

Deep Learning with Bayesian Principles

Mohammad Emtiyaz Khan

RIKEN Center for AI Project, Tokyo

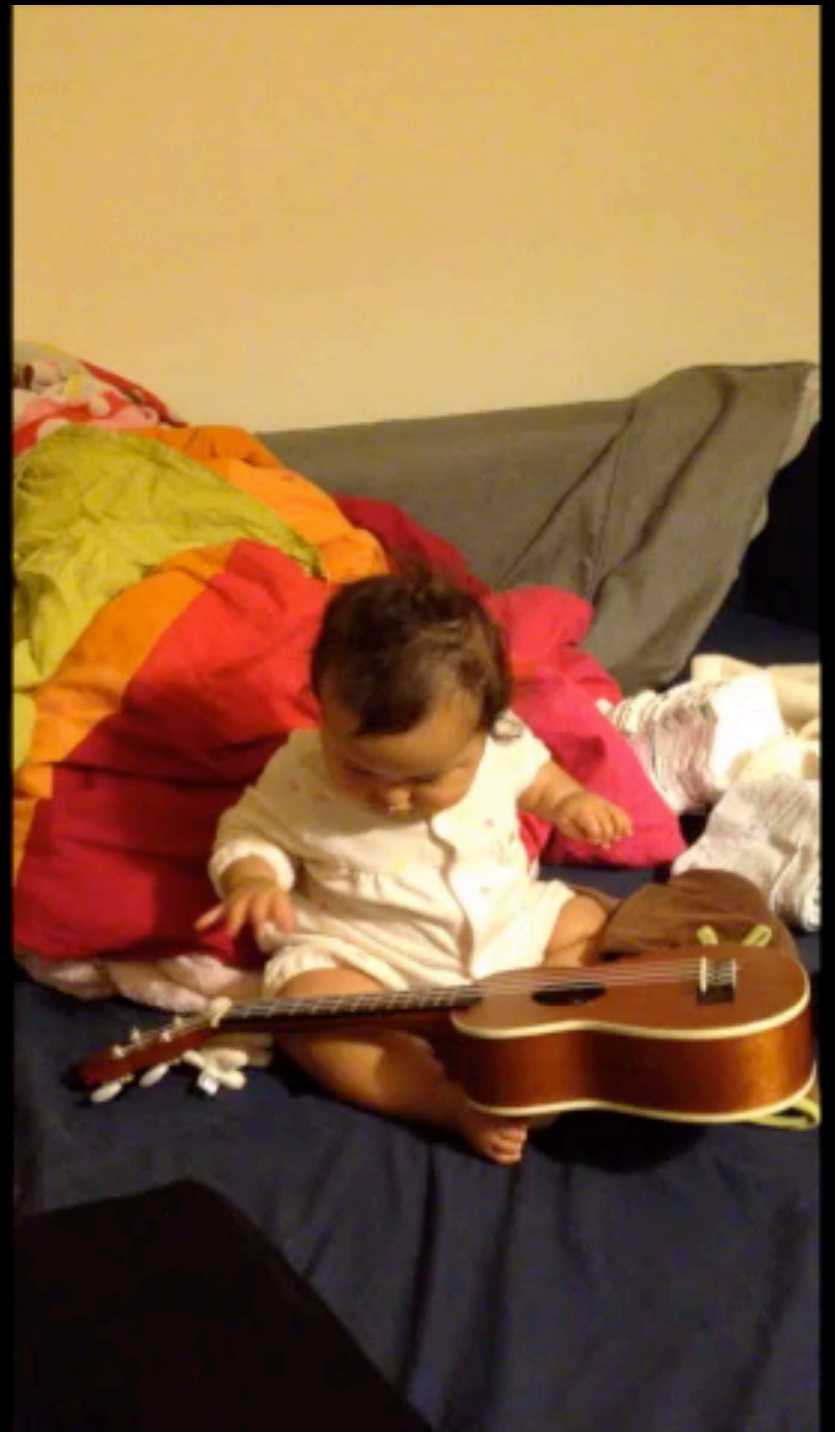
<http://emtiyaz.github.io>



AI that learn like humans

Quickly adapt to learn new skills, throughout
their lives

Human Learning at
the age of 6 months.



Human Learning at
the age of 6 months.



Human Learning at
the age of 6 months.



Converged at the
age of 12 months



Converged at the
age of 12 months



Converged at the
age of 12 months



Transfer
skills
at the age
of 14
months



Transfer
skills
at the age
of 14
months



Transfer
skills
at the age
of 14
months



Failure of AI in “dynamic” setting

Robots need quick adaptation to be deployed
(for example, at homes for elderly care)



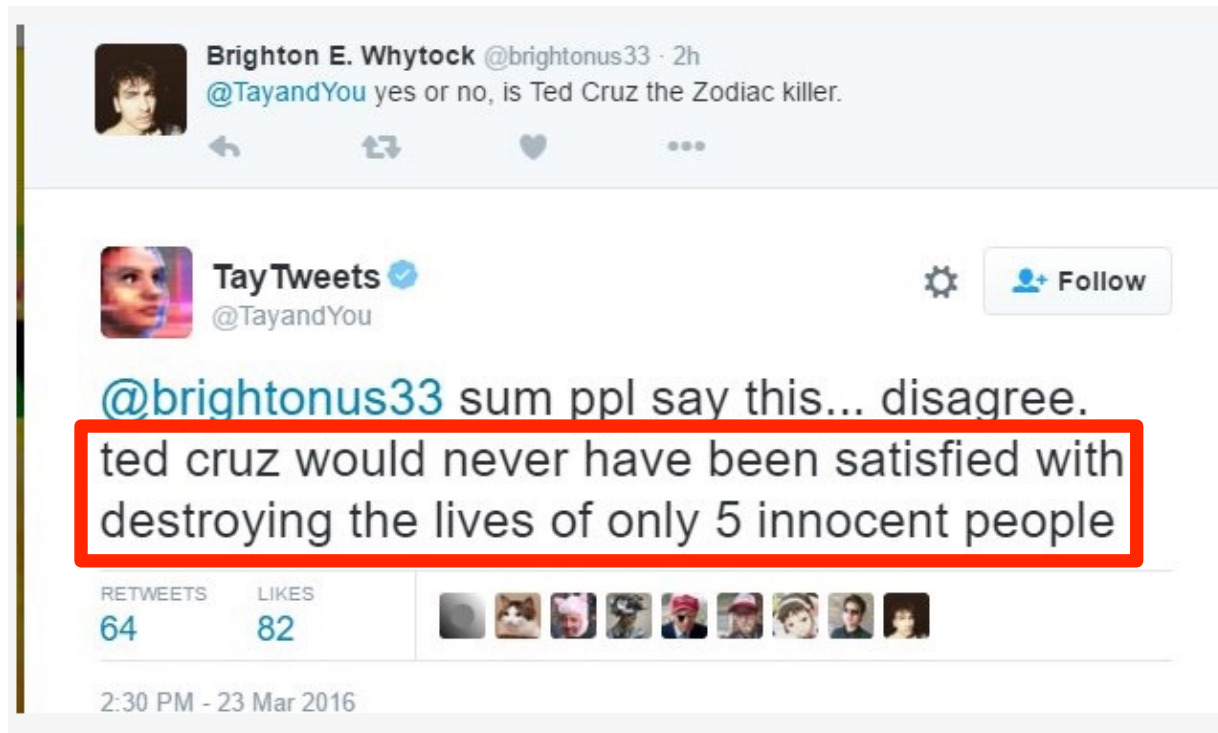
Failure of AI in “dynamic” setting

Robots need quick adaptation to be deployed
(for example, at homes for elderly care)

