# An Intro to Bayesian Inference

Ludwig Winkler

Machine Learning Group
TU Berlin

March 30, 2020

# Outline

Bayesian Machine Learning

Sampling
- MCMC
- Stochastic Gradient MCMC
- Hamiltonian Monte Carlo

Variational Inference

Bayesian Deep Learning

# Why Bayesian Machine Learning?

Bayesian ML generalizes Deterministic ML

# Why Bayesian Machine Learning?

Bayesian ML generalizes Deterministic ML

- Gaussian Process $>$ Kernel Ridge Regression
- Bayesian Neural Networks $>$ Neural Networks
- Variational AutoEncoder $>$ AutoEncoder

# Why Bayesian Machine Learning?

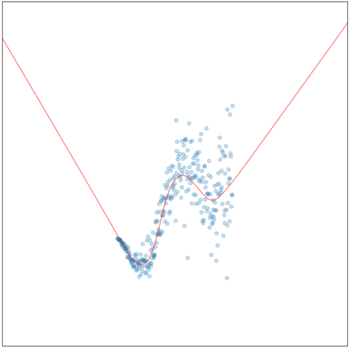
Bayesian ML generalizes Deterministic ML

| | |
|---|---|
| ◦ Gaussian Process | $>$ Kernel Ridge Regression |
| ◦ Bayesian Neural Networks | $>$ Neural Networks |
| ◦ Variational AutoEncoder | $>$ AutoEncoder |

Bayesian ML offers

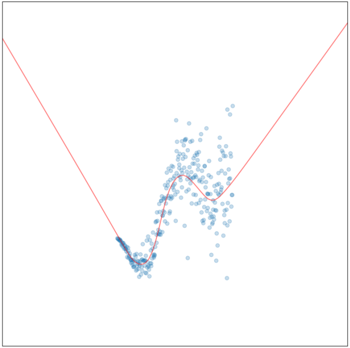
- Uncertainty estimates
- More robustness for little data
- Comprehensive mathematical framework

# Motivation

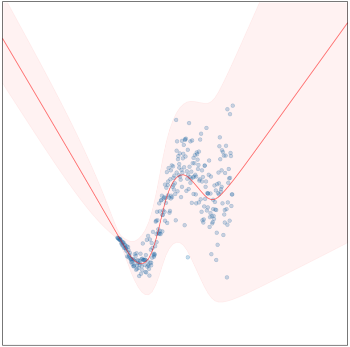


Kinda ok ...

# Motivation

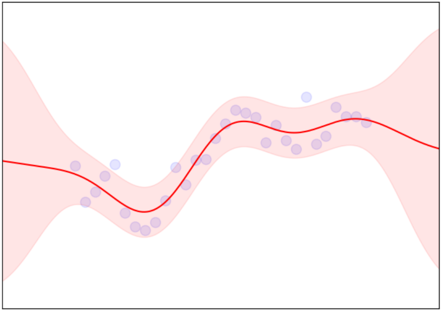


Kinda ok ...



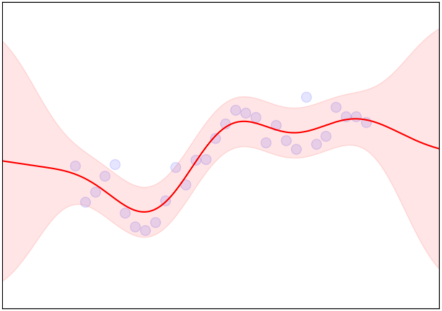
... more interesting

# Gaussian Process

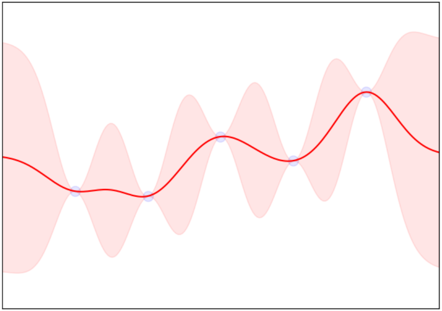


Some data ...

# Gaussian Process

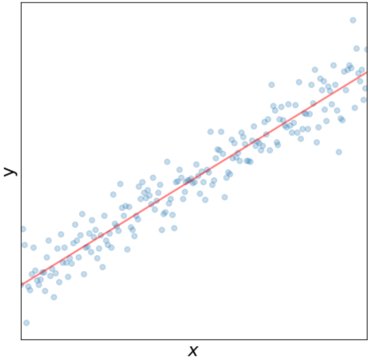


Some data ...



