beta}(a_0, b_0) \tag{8}$$

$$x_i \sim \text{Bern}(z) \qquad \text{for } i = 1 \dots n \tag{9}$$

Sometimes repeated (independent / iid) sampling might be written as:

$$x_i \overset{\text{iid}}{\sim} \text{Bern}(z) \tag{10}$$

It means the same thing as:

$$P(\boldsymbol{x}, z) = \text{Beta}(z;\, a_0, b_0) \prod_{i=1}^{n} \text{Bern}(x_i;\, z) \tag{11}$$

Sidebar: (Hyper)parameters

Notice that the distribution depends on a_0 and b_0 (which do not appear on the LHS):

$$P(\boldsymbol{x}, z) = \text{Beta}(z;\, a_0, b_0) \prod_{i=1}^{n} \text{Bern}(x_i;\, z) \tag{12}$$

Sometimes we make explicit conditioning on (hyper)parameters—e.g.,

$$P_\theta(\boldsymbol{x}, z) \equiv P(\boldsymbol{x}, z) \qquad \text{where } \theta \equiv \{a_0, b_0\} \tag{13}$$

Distinguishing parameters θ from latent variables $\boldsymbol{z}$ will become important.

Q: What is the difference between (hyper)parameters and latent variables?

A: It's what we do seek to do with them:

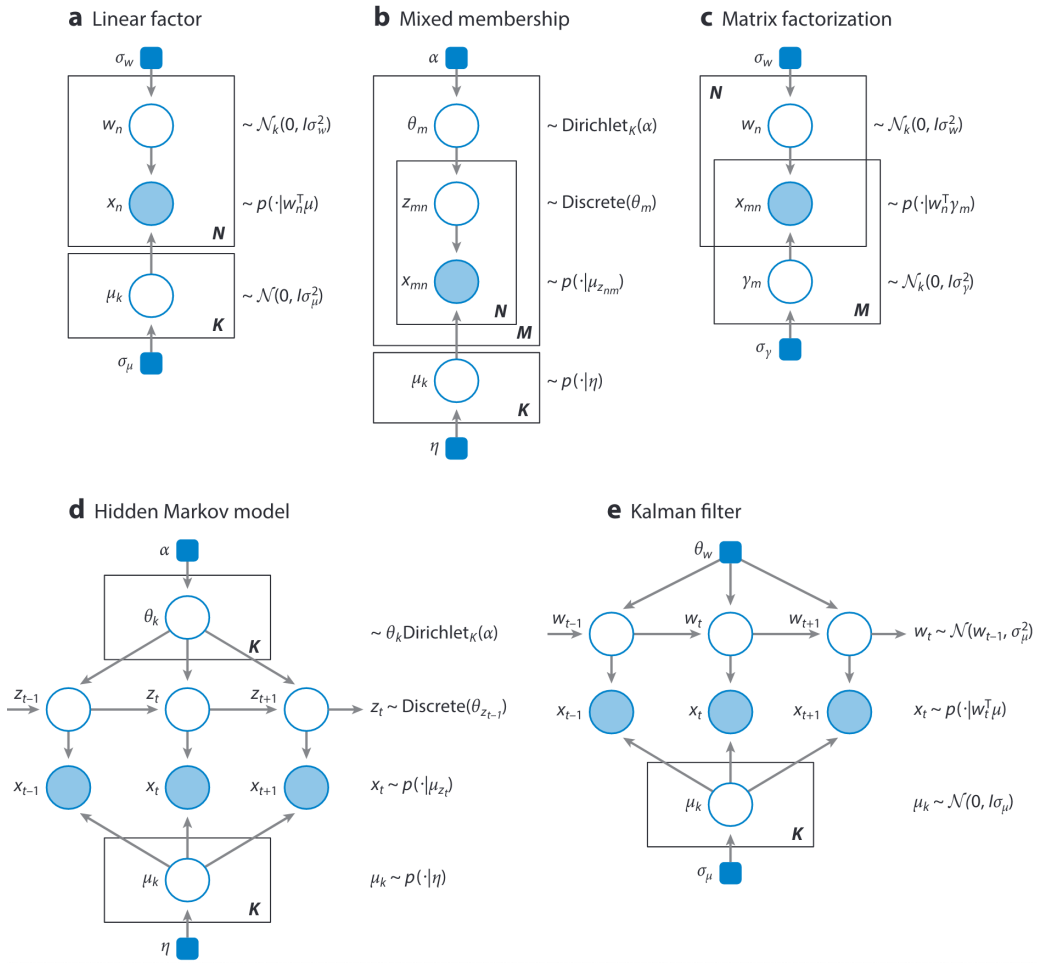
- we *infer* latent variables $P(\boldsymbol{z} \mid \boldsymbol{x})$
- we *learn* parameters $\hat{\theta} \leftarrow \text{argmax}_\theta P_\theta(\boldsymbol{x})$
- we *set* (or *tune*) hyperparameters

(Not everyone will follow this definition; be warned.)