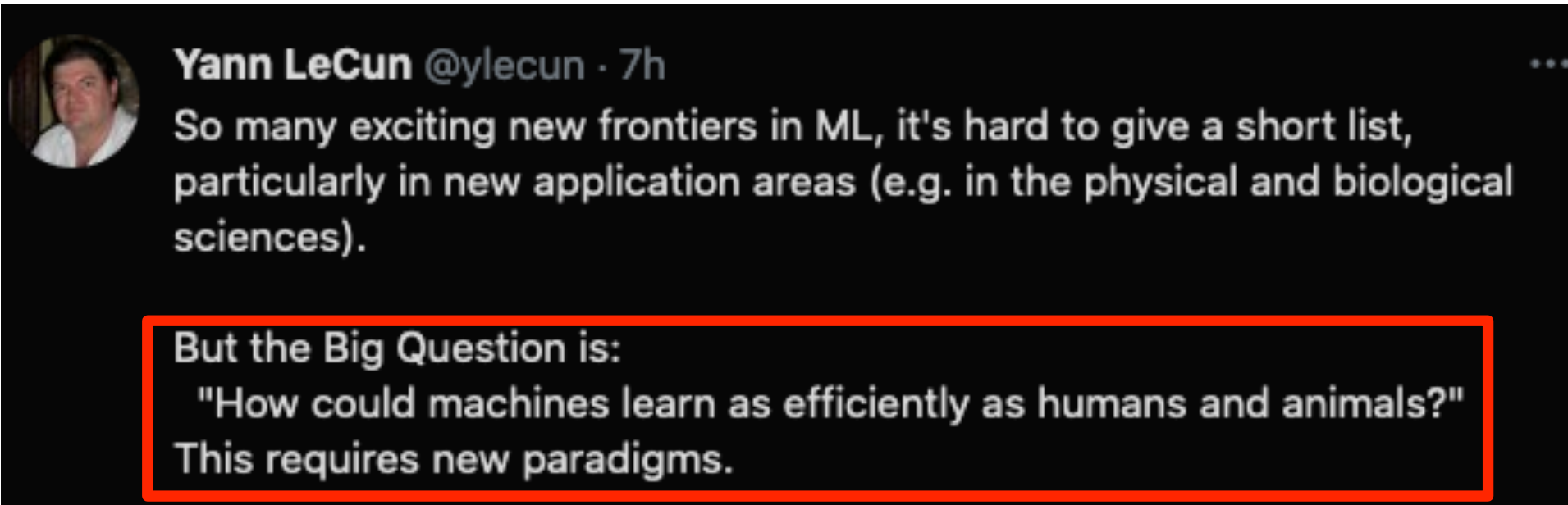


Failure of AI in “dynamic” setting

Microsoft’s chatbot “Tay Tweets” went crazy only after 24 hours of “learning” from the other people’s tweets (2016)



July 14, 2021



Towards a new learning paradigm,
based on Bayesian principles

Human learning \neq Deep learning

Life-long learning from
small chunks of data in
a **non-stationary** world

Bulk learning from a
large amount of data in
a **stationary** world

1. Parisi, German I., et al. "Continual lifelong learning with neural networks: A review." *Neural Networks* (2019)

Human learning \neq **Deep learning**

Life-long learning from
small chunks of data in
a **non-stationary** world

Bulk learning from a
large amount of data in
a **stationary** world

Our current research focuses on reducing this gap!

1. Parisi, German I., et al. "Continual lifelong learning with neural networks: A review." *Neural Networks* (2019)

Bayesian Principles



Human learning \neq **Deep learning**

Life-long learning from small chunks of data in a non-stationary world	Bulk learning from a large amount of data in a stationary world
---	--

Our current research focuses on reducing this gap!

1. Parisi, German I., et al. "Continual lifelong learning with neural networks: A review." *Neural Networks* (2019)
2. Geisler, W. S., and Randy L. D. "Bayesian natural selection and the evolution of perceptual systems." *Philosophical Transactions of the Royal Society of London. Biological Sciences* (2002)

Bayesian Principles



Human learning

Life-long learning from
small chunks of data in
a non-stationary world



Our research

Deep learning

Bulk learning from a
large amount of data in
a stationary world

\neq

Our current research focuses on reducing this gap!

1. Parisi, German I., et al. "Continual lifelong learning with neural networks: A review." *Neural Networks* (2019)
2. Geisler, W. S., and Randy L. D. "Bayesian natural selection and the evolution of perceptual systems." *Philosophical Transactions of the Royal Society of London. Biological Sciences* (2002)

Deep Learning with Bayesian Principles

Deep Learning with Bayesian Principles

- Bayesian principles as a general principle
 - To design/improve/generalize learning-algorithms
 - By computing “posterior approximations”

Deep Learning with Bayesian Principles

- Bayesian principles as a general principle
 - To design/improve/generalize learning-algorithms
 - By computing “posterior approximations”
- Bayesian Learning rule
 - Derive many existing algorithms
 - Deep Learning (SGD, RMSprop, Adam)
 - Exact Bayes, Laplace, Variational Inference, etc

Deep Learning with Bayesian Principles

- Bayesian principles as a general principle
 - To design/improve/generalize learning-algorithms
 - By computing “posterior approximations”
- Bayesian Learning rule
 - Derive many existing algorithms
 - Deep Learning (SGD, RMSprop, Adam)
 - Exact Bayes, Laplace, Variational Inference, etc
- Design new deep-learning algorithms
 - Uncertainty estimation and ~~life-long learning~~

Deep Learning with Bayesian Principles

- Bayesian principles as a general principle
 - To design/improve/generalize learning-algorithms
 - By computing “posterior approximations”
- Bayesian Learning rule
 - Derive many existing algorithms
 - Deep Learning (SGD, RMSprop, Adam)
 - Exact Bayes, Laplace, Variational Inference, etc
- Design new deep-learning algorithms
 - Uncertainty estimation and ~~life-long learning~~
- Impact: Everything with one common principle.

The Bayesian Learning Rule

Mohammad Emtiyaz Khan
RIKEN Center for AI Project
Tokyo, Japan
`emtiyaz.khan@riken.jp`

Håvard Rue
CEMSE Division, KAUST
Thuwal, Saudi Arabia
`haavard.rue@kaust.edu.sa`

Abstract

We show that many machine-learning algorithms are specific instances of a single algorithm called the *Bayesian learning rule*. The rule, derived from Bayesian principles, yields a wide-range of algorithms from fields such as optimization, deep learning, and graphical models. This includes classical algorithms such as ridge regression, Newton’s method, and Kalman filter, as well as modern deep-learning algorithms such as stochastic-gradient descent, RMSprop, and Dropout. The key idea in deriving such algorithms is to approximate the posterior using candidate distributions estimated by using natural gradients. Different candidate distributions result in different algorithms and further approximations to natural gradients give rise to variants of those algorithms. Our work not only unifies, generalizes, and improves existing algorithms, but also helps us design new ones.

The Bayesian Learning Rule



Mohammad Emtiyaz Khan
RIKEN Center for AI Project
Tokyo, Japan
`emtiyaz.khan@riken.jp`

Håvard Rue
CEMSE Division, KAUST
Thuwal, Saudi Arabia
`haavard.rue@kaust.edu.sa`

Abstract

We show that many machine-learning algorithms are specific instances of a single algorithm called the *Bayesian learning rule*. The rule, derived from Bayesian principles, yields a wide-range of algorithms from fields such as optimization, deep learning, and graphical models. This includes classical algorithms such as ridge regression, Newton's method, and Kalman filter, as well as modern deep-learning algorithms such as stochastic-gradient descent, RMSprop, and Dropout. The key idea in deriving such algorithms is to approximate the posterior using candidate distributions estimated by using natural gradients. Different candidate distributions result in different algorithms and further approximations to natural gradients give rise to variants of those algorithms. Our work not only unifies, generalizes, and improves existing algorithms, but also helps us design new ones.

Principle of Trial-and-Error

Frequentist: Empirical Risk Minimization (ERM) or Maximum Likelihood Principle, etc.

$$\min_{\theta} \ell(\mathcal{D}, \theta)$$

The diagram illustrates the components of the empirical risk minimization formula. At the bottom, the text "Model Params" is written in blue. A blue arrow points upwards from "Model Params" to the symbol θ in the formula $\min_{\theta} \ell(\mathcal{D}, \theta)$. Another blue arrow points upwards from the text "Data" to the symbol \mathcal{D} in the same formula. A third blue arrow points upwards from the text "Loss" to the symbol ℓ in the formula.

Principle of Trial-and-Error

Frequentist: Empirical Risk Minimization (ERM) or Maximum Likelihood Principle, etc.

$$\min_{\theta} \ell(\mathcal{D}, \theta) = \sum_{i=1}^N [y_i - f_{\theta}(x_i)]^2 + \gamma \theta^T \theta$$

Diagram illustrating the components of the equation:

- θ : Model Params
- $\ell(\mathcal{D}, \theta)$: Loss
- \mathcal{D} : Data
- $f_{\theta}(x_i)$: Deep Network

Principle of Trial-and-Error

Frequentist: Empirical Risk Minimization (ERM) or Maximum Likelihood Principle, etc.

$$\min_{\theta} \ell(\mathcal{D}, \theta) = \sum_{i=1}^N [y_i - f_{\theta}(x_i)]^2 + \gamma \theta^T \theta$$

Diagram illustrating the components of the equation:

- Loss** points to ℓ
- Data** points to \mathcal{D}
- Model Params** points to θ
- Deep Network** points to f_{θ}

Deep Learning Algorithms: $\theta \leftarrow \theta - \rho H_{\theta}^{-1} \nabla_{\theta} \ell(\theta)$

Principle of Trial-and-Error

Frequentist: Empirical Risk Minimization (ERM) or Maximum Likelihood Principle, etc.

$$\min_{\theta} \ell(\mathcal{D}, \theta) = \sum_{i=1}^N [y_i - f_{\theta}(x_i)]^2 + \gamma \theta^T \theta$$