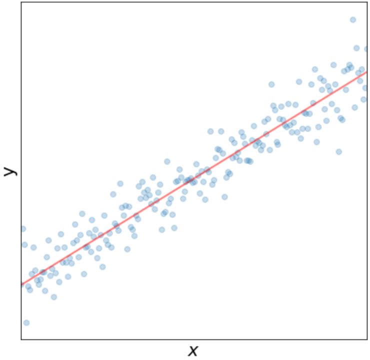
... and even less data

## Linear Regression



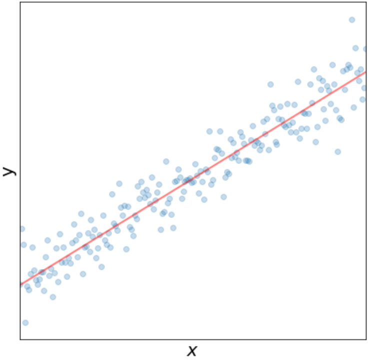

$$y = w * x + b$$

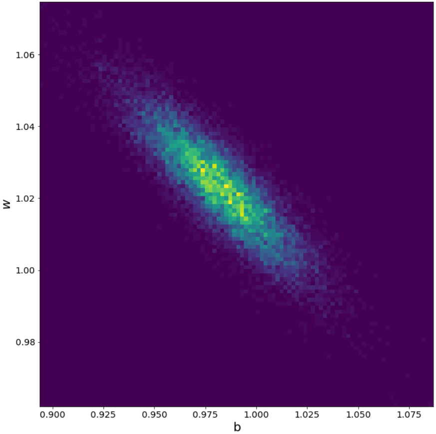
# Bayesian Linear Regression




$$y = W * x + B$$

# Bayesian Linear Regression

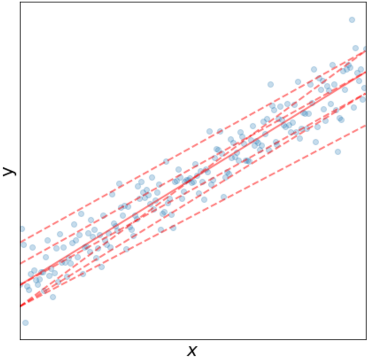


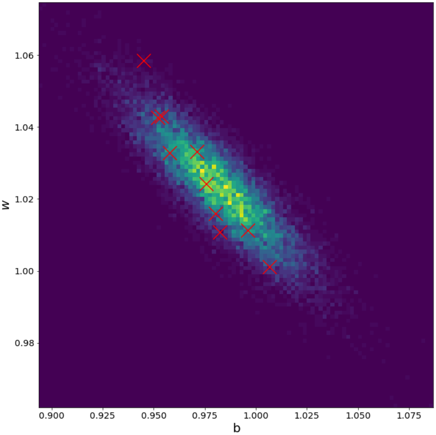




$$y = W * x + B$$

$$p(w, b|\mathcal{D})$$

# Bayesian Linear Regression




$$y = W * x + B$$

$$p(w, b|\mathcal{D})$$

# Bayes' Theorem

- Two random variables $A$ and $B$ with different realizations

# Bayes' Theorem

- Two random variables $A$ and $B$ with different realizations
- Joint probability $p(A, B)$ encodes probability of two specific realizations happening together

$$p(A, B) = p(B, A)$$

# Bayes' Theorem

- Two random variables $A$ and $B$ with different realizations
- Joint probability $p(A, B)$ encodes probability of two specific realizations happening together

$$p(A, B) = p(B, A)$$

- Joint probability contains no information on sequential order

$$p(A|B)p(B) = p(B|A)p(A)$$

# Bayes' Theorem

- Two random variables $A$ and $B$ with different realizations
- Joint probability $p(A, B)$ encodes probability of two specific realizations happening together

$$p(A, B) = p(B, A)$$

- Joint probability contains no information on sequential order

$$p(A|B)p(B) = p(B|A)p(A)$$

... and we arrive at Bayes' Theorem

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

# Bayes' Theorem

$$\overbrace{p(A|B)}^{\text{Posterior}} = \frac{\overbrace{p(B|A)}^{\text{Likelihood}}\,\overbrace{p(A)}^{\text{Prior}}}{\underbrace{p(B)}_{\text{Evidence}}}$$

# Bayes' Theorem

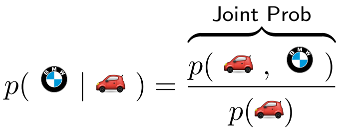
$$\overbrace{p(A|B)}^{\text{Posterior}} = \frac{\overbrace{p(B|A)}^{\text{Likelihood}}\,\overbrace{p(A)}^{\text{Prior}}}{\underbrace{p(B)}_{\text{Evidence}}}$$

An example ...

# Bayes' Theorem

$$\overbrace{p(A|B)}^{\text{Posterior}} = \frac{\overbrace{p(B|A)}^{\text{Likelihood}}\,\overbrace{p(A)}^{\text{Prior}}}{\underbrace{p(B)}_{\text{Evidence}}}$$

An example ...