Probabilistic graphical models (PGMs)

As our models get more complicated, it is useful to have a (visual) language for composing them from repurposable and modular units... enter PGMs!

Figure from [Blei, 2014]:



A PGM encodes the conditional independencies in a joint distribution.

Probabilistic graphical models (PGMs)

Example: Bayesian mixture model

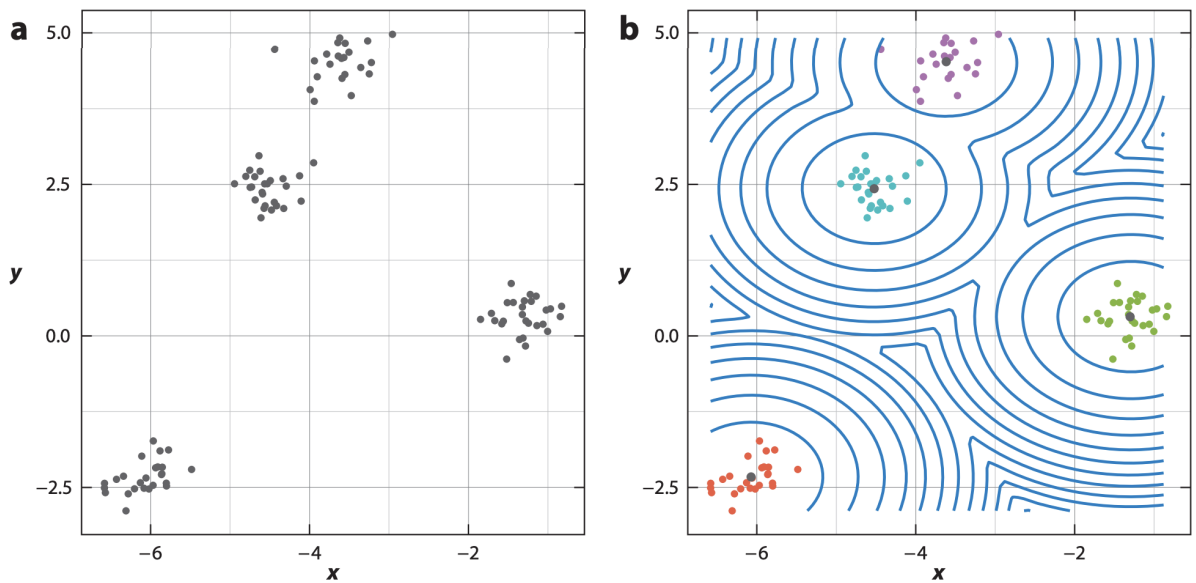
$$\boldsymbol{\pi} \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_K) \tag{14}$$

$$\mu_k \sim \mathcal{N}(\mathbf{0}, \mathrm{I}) \qquad \text{for } k = 1 \ldots K \tag{15}$$

$$z_i \sim \mathrm{Cat}(\boldsymbol{\pi}) \qquad \text{for } i = 1 \ldots n \tag{16}$$

$$x_i \sim \mathcal{N}(\mu_{z_i}, \sigma^2) \qquad \text{for } i = 1 \ldots n \tag{17}$$

Figure from [Blei, 2014] (ignore the x- and y-labels):



Volunteer: Draw the graphical model (board)

Conditional conjugacy

The Bayesian mixture model is not (fully) conjugate; the full posterior is intractable:

$$P(\boldsymbol{z},\, \boldsymbol{\pi}, \boldsymbol{\mu} \mid \boldsymbol{x}) = \; ? \tag{18}$$

However, all of the *complete conditionals* are tractable—e.g.,

$$\begin{aligned} P(\mu_k \mid -) &\equiv P(\mu_k \mid \boldsymbol{x},\, \boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}_{\setminus k}) && (19)\\ &= \mathcal{N}(\mu_k; \cdots) && (20) \end{aligned}$$

$$\begin{aligned} P(z_i \mid -) &\equiv P(z_i \mid \boldsymbol{x},\, \boldsymbol{z}_{\setminus i}, \boldsymbol{\pi}, \boldsymbol{\mu}) && (21)\\ &= \mathrm{Cat}(z_i; \cdots) && (22) \end{aligned}$$

Why? Because all edges in the PGM encode conditionally conjugate relationships.

Closed-form complete conditionals play an important role in approximate inference.

Gibbs sampling

Returning to the more abstract notation where $\boldsymbol{z} \equiv z_{1:m}$ denote all latent variables.

A very simple example of approximate inference that benefits from closed-form complete conditionals is *Gibbs sampling*.

repeat
  for latent variable $j = 1$ to m **do**
    $z_j \sim P(z_j \mid \boldsymbol{z}_{\setminus j}, \boldsymbol{x})$ re-sample from complete conditional
  end for
until compute budget is exhausted

With a large enough budget, this will eventually generate samples from the exact posterior $z_j \sim P(z_j \mid \boldsymbol{x})$.