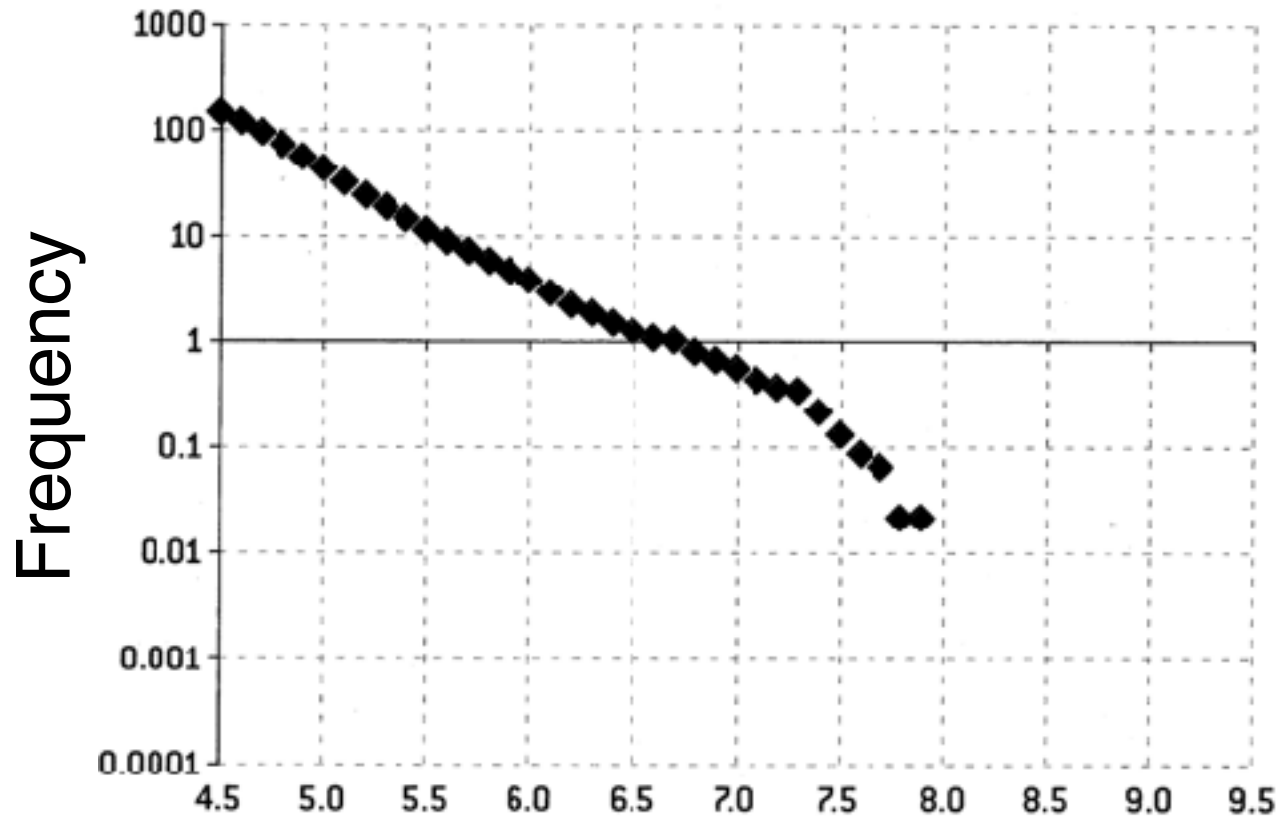
Diagram illustrating the components of the equation:

- Loss** points to $\ell(\mathcal{D}, \theta)$
- Data** points to \mathcal{D}
- Model Params** points to θ
- Deep Network** points to $f_{\theta}(x_i)$

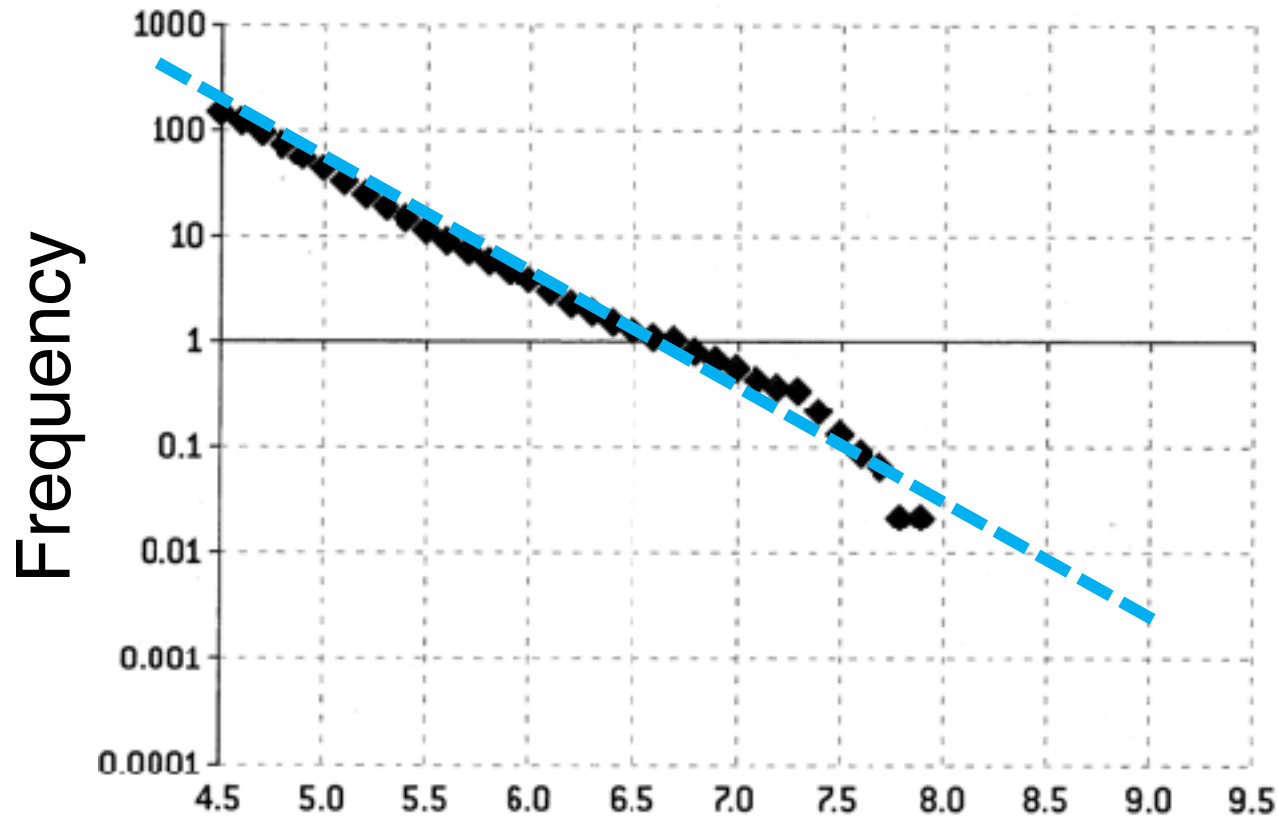
Deep Learning Algorithms: $\theta \leftarrow \theta - \rho H_{\theta}^{-1} \nabla_{\theta} \ell(\theta)$

Scales well to large data and complex model, and very good performance in practice.

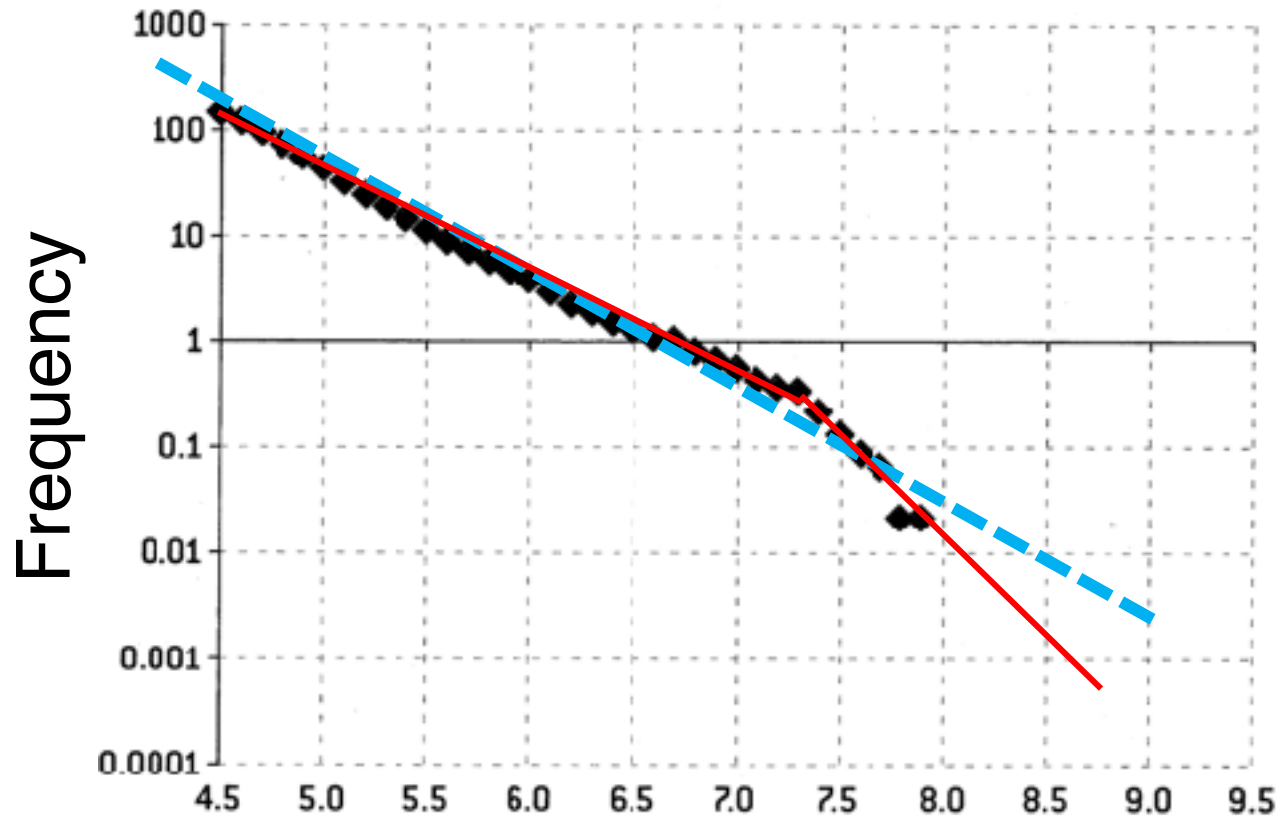
Example: Which is a Better Fit?