# Bayesian Machine Learning

- Relationship between model parameter $\theta$ and data $\mathcal{D}$ of interest

$$\overbrace{p(\theta|\mathcal{D})}^{\text{Posterior}} = \frac{\overbrace{p(\mathcal{D}|\theta)}^{\text{Likelihood}}\ \overbrace{p(\theta)}^{\text{Prior}}}{\underbrace{p(\mathcal{D})}_{\text{Evidence}}}$$

# Bayesian Machine Learning

- Relationship between model parameter $\theta$ and data $\mathcal{D}$ of interest

$$\overbrace{p(\theta|\mathcal{D})}^{\text{Posterior}} = \frac{\overbrace{p(\mathcal{D}|\theta)}^{\text{Likelihood}}\ \overbrace{p(\theta)}^{\text{Prior}}}{\underbrace{p(\mathcal{D})}_{\text{Evidence}}}$$

- Likelihood encodes structure of machine learning model

$$p(\mathcal{D}|\theta) = p(y|x, \theta)$$

# Bayesian Machine Learning

- Relationship between model parameter $\theta$ and data $\mathcal{D}$ of interest

$$\overbrace{p(\theta|\mathcal{D})}^{\text{Posterior}} = \frac{\overbrace{p(\mathcal{D}|\theta)}^{\text{Likelihood}}\ \overbrace{p(\theta)}^{\text{Prior}}}{\underbrace{p(\mathcal{D})}_{\text{Evidence}}}$$

- Likelihood encodes structure of machine learning model

$$p(\mathcal{D}|\theta) = p(y|x, \theta)$$

- Posterior used for probabilistic prediction

$$p(y^*|x^*) = \int p(y^*|x^*, \theta)p(\theta|\mathcal{D})d\theta$$

# Sampling

- Evidence $p(\mathcal{D})$ usually intractable

$$p(\theta|\mathcal{D}) = \frac{1}{p(\mathcal{D})} p(\mathcal{D}|\theta) p(\theta)$$
$$p(\mathcal{D}) = \int p(\mathcal{D}, \theta) d\theta$$

# Sampling

- Evidence $p(\mathcal{D})$ usually intractable

$$p(\theta|\mathcal{D}) = \frac{1}{p(\mathcal{D})} p(\mathcal{D}|\theta)p(\theta)$$
$$p(\mathcal{D}) = \int p(\mathcal{D}, \theta) d\theta$$

- Sampling leads to asymptotically correct posterior distribution

$$\overbrace{p(\theta|\mathcal{D})}^{\text{Posterior}} \propto \overbrace{p(\mathcal{D}|\theta)}^{\text{Likelihood}} \overbrace{p(\theta)}^{\text{Prior}}$$

# Sampling

- Evidence $p(\mathcal{D})$ usually intractable

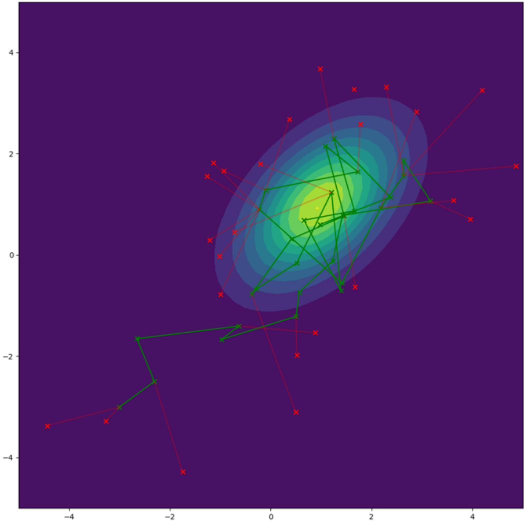
$$p(\theta|\mathcal{D}) = \frac{1}{p(\mathcal{D})} p(\mathcal{D}|\theta)p(\theta)$$
$$p(\mathcal{D}) = \int p(\mathcal{D}, \theta) d\theta$$

- Sampling leads to asymptotically correct posterior distribution

$$\overbrace{p(\theta|\mathcal{D})}^{\text{Posterior}} \propto \overbrace{p(\mathcal{D}|\theta)}^{\text{Likelihood}} \overbrace{p(\theta)}^{\text{Prior}}$$

- How to sample from these possibly extremely complex distributions?
- Naive sampling is feasible but not very smart

# Markov Chain Monte Carlo



# Markov Chain Monte Carlo

- Manhattan Project physicists come to the rescue

# Markov Chain Monte Carlo

- Manhattan Project physicists come to the rescue
- Metroplis-Hastings Algorithm to create sampling chain

# Markov Chain Monte Carlo

- Manhattan Project physicists come to the rescue

- Metroplis-Hastings Algorithm to create sampling chain
  1. Sample proposal ("Monte Carlo") around current state ("Markov")

# Markov Chain Monte Carlo

- Manhattan Project physicists come to the rescue

- Metroplis-Hastings Algorithm to create sampling chain
  1. Sample proposal ("Monte Carlo") around current state ("Markov")
  2. Accept proposal with higher value with proportional probability

# Markov Chain Monte Carlo

- Manhattan Project physicists come to the rescue

- Metroplis-Hastings Algorithm to create sampling chain
  1. Sample proposal ("Monte Carlo") around current state ("Markov")
  2. Accept proposal with higher value with proportional probability
  3. Move to proposal if accepted ("Chain")