MCMC

Gibbs sampling is a special case of Markov chain Monte Carlo (MCMC).

We run a Markov chain $\ldots \boldsymbol{z}_{s-1}, \boldsymbol{z}_s, \boldsymbol{z}_{s+1} \ldots$ with transition operator $\mathcal{T}(\boldsymbol{z}_s; \boldsymbol{z}_{s-1})$

$$\Pr(\boldsymbol{z}_s) = \int d\boldsymbol{z}_{s-1} \mathcal{T}(\boldsymbol{z}_s; \boldsymbol{z}_{s-1}) \Pr(\boldsymbol{z}_{s-1}) \tag{23}$$

such that the stationary distribution is the exact posterior $\Pr(\boldsymbol{z}_\infty) = P(\boldsymbol{z} \mid \boldsymbol{x})$

MCMC is traditionally (and still) the main workhorse in Bayesian statistics...

- principled
- asymptotically exact

However, it is practically difficult to scale MCMC up to large data sets...

- inherently serial / sequential (hard to parallelize)
- guarantees are based on stationarity (tricky to sub-sample data)
- requires many samples in high dimensions

(Note: Scalable MCMC is an active research area.)

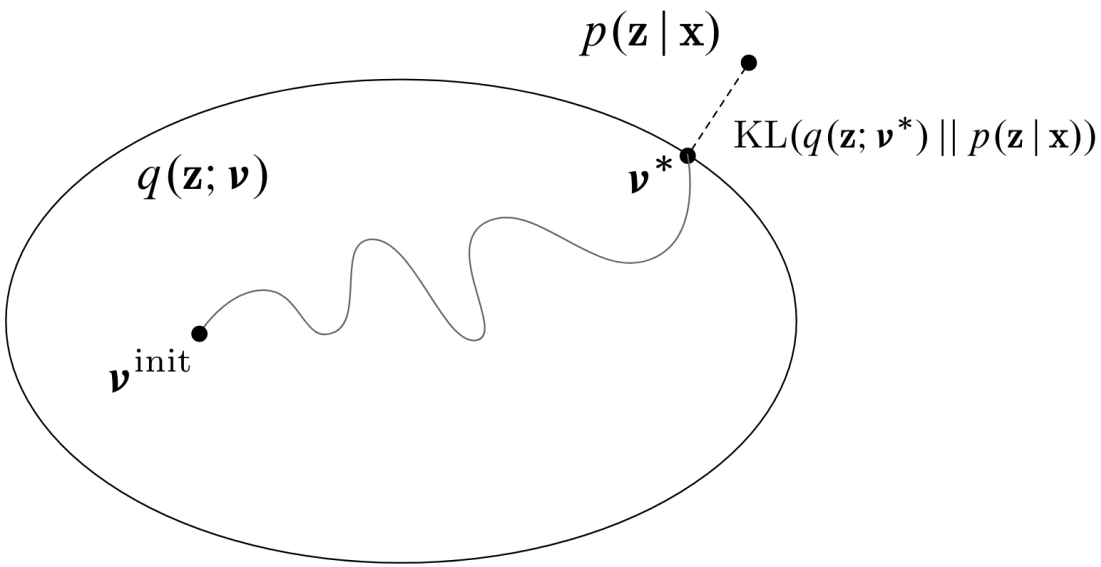
Variational inference

Define the *variational family*: $q(\boldsymbol{z}; \boldsymbol{\nu})$.
(A family of probability distributions over $\boldsymbol{z}$, parameterized by ν.)

Now find the member of that family $q_{\nu^\star}(\boldsymbol{z})$ such that

$$\nu^\star \leftarrow \text{argmin}_\nu \, \text{KL}\Big(q(\boldsymbol{z}; \boldsymbol{\nu}) \;||\; p(\boldsymbol{z} \mid \boldsymbol{x})\Big) \tag{24}$$

Figure from [Blei et al., 2016]:



Variational inference **turns posterior inference into optimization**.

Variational inference

How do you minimize the KL divergence to *a distribution you cannot form*?

$$\mathrm{KL}\Big(q(\boldsymbol{z};\,\boldsymbol{\nu})\;||\;p(\boldsymbol{z}\mid\boldsymbol{x})\Big) = \mathbb{E}_{q(\boldsymbol{z};\,\boldsymbol{\nu})}\left[\log\frac{q(\boldsymbol{z};\,\boldsymbol{\nu})}{p(\boldsymbol{z}\mid\boldsymbol{x})}\right] \tag{25}$$

$$= \mathbb{E}_{q(\boldsymbol{z};\,\boldsymbol{\nu})}\left[\log\frac{q(\boldsymbol{z};\,\boldsymbol{\nu})}{p(\boldsymbol{z},\boldsymbol{x})}\right] + \underbrace{\log p(\boldsymbol{x})}_{\text{(log) evidence}} \tag{26}$$

$$\propto_{\nu} \mathbb{E}_{q(\boldsymbol{z};\,\boldsymbol{\nu})}\left[\log\frac{q(\boldsymbol{z};\,\boldsymbol{\nu})}{p(\boldsymbol{z},\boldsymbol{x})}\right] \tag{27}$$