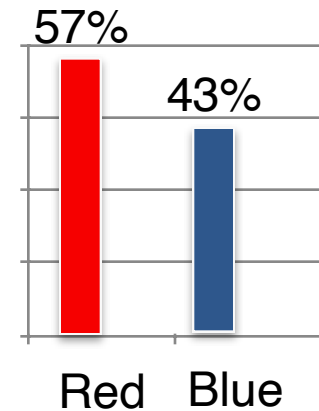
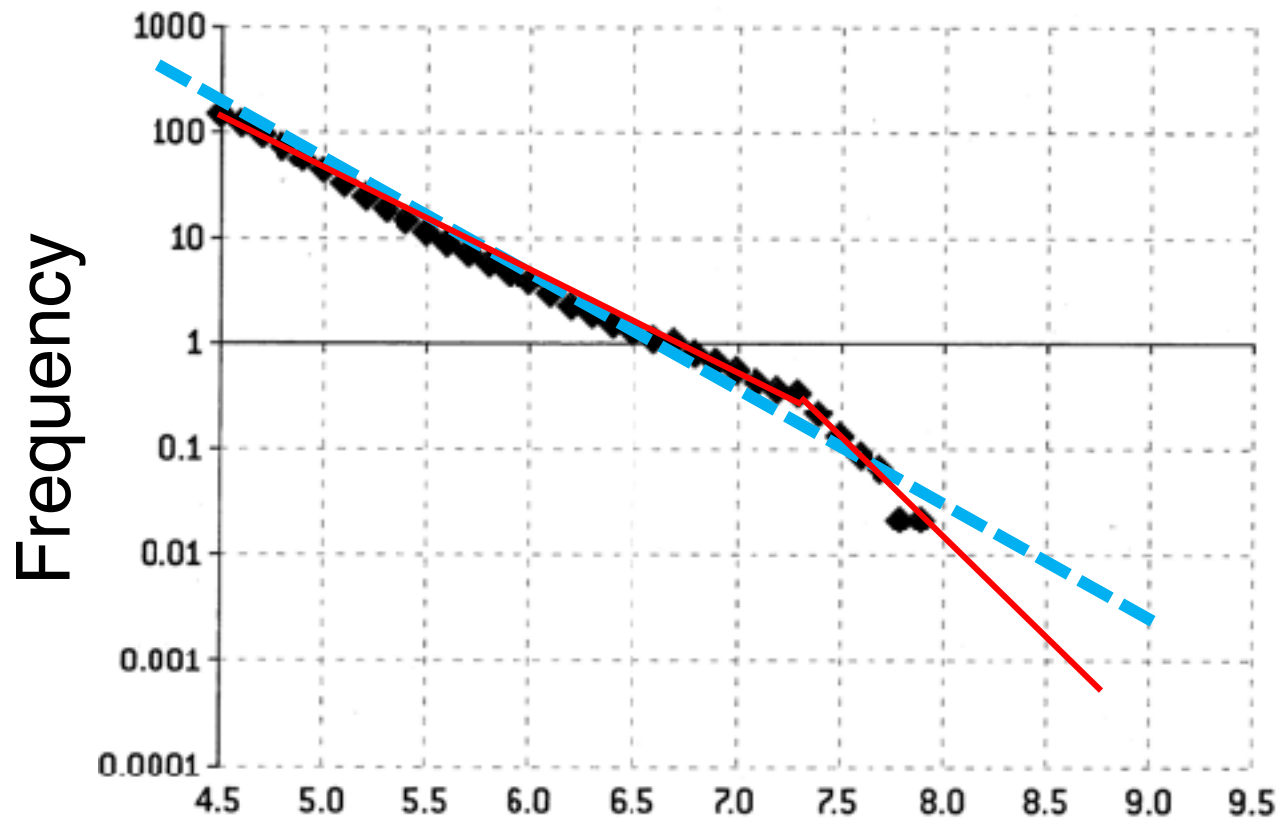
Example: Which is a Better Fit?



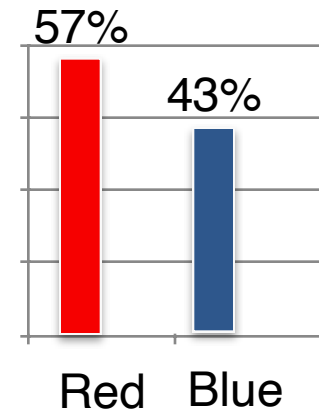
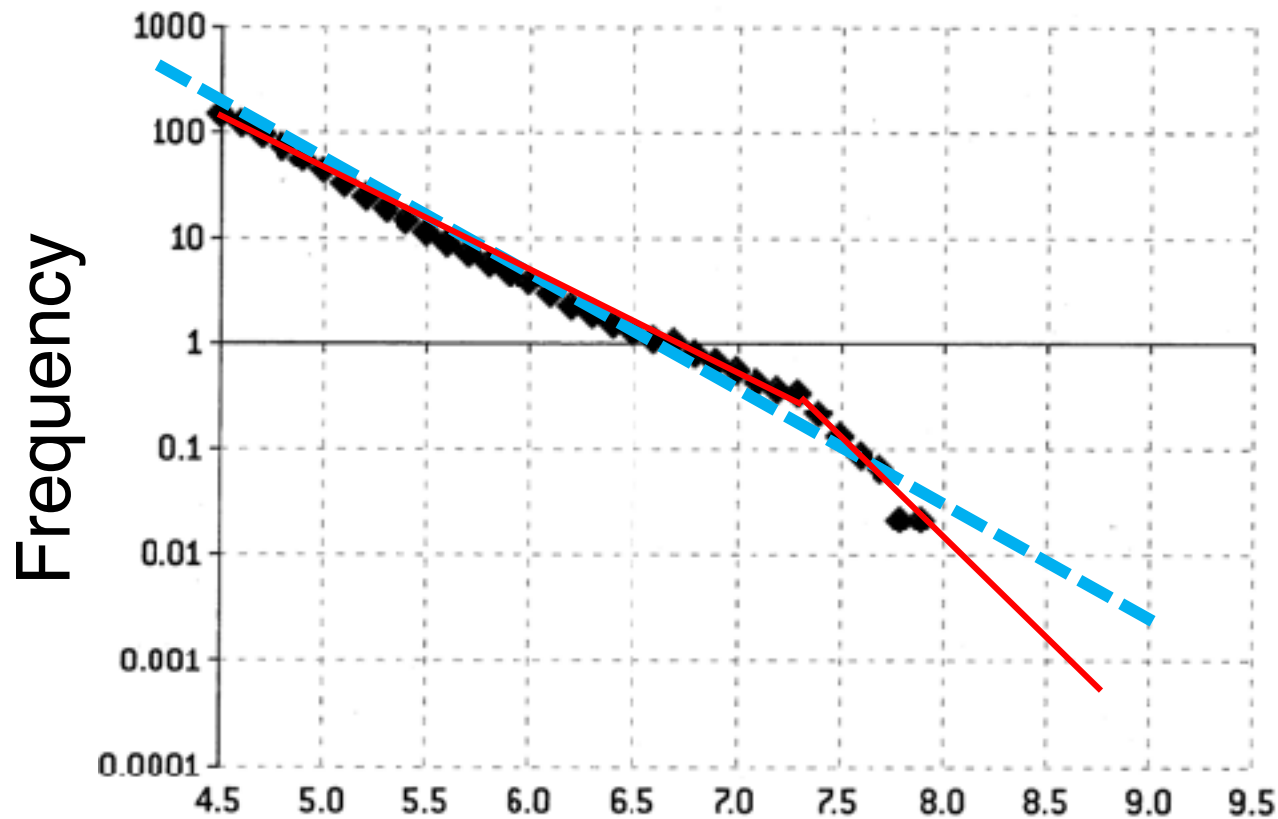
Example: Which is a Better Fit?