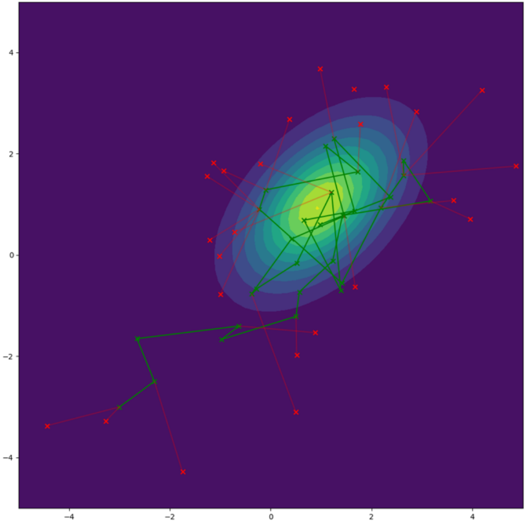
# Markov Chain Monte Carlo

- Manhattan Project physicists come to the rescue

- Metroplis-Hastings Algorithm to create sampling chain
  1. Sample proposal ("Monte Carlo") around current state ("Markov")
  2. Accept proposal with higher value with proportional probability
  3. Move to proposal if accepted ("Chain")

- Convergence to stationary distribution through detailed balance

# Metropolis-Hastings Algorithm



# Stochastic Gradient MCMC - Preliminaries

- $\log$ as monotonic function offers numerically advantages

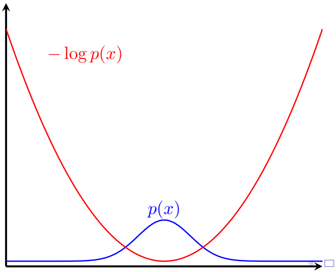
$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)\ p(\theta)$$
$$\Downarrow$$
$$\log p(\theta|\mathcal{D}) \propto \log p(\mathcal{D}|\theta) + \log p(\theta)$$

# Stochastic Gradient MCMC - Preliminaries

- $\log$ as monotonic function offers numerically advantages

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)\ p(\theta)$$
$$\Downarrow$$
$$\log p(\theta|\mathcal{D}) \propto \log p(\mathcal{D}|\theta) + \log p(\theta)$$

- Persistent gradient with $-\log p(x)$

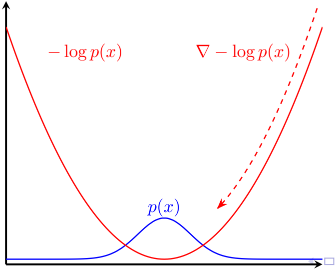


# Stochastic Gradient MCMC - Preliminaries

- $\log$ as monotonic function offers numerically advantages

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)\ p(\theta)$$
$$\Downarrow$$
$$\log p(\theta|\mathcal{D}) \propto \log p(\mathcal{D}|\theta) + \log p(\theta)$$

- Persistent gradient with $-\log p(x)$



# Stochastic Gradient MCMC

- Exploit knowledge of probabilistic loss surface during sampling

# Stochastic Gradient MCMC

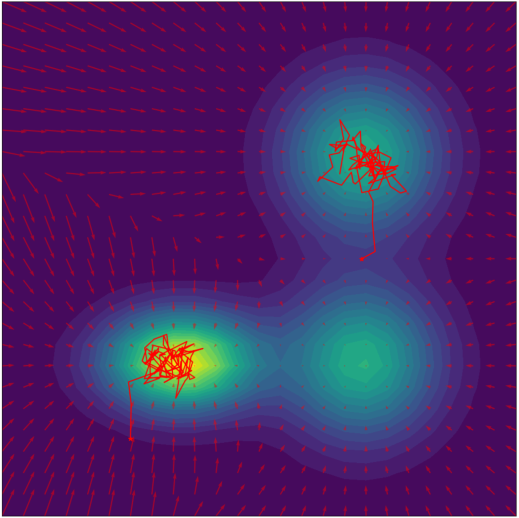
- Exploit knowledge of probabilistic loss surface during sampling

$$\Delta\theta_t = \frac{\epsilon_t}{2}\Big(\underbrace{\nabla_\theta \log p(\mathcal{D}|\theta_t) + \nabla_\theta \log p(\theta_t)}_{\text{drift}}\Big) + \underbrace{\mathcal{N}(0, \epsilon_t)}_{\text{diffusion}}$$

- Connection to Stochastic Differential Equation (SDE)

$$dX_t = \underbrace{\mu(X_t, t)}_{\text{drift}}\, dt + \underbrace{\sigma(X_t, t)}_{\text{diffusion}}\, dB_t$$

# Stochastic Gradient MCMC



# Hamiltonian Monte Carlo

# Hamiltonian Monte Carlo

- Simulates physically correct particle trajectory on probability surface

# Hamiltonian Monte Carlo

- Simulates physically correct particle trajectory on probability surface
- Constant energy $\mathcal{H}(\theta, \dot{\theta})$ shifted between potential and kinetic energy

# Hamiltonian Monte Carlo

- Simulates physically correct particle trajectory on probability surface
- Constant energy $\mathcal{H}(\theta, \dot{\theta})$ shifted between potential and kinetic energy
- Sampling of kinetic energy for each trajectory

...