Re-arranging terms reveals the *evidence lower bound (ELBO)*:

$$\underbrace{\log p(\boldsymbol{x})}_{\text{(log) evidence}} = \underbrace{\mathbb{E}_{q(\boldsymbol{z};\,\boldsymbol{\nu})}\left[\log\frac{p(\boldsymbol{z},\boldsymbol{x})}{q(\boldsymbol{z};\,\boldsymbol{\nu})}\right]}_{\text{evidence lower bound (ELBO)}} + \underbrace{\mathrm{KL}\Big(q(\boldsymbol{z};\,\boldsymbol{\nu})\;||\;p(\boldsymbol{z}\mid\boldsymbol{x})\Big)}_{\geq 0} \tag{28}$$

Maximizing the ELBO (which we can form / compute) $\equiv$ minimizing the KL.

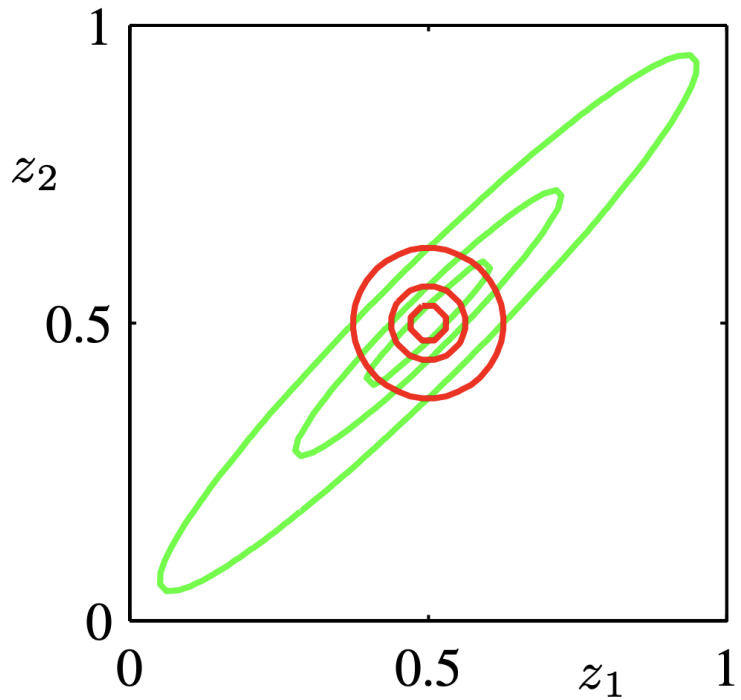
"Classical" variational inference

Choose a factorized or "mean-field" variational family

$$q(\boldsymbol{z}; \boldsymbol{\nu}) = \prod_{j=1}^{m} q(z_j; \nu_j) \tag{29}$$

Every latent variable z_j is governed by its own $q(z_j; \nu_j)$, with its own ν_j.

Figure from [Bishop, 2006, Chap. 10]: Green is exact posterior $P(z_1, z_2 \mid \boldsymbol{x})$. Red is the best-fit mean-field approximation. Notice that there is no correlation between z_1 and z_2.



Very simple variational family facilitates optimization...

"Classical" variational inference

Bishop [2006]: For mean-field, we maximize the ELBO with respect to ν_j by setting it to

$$q(z_j; \nu_j^\star) \propto \exp\left(\mathbb{E}_{q(\boldsymbol{z}_{\setminus j}; \boldsymbol{\nu}_{\setminus j})}\left[\log \underbrace{p(z_j \mid \boldsymbol{z}_{\setminus j}, \boldsymbol{x})}_{\text{complete conditional}}\right]\right) \tag{30}$$

When the model is conditionally conjugate, this is available in closed form...

$$q(z_j; \nu_j^\star) = p\left(z_j \mid f^{-1}(\mathbb{E}_q[f(\boldsymbol{z}_{\setminus j})]), \boldsymbol{x}\right) \tag{31}$$

...and is in the same family as the prior (again due to exponential family magic).

Mean-field coordinate ascent variational inference (MF-CAVI):

repeat
  for latent variable $j = 1$ to m **do**
    $q(z_j; \nu_j^\star) \leftarrow p\left(z_j \mid f^{-1}(\mathbb{E}_q[f(\boldsymbol{z}_{\setminus j})])\right)$ optimal coordinate-wise update
  end for
until convergence

(Looks a lot like Gibbs sampling. Also looks a lot like EM. Connections to both.)

"Classical" variational inference

MF-CAVI works very well in practice (my experience)

- faster than MCMC
- good point estimates

Still...