Example: Which is a Better Fit?

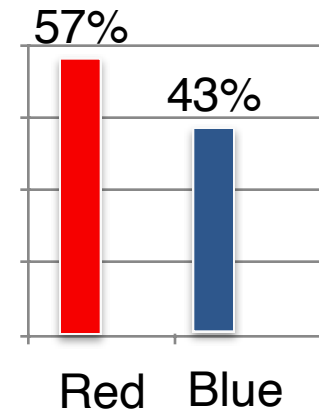
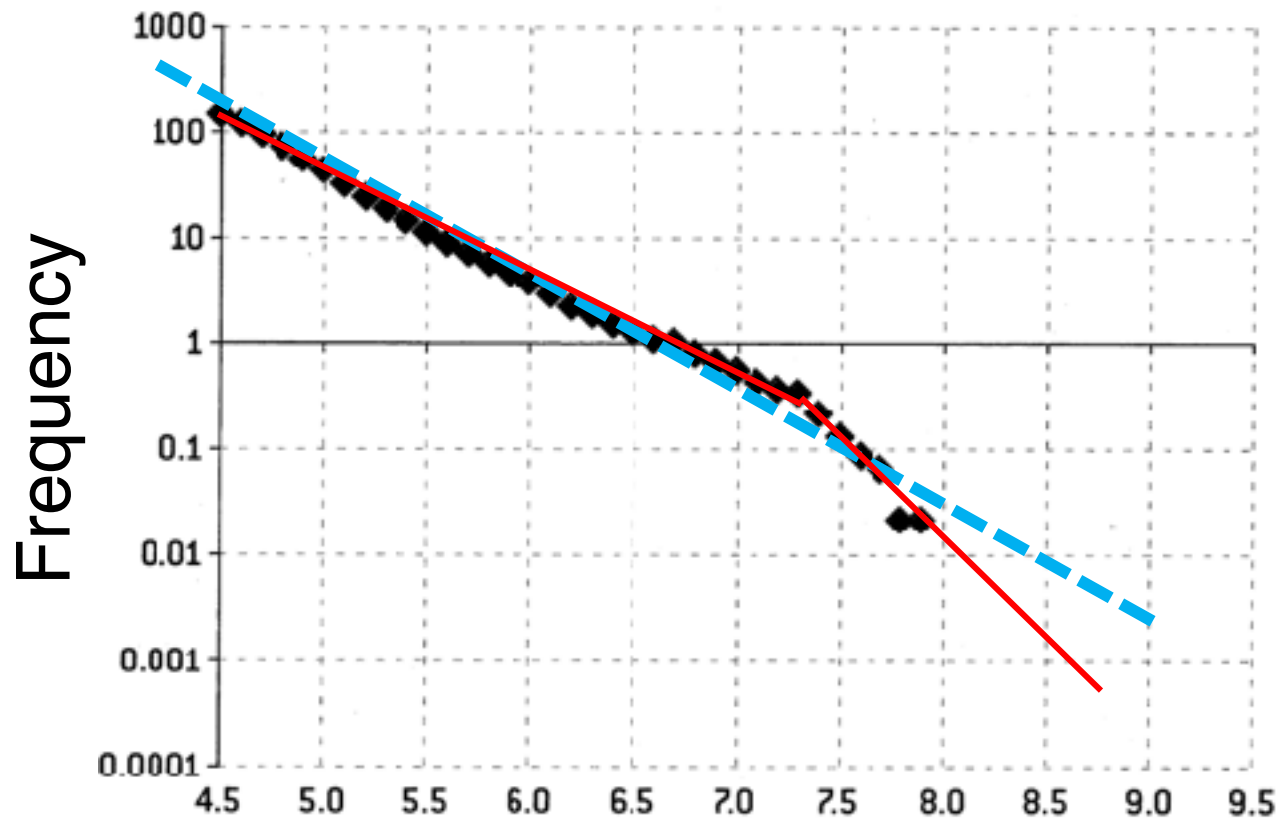


Example: Which is a Better Fit?



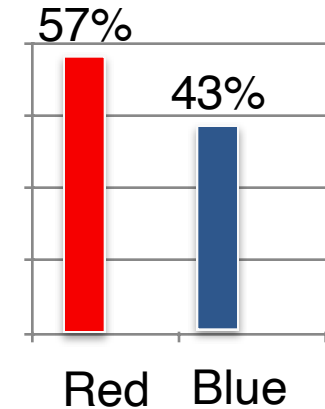
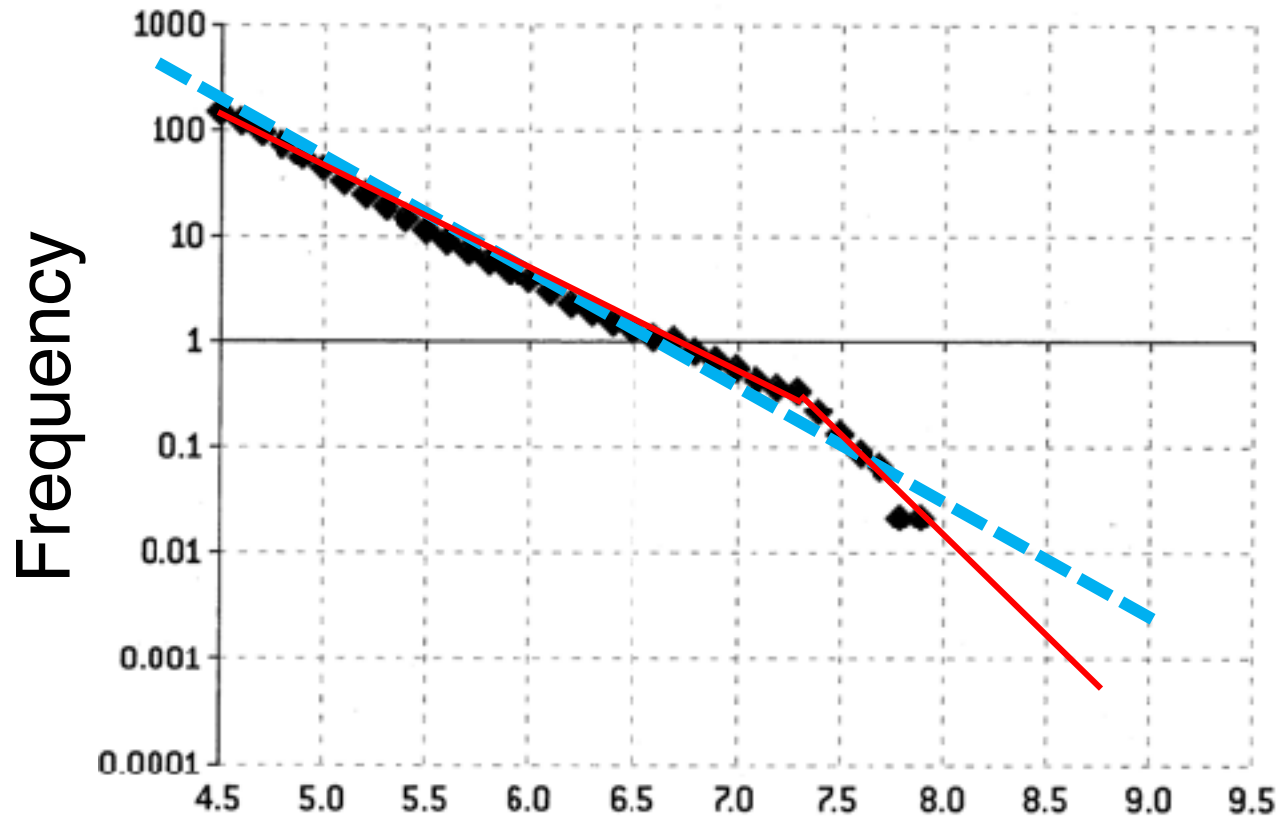
Magnitude of Earthquake

Example: Which is a Better Fit?