# Hamiltonian Monte Carlo

- Simulates physically correct particle trajectory on probability surface
- Constant energy $\mathcal{H}(\theta, \dot{\theta})$ shifted between potential and kinetic energy
- Sampling of kinetic energy for each trajectory

...

Math is kinda intricate ...

# Hamiltonian Monte Carlo

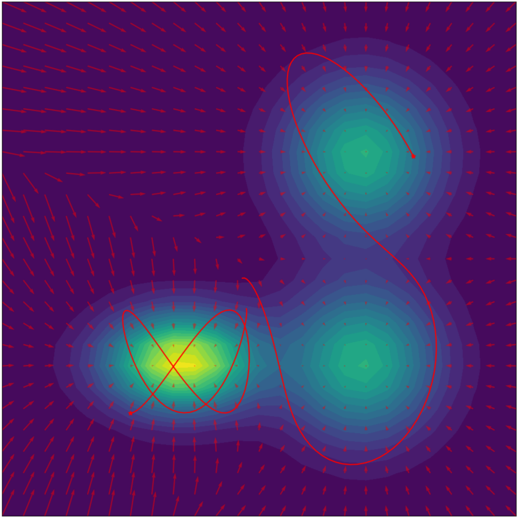
- Simulates physically correct particle trajectory on probability surface
- Constant energy $\mathcal{H}(\theta, \dot{\theta})$ shifted between potential and kinetic energy
- Sampling of kinetic energy for each trajectory

...

Math is kinda intricate ...

...so let's look at pictures

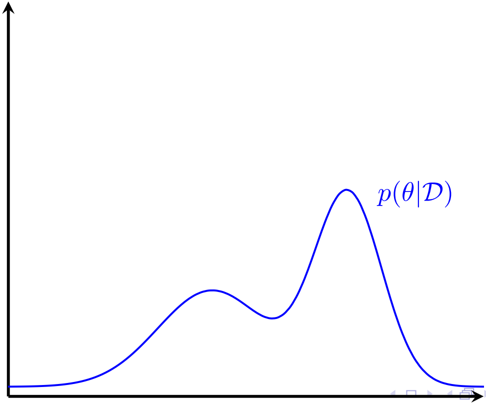
# Hamiltonian Monte Carlo



# Variational Inference

- Find some parametric approximation to true posterior

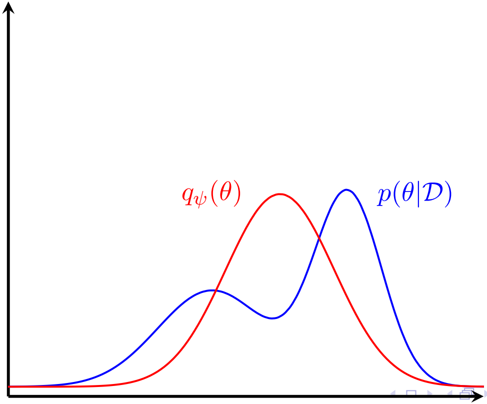
$$q_\psi(\theta) \approx p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$$



# Variational Inference

- Find some parametric approximation to true posterior

$$q_\psi(\theta) \approx p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$$



- Try to optimize parameters $\psi$ by minimize loss $\mathcal{L}$

$$\min_{\psi} \quad \mathcal{L}(q_{\psi}(\theta), p(\theta|\mathcal{D}))$$

- Try to optimize parameters $\psi$ by minimize loss $\mathcal{L}$

$$\min_{\psi} \quad \mathcal{L}(q_{\psi}(\theta), p(\theta|\mathcal{D}))$$

- Information-theoretic measure of probability distributions

$$\begin{aligned} &\min_{\psi} \quad \mathbb{KL}\left[q_{\psi}(\theta)||p(\theta|\mathcal{D})\right] \\ =&\min_{\psi} \quad \mathbb{E}_{q_{\psi}(\theta)}\left[\log \frac{q_{\psi}(\theta)}{p(\theta|\mathcal{D})}\right] \end{aligned}$$

# Variational Inference

- Optimize via Kullback-Leibler divergence

$$\min_{\psi} \quad \mathbb{KL}\left[q_{\psi}(\theta)||p(\theta|\mathcal{D})\right]$$

# Variational Inference

- Optimize via Kullback-Leibler divergence

$$
\begin{aligned}
\min_{\psi} \quad & \mathbb{KL}\left[q_\psi(\theta)||p(\theta|\mathcal{D})\right] \\
& = \mathbb{KL}\left[q_\psi(\theta)||p(\mathcal{D}|\theta)p(\theta)\right] + \mathbb{E}_q\left[\log p(\mathcal{D})\right]
\end{aligned}
$$