- requires a pass over the entire data set
- specific to conditionally conjugate models

One remedy is stochastic variational inference (SVI) [Hoffman et al., 2013]

- recasts MF-CAVI as performing natural gradient ascent
- performs stochastic gradient ascent by sub-sampling data
- however, still tailored (e.g., conditional conjugacy)

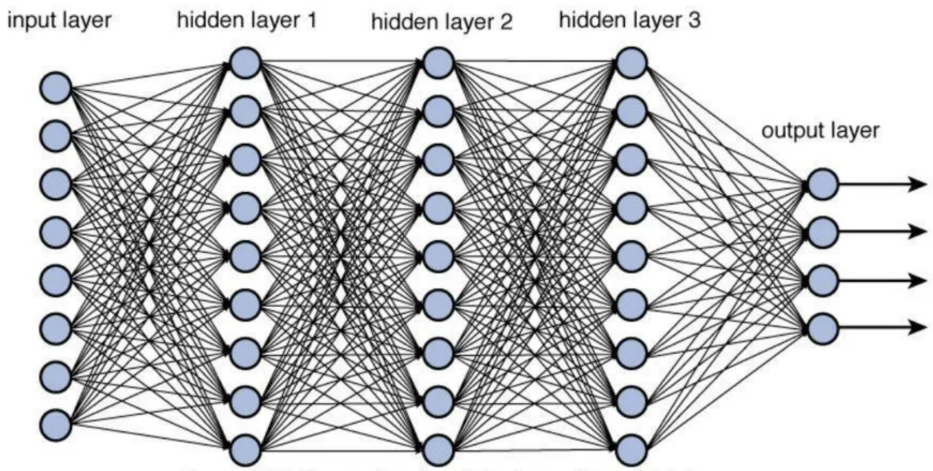
This class: recruit modern ML tools to make VI

- generic (to many models, simple and complex)
- scalable (to large data sets)

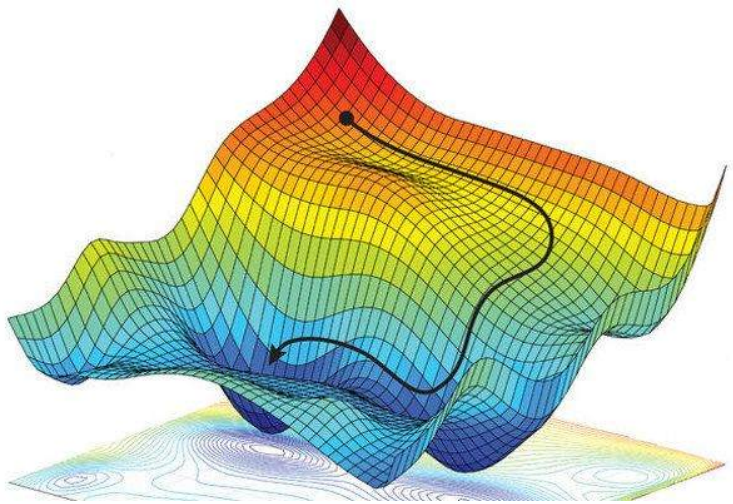
Part III: Preview

Deep learning revolution (2013–present)

Massive (renewed) interest in neural networks, deep learning, large models, etc.



Fueled (in large part) by stochastic optimization and automatic differentiation.



Automatic differentiation

Automatic differentiation (AD) is **not**

- numerical differentiation

$$\frac{\nabla f(\boldsymbol{x})}{\nabla x_i} \approx \frac{f(\boldsymbol{x} + \delta \boldsymbol{e}_i) - f(\boldsymbol{x})}{\delta}$$

- symbolic differentiation ("just the chain rule"[1])

Automatic differentiation (AD) is

> *a specific family of techniques that* **compute derivatives through accumulation of values during code execution** *to generate numerical derivative evaluations rather than derivative expressions [Baydin et al., 2018]*

Rapid development of frameworks for automatic differentiation since 2013(ish).

Takeaway: to evaluate gradients $\nabla_\theta f(\theta)$, a practitioner need only implement $f(\theta)$ (conditions on $f(\cdot)$ apply)

[1] "Backprop is not just the chain rule" by Tim Vieira (2017)
https://timvieira.github.io/blog/post/2017/08/18/backprop-is-not-just-the-chain-rule/

Modernizing Bayesian inference

Automatic differentiation (and other tools from modern ML) have been imported into Bayesian inference to make it more *scalable*, *generic*, and *accurate*.

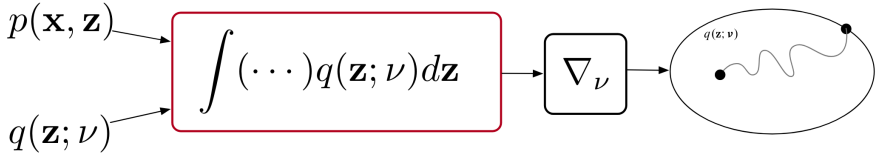
Revolution in MCMC: Hamiltonian/hybrid Monte Carlo (HMC) and the No U-Turn Sampler (NUTS) [Hoffman and Gelman, 2014] are the defaults in libraries like Stan, Pyro, PyMC3. (We will not cover MCMC in this class.)