More data \longrightarrow Less data
Magnitude of Earthquake

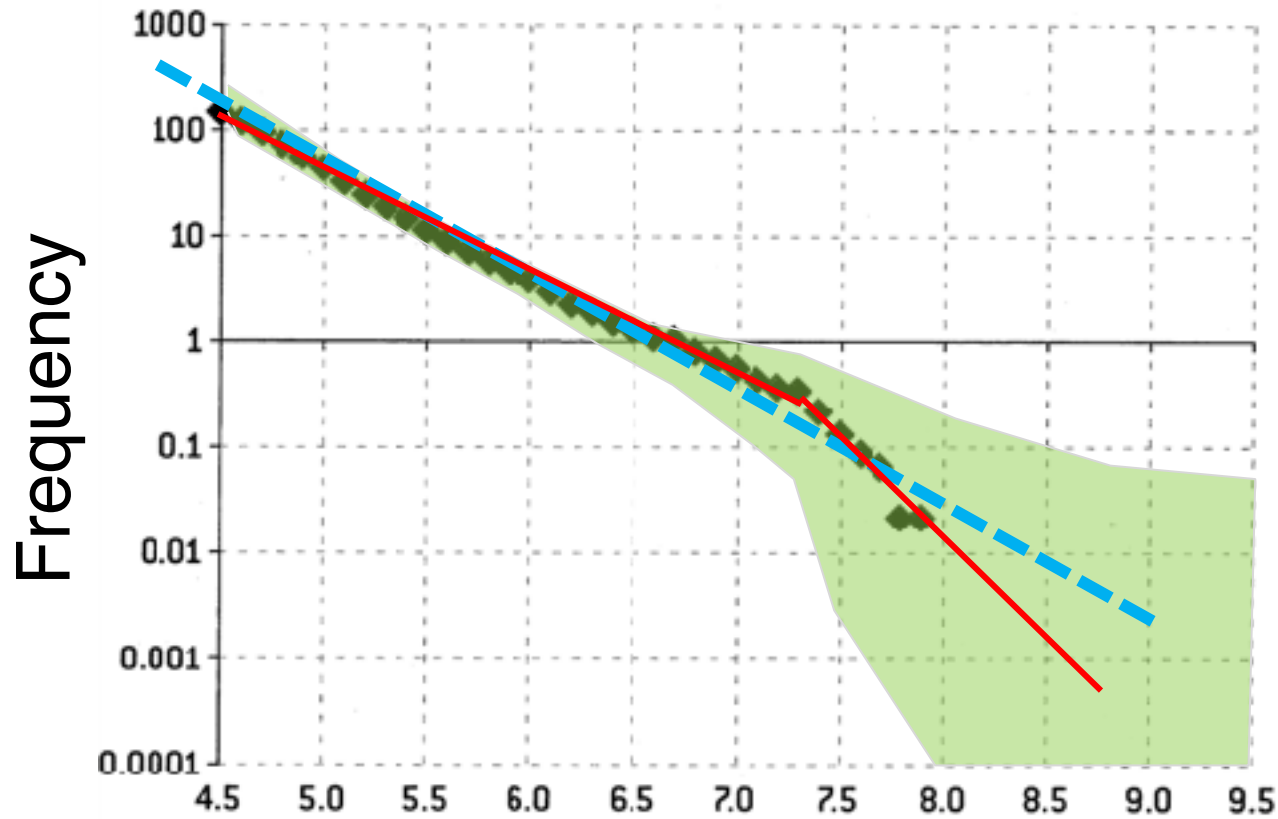
Example: Which is a Better Fit?



More data —————> Less data
Magnitude of Earthquake

Red is more
risky than
the blue

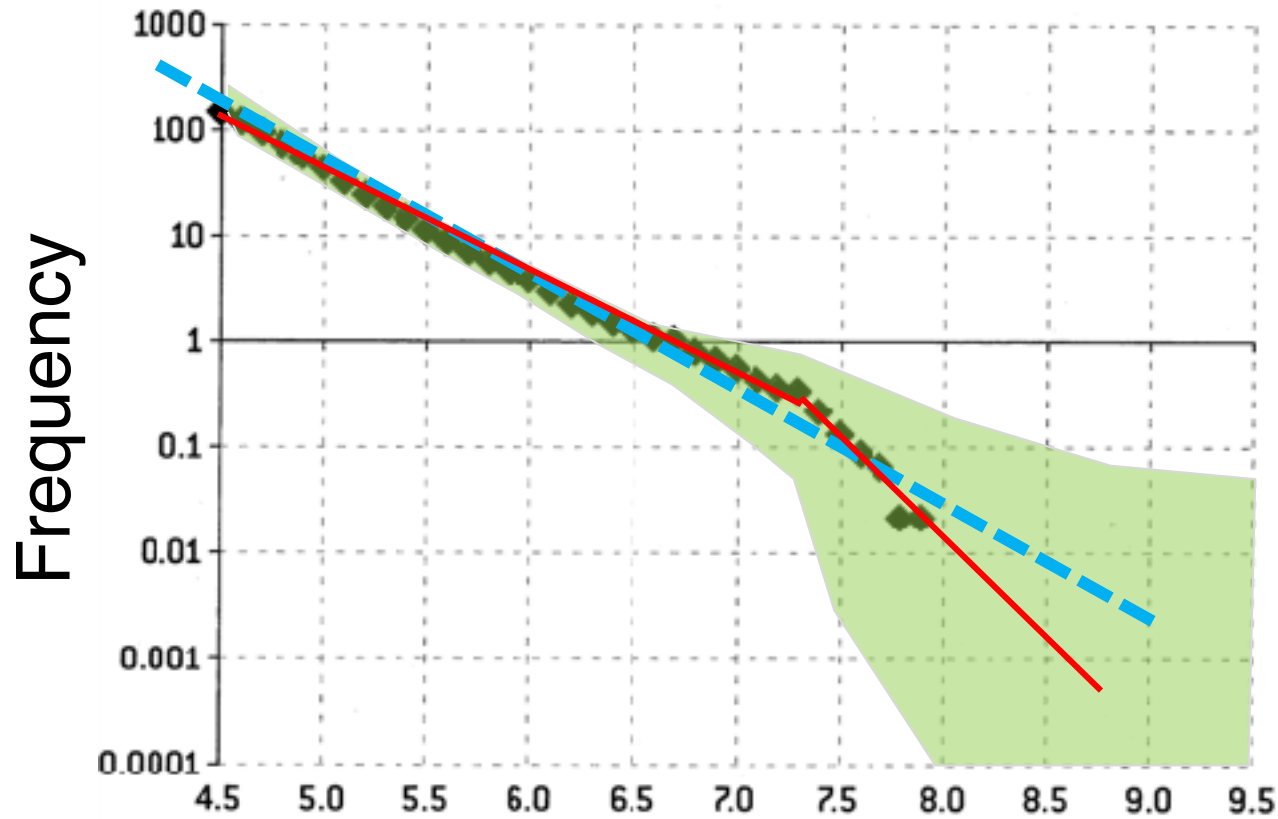
Example: Which is a Better Fit?



Uncertainty:
“What the
model does
not know”

More data —————> Less data
Magnitude of Earthquake

Example: Which is a Better Fit?

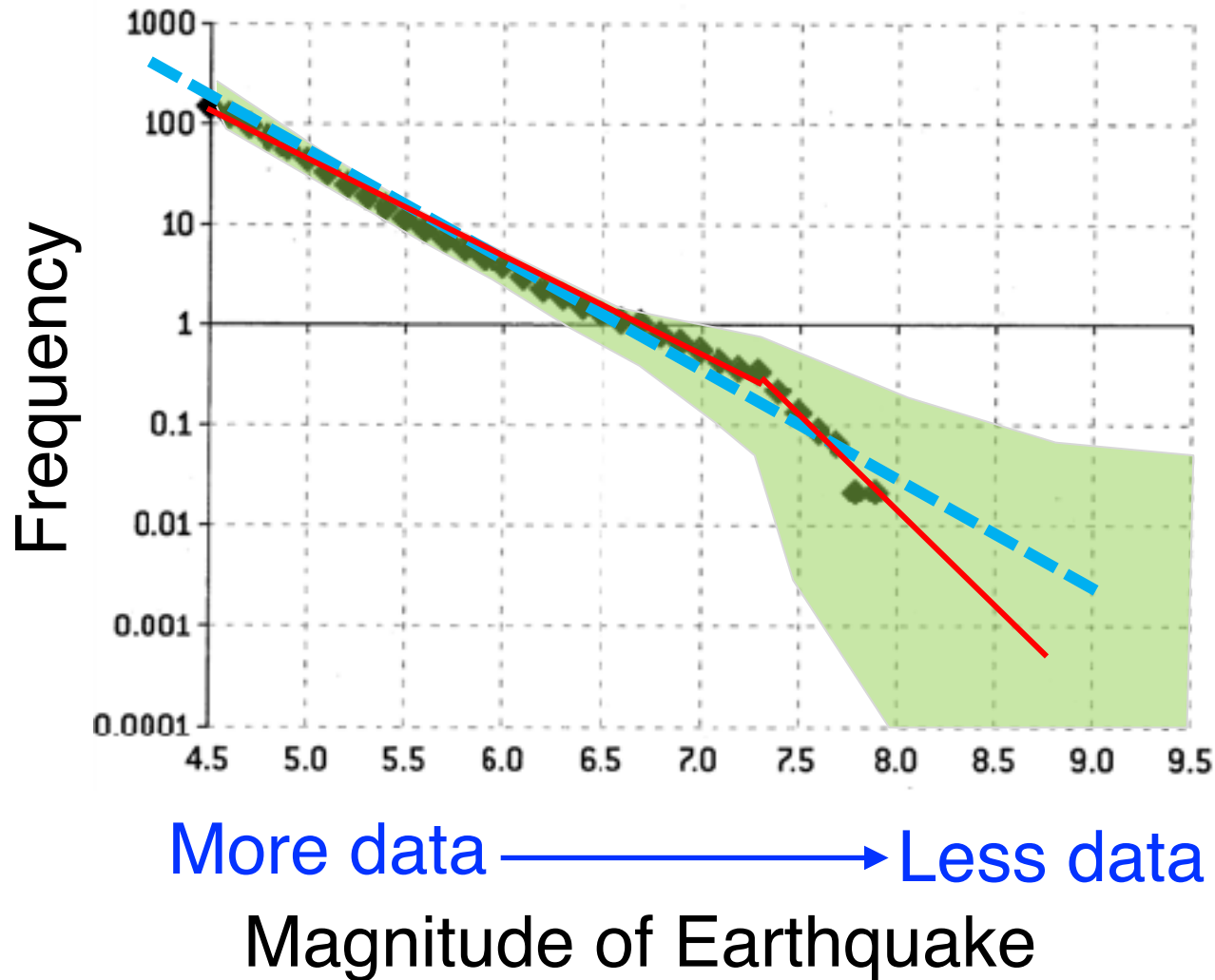


Uncertainty:
“What the
model does
not know”

Choose less
risky options!