# Variational Inference

- Optimize via Kullback-Leibler divergence

$$
\begin{aligned}
\min_{\psi} \quad & \mathbb{KL}\left[q_\psi(\theta) || p(\theta|\mathcal{D})\right] \\
&= \mathbb{KL}\left[q_\psi(\theta) || p(\mathcal{D}|\theta)p(\theta)\right] + \mathbb{E}_q\left[\log p(\mathcal{D})\right] \\
&= \mathbb{E}_q\left[\log \frac{q_\psi(\theta)}{p(\mathcal{D}|\theta)p(\theta)}\right] + \mathbb{E}_q\left[\log p(\mathcal{D})\right]
\end{aligned}
$$

# Variational Inference

- Optimize via Kullback-Leibler divergence

$$
\begin{aligned}
\min_{\psi} \quad & \mathbb{KL}\left[q_\psi(\theta)||p(\theta|\mathcal{D})\right] \\
&= \mathbb{KL}\left[q_\psi(\theta)||p(\mathcal{D}|\theta)p(\theta)\right] + \mathbb{E}_q\left[\log p(\mathcal{D})\right] \\
&= \mathbb{E}_q\left[\log \frac{q_\psi(\theta)}{p(\mathcal{D}|\theta)p(\theta)}\right] + \mathbb{E}_q\left[\log p(\mathcal{D})\right] \\
&= \mathbb{E}_q\left[\log \frac{q_\psi(\theta)}{p(\theta)}\right] - \mathbb{E}_q\left[\log p(\mathcal{D}|\theta)\right] + \mathbb{E}_q\left[\log p(\mathcal{D})\right]
\end{aligned}
$$

# Variational Inference

- Optimize via Kullback-Leibler divergence

$$
\begin{aligned}
\min_{\psi} \quad & \mathbb{KL}\left[q_\psi(\theta)||p(\theta|\mathcal{D})\right] \\
&= \mathbb{KL}\left[q_\psi(\theta)||p(\mathcal{D}|\theta)p(\theta)\right] + \mathbb{E}_q\left[\log p(\mathcal{D})\right] \\
&= \mathbb{E}_q\left[\log \frac{q_\psi(\theta)}{p(\mathcal{D}|\theta)p(\theta)}\right] + \mathbb{E}_q\left[\log p(\mathcal{D})\right] \\
&= \mathbb{E}_q\left[\log \frac{q_\psi(\theta)}{p(\theta)}\right] - \mathbb{E}_q\left[\log p(\mathcal{D}|\theta)\right] + \mathbb{E}_q\left[\log p(\mathcal{D})\right] \\
&= \underbrace{-\mathbb{E}_q\left[\log p(\mathcal{D}|\theta)\right]}_{\text{model likelihood}} + \underbrace{\mathbb{KL}[q_\psi(\theta)||p(\theta)]}_{\text{regularization}} + \underbrace{\mathbb{E}_q\left[\log p(\mathcal{D})\right]}_{\text{evidence}}
\end{aligned}
$$

# Variational Inference

- Optimize via Kullback-Leibler divergence

$$
\begin{aligned}
\min_{\psi} \quad & \mathbb{KL}\left[q_\psi(\theta) || p(\theta|\mathcal{D})\right] \\
&= \mathbb{KL}\left[q_\psi(\theta) || p(\mathcal{D}|\theta)p(\theta)\right] + \mathbb{E}_q\left[\log p(\mathcal{D})\right] \\
&= \mathbb{E}_q\left[\log \frac{q_\psi(\theta)}{p(\mathcal{D}|\theta)p(\theta)}\right] + \mathbb{E}_q\left[\log p(\mathcal{D})\right] \\
&= \mathbb{E}_q\left[\log \frac{q_\psi(\theta)}{p(\theta)}\right] - \mathbb{E}_q\left[\log p(\mathcal{D}|\theta)\right] + \mathbb{E}_q\left[\log p(\mathcal{D})\right] \\
&= \underbrace{-\mathbb{E}_q\left[\log p(\mathcal{D}|\theta)\right]}_{\text{model likelihood}} + \underbrace{\mathbb{KL}[q_\psi(\theta) || p(\theta)]}_{\text{regularization}} + \underbrace{\mathbb{E}_q\left[\log p(\mathcal{D})\right]}_{\text{evidence}} \\
\nabla_\psi \mathcal{L} = & -\nabla_\psi \mathbb{E}_q\left[\log p(\mathcal{D}|\theta)\right] + \nabla_\psi \mathbb{KL}[q_\psi(\theta) || p(\theta)]
\end{aligned}
$$

# Variational Inference

- Automatic regularization trade-off thanks to Bayesian methodology

$$\min_{\psi} \mathbb{KL}\left[q_{\psi}(\theta) || p(\theta|\mathcal{D})\right]$$

# Variational Inference

- Automatic regularization trade-off thanks to Bayesian methodology

$$
\begin{aligned}
&\min_{\psi} \mathbb{KL}\left[q_{\psi}(\theta)||p(\theta|\mathcal{D})\right] \\
&\quad \leq \; -\mathbb{E}_q\left[\log p(\mathcal{D}|\theta)\right] + \mathbb{KL}[q_{\psi}(\theta)||p(\theta)]
\end{aligned}
$$