Revolution in VI: Variational inference has also been transformed by these tools...

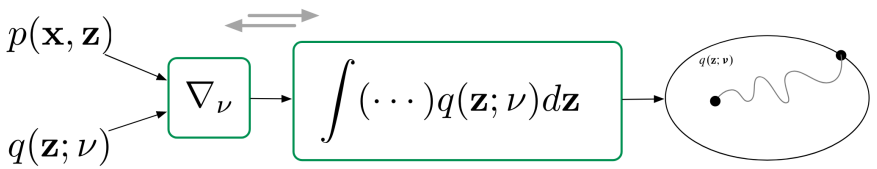
From classical to contemporary VI

Slides from Blei et al. [2016]:

The Problem in the Classical VI Recipe



The New VI Recipe



Use stochastic optimization!

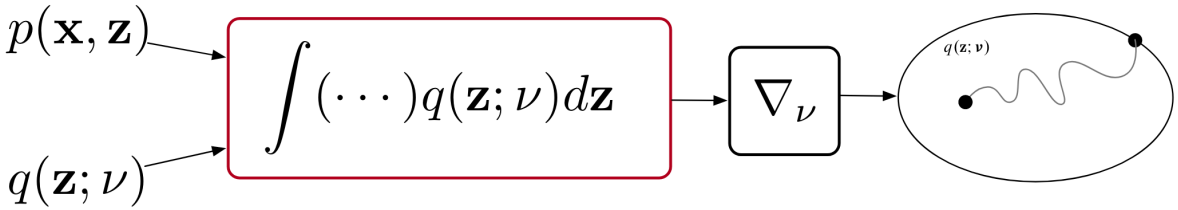
From classical to contemporary VI

MF-CAVI ("classical VI") implicitly maximizes the evidence lower bound:

$$\mathcal{L}(\boldsymbol{\nu}) \triangleq \mathbb{E}_{q(\boldsymbol{z};\,\boldsymbol{\nu})}\left[\log \frac{p(\boldsymbol{z},\boldsymbol{x})}{q(\boldsymbol{z};\,\boldsymbol{\nu})}\right] = \int d\boldsymbol{z}\, q(\boldsymbol{z};\,\boldsymbol{\nu})\left[\log \frac{p(\boldsymbol{z},\boldsymbol{x})}{q(\boldsymbol{z};\,\boldsymbol{\nu})}\right] \tag{32}$$

We need a closed form for the ELBO for this to work.

The Problem in the Classical VI Recipe



From classical to contemporary VI

Consider the gradient of the ELBO with respect to the variational parameters

$$\nabla_{\boldsymbol{\nu}} \mathcal{L}(\boldsymbol{\nu}) = \nabla_{\boldsymbol{\nu}} \, \mathbb{E}_{q(\boldsymbol{z};\, \boldsymbol{\nu})} \left[\log \frac{p(\boldsymbol{z}, \boldsymbol{x})}{q(\boldsymbol{z};\, \boldsymbol{\nu})} \right] = \mathbb{E}_{q(\boldsymbol{z};\, \boldsymbol{\nu})} \left[\underbrace{\nabla_{\boldsymbol{\nu}} \log q(\boldsymbol{z};\, \boldsymbol{\nu})}_{\text{score function}} \log \frac{p(\boldsymbol{z}, \boldsymbol{x})}{q(\boldsymbol{z};\, \boldsymbol{\nu})} \right] \tag{33}$$

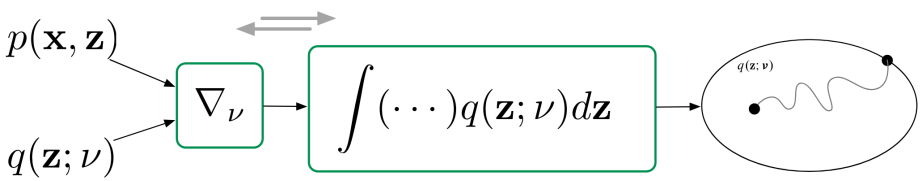
Approximate expectation with Monte Carlo by sampling $\boldsymbol{z}_s \sim q(\boldsymbol{z}_s;\, \boldsymbol{\nu})$

$$\approx \frac{1}{S} \sum_{s=1}^{S} \nabla_{\boldsymbol{\nu}} \log q(\boldsymbol{z}_s;\, \boldsymbol{\nu}) \, \log \frac{p(\boldsymbol{z}_s, \boldsymbol{x})}{q(\boldsymbol{z}_s;\, \boldsymbol{\nu})} \tag{34}$$

This is the *score function (aka REINFORCE) gradient estimator*.

This is the big idea with "black box" VI [Ranganath et al., 2014, Mnih and Gregor, 2014] (**week 2**) and much of contemporary VI.

The New VI Recipe



Use stochastic optimization!