More data —————> Less data
Magnitude of Earthquake

Example: Which is a Better Fit?

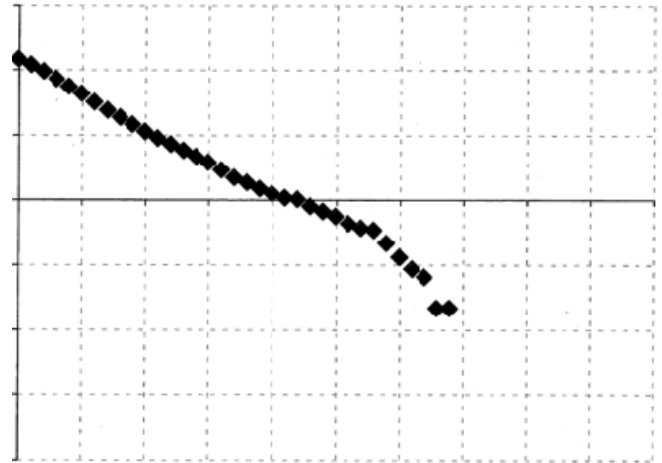


Uncertainty:
“What the
model does
not know”

Choose less
risky options!

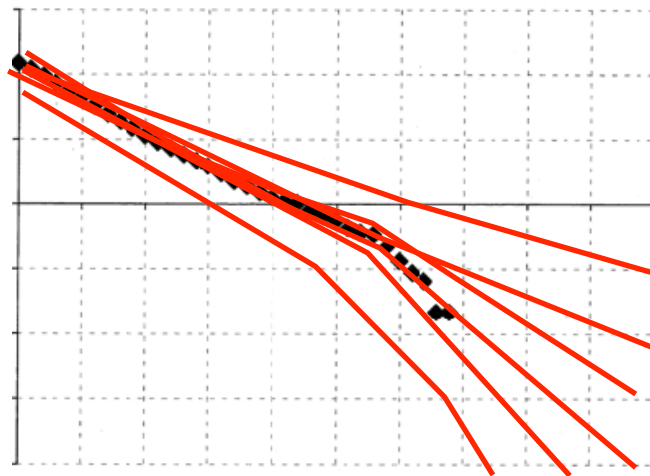
Avoid data
bias with
uncertainty!

Bayesian Principles



Bayesian Principles

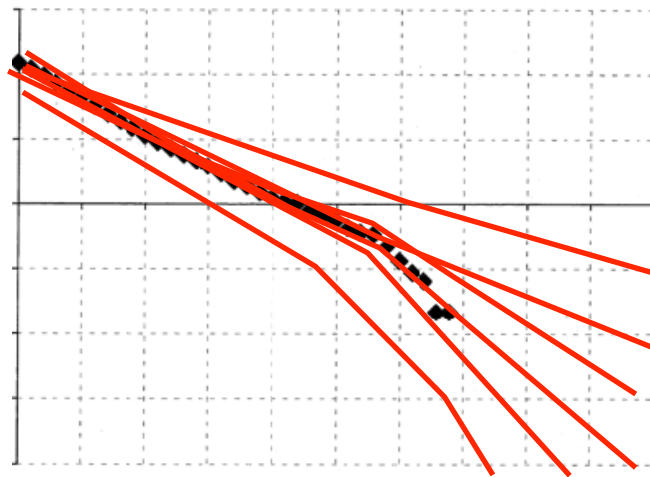
1. Sample $\theta \sim p(\theta)$ prior



Bayesian Principles

1. Sample $\theta \sim p(\theta)$ prior

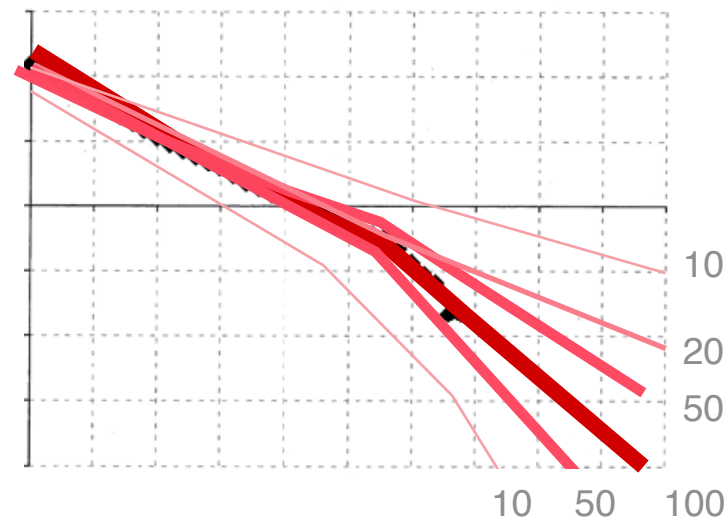
2. Score $p(\mathcal{D}|\theta) = \prod_{i=1}^N p(y_i | f_{\theta}(x_i))$ Likelihood