# Variational Inference

- Automatic regularization trade-off thanks to Bayesian methodology

$$
\begin{aligned}
&\min_{\psi} \mathbb{KL}\left[q_\psi(\theta)||p(\theta|\mathcal{D})\right] \\
&\quad \leq \ -\mathbb{E}_q\left[\log p(\mathcal{D}|\theta)\right] + \mathbb{KL}[q_\psi(\theta)||p(\theta)] \\
&\quad = \ -\sum_{n=0}^{N} \mathbb{E}_q\left[\log p(y_n|x_n,\theta)\right] + \mathbb{KL}[q_\psi(\theta)||p(\theta)]
\end{aligned}
$$

# Variational Inference

- Automatic regularization trade-off thanks to Bayesian methodology

$$
\begin{aligned}
&\min_{\psi} \mathbb{KL}\left[q_\psi(\theta)||p(\theta|\mathcal{D})\right] \\
&\quad\leq -\mathbb{E}_q\left[\log p(\mathcal{D}|\theta)\right] + \mathbb{KL}[q_\psi(\theta)||p(\theta)] \\
&\quad= -\sum_{n=0}^{N} \mathbb{E}_q\left[\log p(y_n|x_n,\theta)\right] + \mathbb{KL}[q_\psi(\theta)||p(\theta)] \\
&\quad= \sum_{n=0}^{N} \left[-\mathbb{E}_q\left[\log p(y_n|x_n,\theta)\right] + \frac{1}{N}\mathbb{KL}[q_\psi(\theta)||p(\theta)]\right]
\end{aligned}
$$

# Variational Inference

- Automatic regularization trade-off thanks to Bayesian methodology

$$
\begin{aligned}
\min_{\psi} \mathbb{KL}\left[q_\psi(\theta) || p(\theta|\mathcal{D})\right] \\
&\leq -\mathbb{E}_q\left[\log p(\mathcal{D}|\theta)\right] + \mathbb{KL}[q_\psi(\theta)||p(\theta)] \\
&= -\sum_{n=0}^{N} \mathbb{E}_q\left[\log p(y_n|x_n,\theta)\right] + \mathbb{KL}[q_\psi(\theta)||p(\theta)] \\
&= \sum_{n=0}^{N}\left[-\mathbb{E}_q\left[\log p(y_n|x_n,\theta)\right] + \frac{1}{N}\mathbb{KL}[q_\psi(\theta)||p(\theta)]\right]
\end{aligned}
$$

- The more data ... the less regularization

# Semi-Implicit Variational Inference

- VI assumes parametric distribution e.g. Normal, Gamma, Beta ...
- Reformulate $q_\psi(\theta)$ as neural network with noise as input

# Semi-Implicit Variational Inference

- VI assumes parametric distribution e.g. Normal, Gamma, Beta ...
- Reformulate $q_\psi(\theta)$ as neural network with noise as input

$$\epsilon \sim \mathcal{N}(0,1) \longrightarrow T(\epsilon) \longrightarrow q(\psi) \longrightarrow q(\theta|\psi)$$

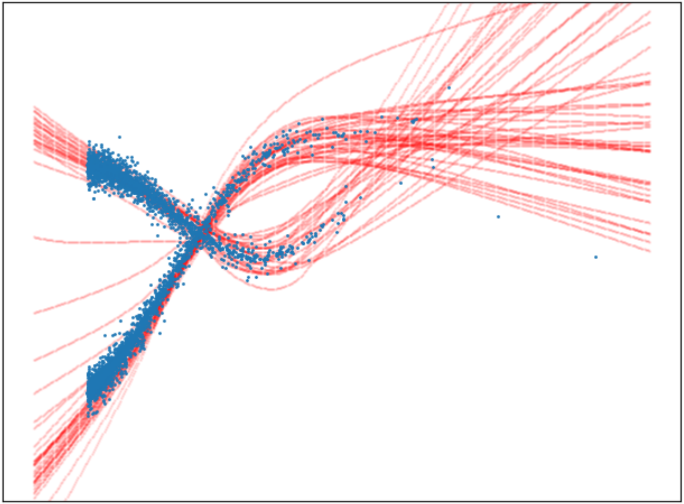
# Semi-Implicit Variational Inference

- VI assumes parametric distribution e.g. Normal, Gamma, Beta ...
- Reformulate $q_\psi(\theta)$ as neural network with noise as input