From classical to contemporary VI

The score function estimator for the gradient of the ELBO:

$$\widehat{\nabla_{\boldsymbol{\nu}} \mathcal{L}(\boldsymbol{\nu})} = \frac{1}{S} \sum_{s=1}^{S} \nabla_{\boldsymbol{\nu}} \log q(\boldsymbol{z}_s; \boldsymbol{\nu}) \log \frac{p(\boldsymbol{z}_s, \boldsymbol{x})}{q(\boldsymbol{z}_s; \boldsymbol{\nu})}, \text{ where } \boldsymbol{z}_s \sim q(\boldsymbol{z}_s; \boldsymbol{\nu}) \tag{35}$$

We use it to perform stochastic gradient ascent:

$$\boldsymbol{\eta}_{t+1} \leftarrow \boldsymbol{\eta}_t + \rho_t \widehat{\nabla_{\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\eta}_t)} \tag{36}$$

How does this improve over MF-CAVI?

1. we no longer require a closed-form ELBO
   - this means $p(\boldsymbol{z}, \boldsymbol{x})$ can be *complex* (only need to evaluate it)
   - this means $q(\boldsymbol{z}; \boldsymbol{\nu})$ can be *complex* (only need to evaluate it, sample from it)
2. inference is *generic*
   - practitioner only needs to supply a function for $p(\boldsymbol{z}, \boldsymbol{\nu})$ and $q(\boldsymbol{z}; \boldsymbol{\nu})$
   - (let automatic differentiation do the rest)
3. (stochastic) optimization is *scalable*
   - amenable to GPU-accelerated implementation
   - for most models, we can further sub-sample the data

These reasons combine to rise *deep generative models*...

Variational autoencoders

VAEs [Kingma and Welling, 2014, Rezende et al., 2014] (**weeks 3–4**) naturally follow...

The probabilistic model $p_\theta(\boldsymbol{x}, \boldsymbol{z})$:

$$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i;\, \mathbf{0}, \mathrm{I}) \tag{37}$$

$$\underbrace{p_\theta(\boldsymbol{x}_i \mid \boldsymbol{z}_i)}_{\text{decoder}} = \mathcal{N}\Big(\boldsymbol{x}_i;\, \mu_\theta(\boldsymbol{z_i}),\, \sigma_\theta(\boldsymbol{z_i})\Big) \tag{38}$$

$\mu_\theta(\cdot)$ and $\sigma_\theta(\cdot)$ come from a neural network with weights θ.

The variational family

$$\underbrace{q_\phi(\boldsymbol{z}_i \mid \boldsymbol{x}_i)}_{\text{encoder}} = \mathcal{N}\Big(\boldsymbol{z}_i;\, \mu_\phi(\boldsymbol{x}_i),\, \sigma_\phi(\boldsymbol{x}_i)\Big) \tag{39}$$

$\mu_\phi(\cdot)$ and $\sigma_\phi(\cdot)$ come from a neural network with weights ϕ.

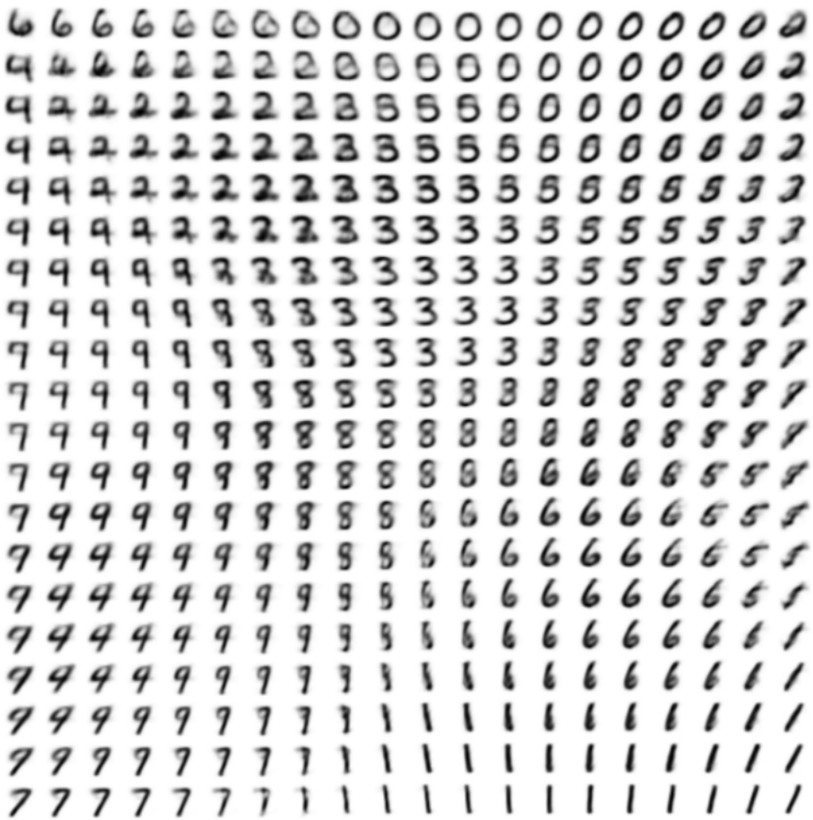
Fit both θ and ϕ with stochastic gradient ascent on the ELBO.