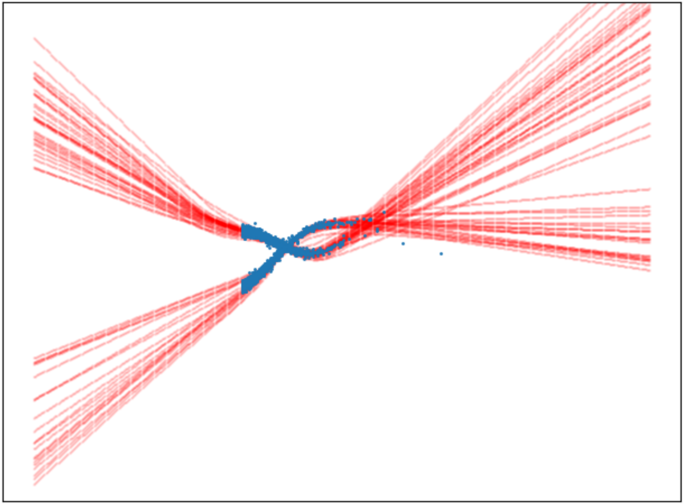
$$\epsilon \sim \mathcal{N}(0,1) \longrightarrow T(\epsilon) \longrightarrow q(\psi) \longrightarrow q(\theta|\psi)$$

- Yields flexible, implicit distribution
- Comptetitive with HMC, but as fast as VI

# Semi-Implicit Variational Inference



# Semi-Implicit Variational Inference



# Bayesian Deep Learning

- "Big Data" necessitates VI

# Bayesian Deep Learning

- "Big Data" necessitates VI

- Bayesian Compression by variance analysis
- $98\%$ compression with $1\%$ accuracy loss

# Bayesian Deep Learning

- "Big Data" necessitates VI

- Bayesian Compression by variance analysis
- $98\%$ compression with $1\%$ accuracy loss

- Sophisticated Variational Inference: KFAC, VOGN

# Bayesian Deep Learning

- "Big Data" necessitates VI

- Bayesian Compression by variance analysis
- $98\%$ compression with $1\%$ accuracy loss

- Sophisticated Variational Inference: KFAC, VOGN

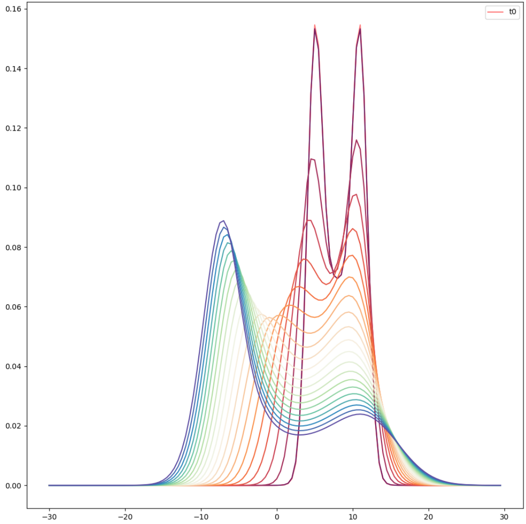
- All the functionality of deep neural networks
- All the advantages of Bayesian Machine Learning

... but BNN's are their own talk

# Sources


- Hastings : Monte Carlo Sampling Methods Using Markov Chains and Their Applications
- Welling : Bayesian Learning via Stochastic Gradient Langevin Dynamics
- Neal : MCMC Using Hamiltonian Dynamics
- Jordan & Wainright : Graphical models, exponential families, and variational inference
- Yin & Zhou: Semi-Implicit Variational Inference
- Osawa : Practical Deep Learning with Bayesian Principles
- Khan : Bayesian Inference through Weight Perturbation

# Normalizing Flows



## Change of Variable

- Instead of learning distribution, can we transform one?
- Assume some bijective transformation $y = f(x) \Leftrightarrow x = f^{-1}(y)$

$$
\begin{aligned}
p(y)dy &= p(x)dx \\
p(y) &= p(x)\left|\frac{dx}{dy}\right| \\
p(y) &= p(x)\left|\frac{df^{-1}(y)}{dy}\right|
\end{aligned}
$$

- Under bijective, continuously differentiable $f(\cdot)$

$$
p(y) = p(x)\left|\frac{df(y)}{dy}\right|^{-1}
$$

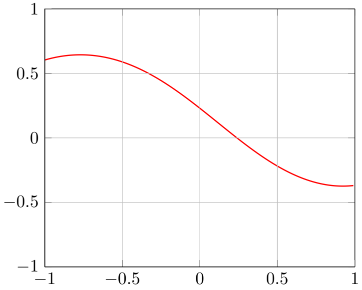
# Normalizing Flow

$$p(y) = p(x) \left| \frac{df(y)}{dy} \right|^{-1}$$

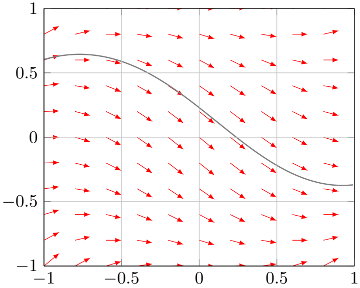
- Modern machine learning: make $f(\cdot)$ a neural network
- Transform $f : X \to Y$ with corresponding adjustement of $p(y)$
- Fokker-Planck SDE without diffusion function
- More practicable application in reversible ResNets

# Differential Equations in a Nutshell

- Relates one or more functions to their derivatives




$y = f(x)$




$\frac{dy}{dt} = f(y,t)$

- Simplest case and notation

$$dy = f(y,t)dt$$