Variational autoencoders

Slide from John Cunningham's "Deep probabilistic models" course:

Learn the autoencoder and then:

- Choose a point z_i in latent space (not drawing from the posterior!)
- Decode this point with $x_i \sim p_\theta(x_i|z_i)$:



Learns a manifold of simple images and how to generate...

Re-injecting structure

Black box (and other forms) of gradient-based VI let us use complex variational families.

$q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ need not be a *complete* black box though...

- normalizing flows [Rezende and Mohamed, 2015, Kingma et al., 2016] (**week 5**)
- hierarchical families [Sønderby et al., 2016, Ranganath et al., 2016] (**week 6**)
- many, many more ways...
- this will build up to diffusion model [Ho et al., 2020, Kingma et al., 2021] (**week 8**)

Moreover, $p_\theta(\boldsymbol{z}, \boldsymbol{x})$ need not be a black box either...

- it can simply be a parametric/prescribed model
- using $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ is more generally called *amortized VI* (**week 7**);
  we will learn about the amortization gap [Cremer et al., 2018] and semi-amortized approaches [Kim et al., 2018]

Combining the scalable/generic machinery of modern ML/AI with
the structure and interpretability of Bayesian modeling is our underlying theme.

Topics & (preliminary) schedule (again)

Week	Theme	Day	Reading(s)	Content
1	Preliminaries	Tue 10/26	n/a	Introduction, review, overview
		Thu 10/28	Blei et al. [2017]	Classical VI
2	Black box VI	Tue 10/3	Ranganath et al. [2014]	Black box VI
		Thu 10/5	Mnih and Gregor [2014]	Neural VI
3	VAEs	Tue 10/10	Kingma and Welling [2014]	Variational autoencoders
		Thu 10/12	Rezende et al. [2014]	Variational autoencoders
4	Dying units	Tue 10/17	Burda et al. [2016]	Importance-weighted VAEs
		Thu 10/19	Tomczak and Welling [2018]	VAEs with a VampPrior
5	Structure in Q	Tue 10/24	Rezende and Mohamed [2015]	Normalizing flows
		Thu 10/26	Kingma et al. [2016]	Normalizing flows
6		Tue 10/31	Sønderby et al. [2016]	Ladder networks
		Thu 11/2	Ranganath et al. [2016]	Hierarchical variational models
7	Amortization gap	Tue 11/7	Cremer et al. [2018]	Amortization gap
		Thu 11/9	Kim et al. [2018]	Semi-amortized VAEs
8	Diffusion	Tue 11/14	Ho et al. [2020]	Diffusion
		Thu 11/16	Kingma et al. [2021]	Diffusion
9	TBD	Tue 11/28	TBD	TBD
		Thu 11/30	TBD	TBD

- presentations start next week (volunteers!)
- signup form: https://tinyurl.com/stat451signup

References I

Atılım Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic Differentiation in Machine Learning: A Survey. *Journal of Machine Learning Research*, 18, 2018.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2.

David M Blei. Build, Compute, Critique, Repeat: Data Analysis with Latent Variable Models. *Annual Review of Statistics and Its Application*, 1(1):203–232, January 2014. ISSN 2326-8298, 2326-831X. doi: 10.1146/annurev-statistics-022513-115657.

David M Blei, Rajesh Ranganath, and Shakir Mohamed. Variational Inference: Foundations and Modern Methods (NeurIPS Tutorial), 2016.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, April 2017. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.2017.1285773.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance Weighted Autoencoders. In *Proceedings of the International Conference on Learning Representations*, 2016.

Chris Cremer, Xuechen Li, and David Duvenaud. Inference Suboptimality in Variational Autoencoders. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, 2020.

Matthew D Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15, 2014.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic Variational Inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

Yoon Kim, Sam Wiseman, Andrew C Miller, David Sontag, and Alexander M Rush. Semi-Amortized Variational Autoencoders. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

References II

Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *Proceedings of the International Conference on Learning Representations*, 2014.

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving Variational Inference with Inverse Autoregressive Flow. In *Advances in Neural Information Processing Systems*, 2016.

Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational Diffusion Models. In *Advances in Neural Information Processing Systems*, 2021.

Andriy Mnih and Karol Gregor. Neural Variational Inference and Learning in Belief Networks. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Rajesh Ranganath, Sean Gerrish, and David M Blei. Black Box Variational Inference. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, 2014.

Rajesh Ranganath, Dustin Tran, and David M Blei. Hierarchical Variational Models. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder Variational Autoencoders. In *Advances in Neural Information Processing Systems*, 2016.

Jakub M Tomczak and Max Welling. VAE with a VampPrior. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, 2018.