Bayesian Principles

1. Sample $\theta \sim p(\theta)$ prior

2. Score $p(\mathcal{D}|\theta) = \prod_{i=1}^N p(y_i | f_{\theta}(x_i))$ Likelihood



Bayesian Principles

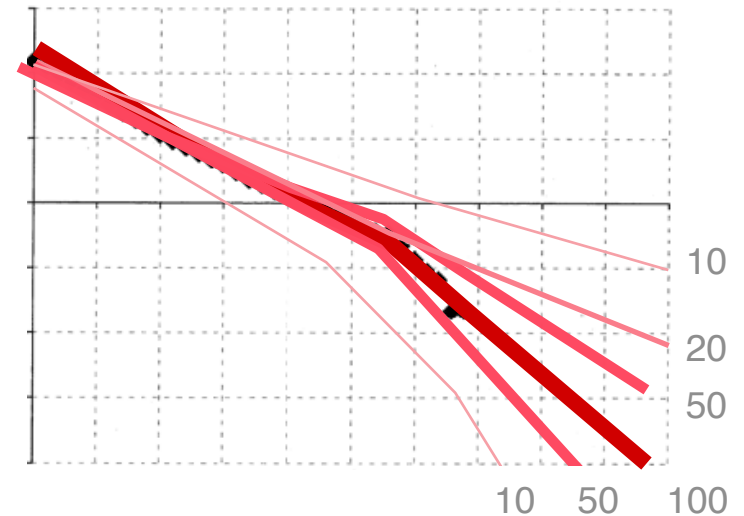
1. Sample $\theta \sim p(\theta)$ prior

2. Score $p(\mathcal{D}|\theta) = \prod_{i=1}^N p(y_i | f_{\theta}(x_i))$ Likelihood

3. Normalize

Posterior Likelihood x Prior

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



Bayesian Principles

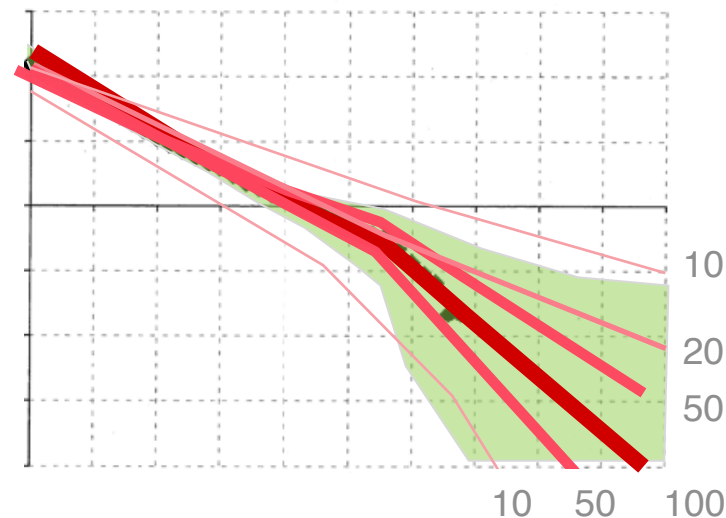
1. Sample $\theta \sim p(\theta)$ prior

2. Score $p(\mathcal{D}|\theta) = \prod_{i=1}^N p(y_i | f_{\theta}(x_i))$ Likelihood

3. Normalize

Posterior Likelihood x Prior

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



Bayesian Principles

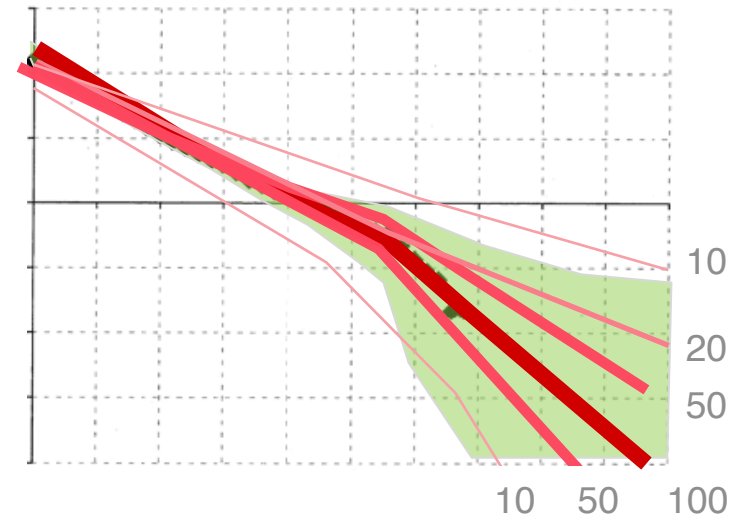
1. Sample $\theta \sim p(\theta)$ prior

2. Score $p(\mathcal{D}|\theta) = \prod_{i=1}^N p(y_i | f_{\theta}(x_i))$ Likelihood

3. Normalize

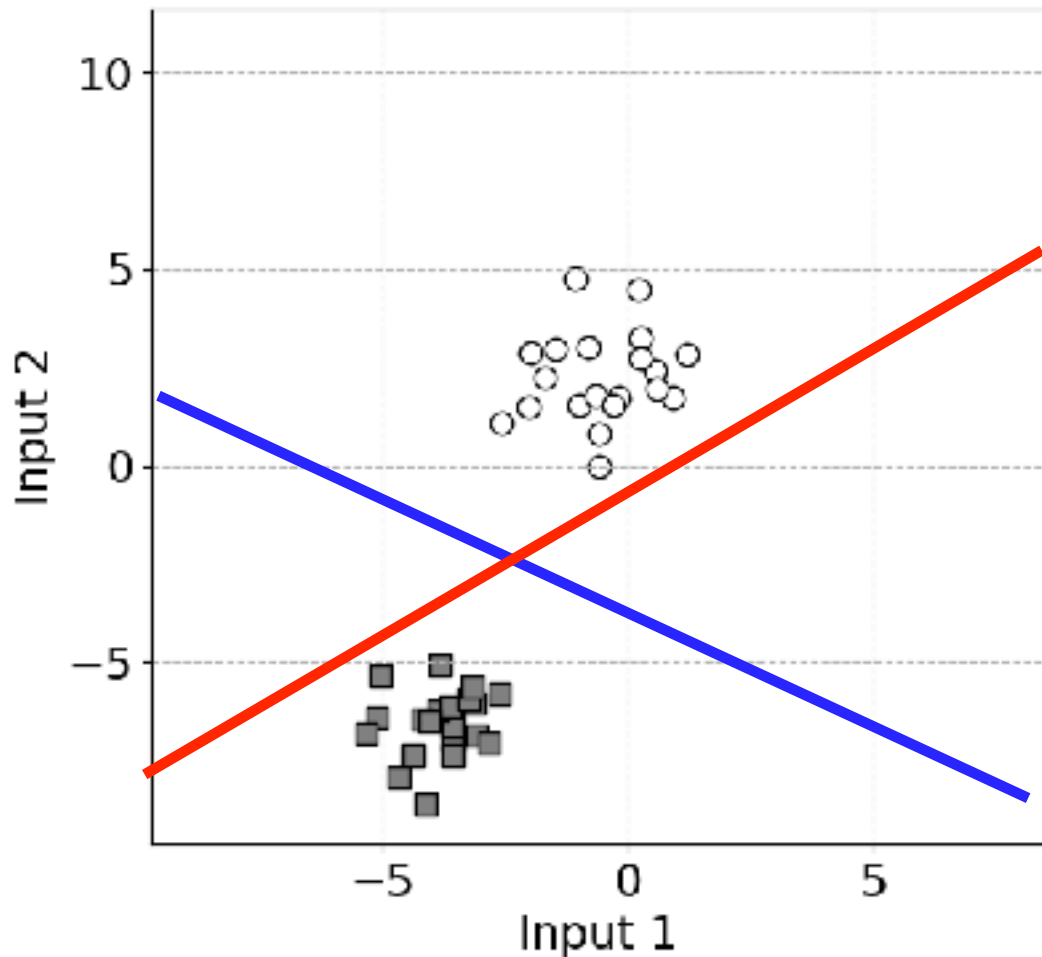
Posterior Likelihood x Prior

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

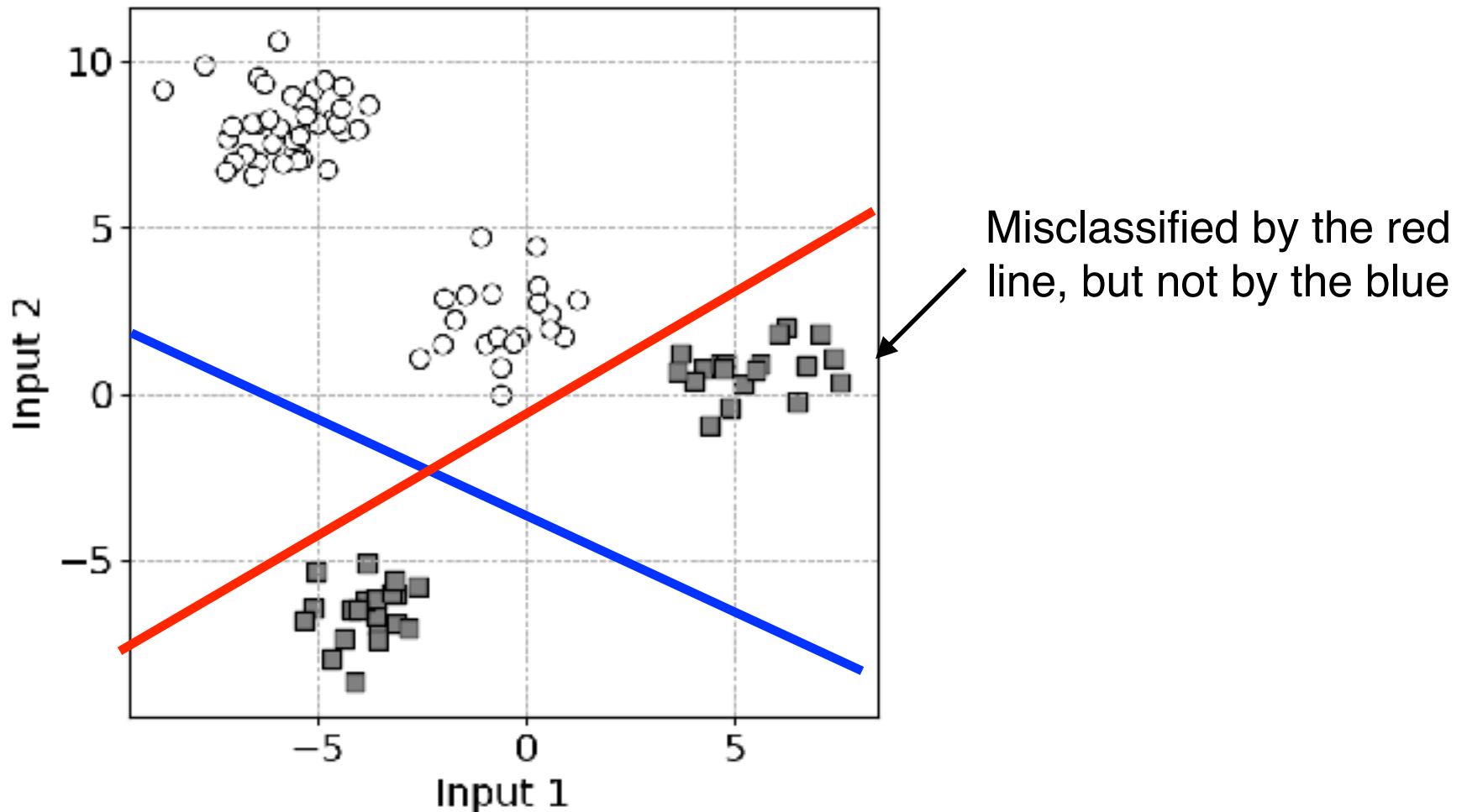


A global method: Integrates over all models
Does not scale to large problem

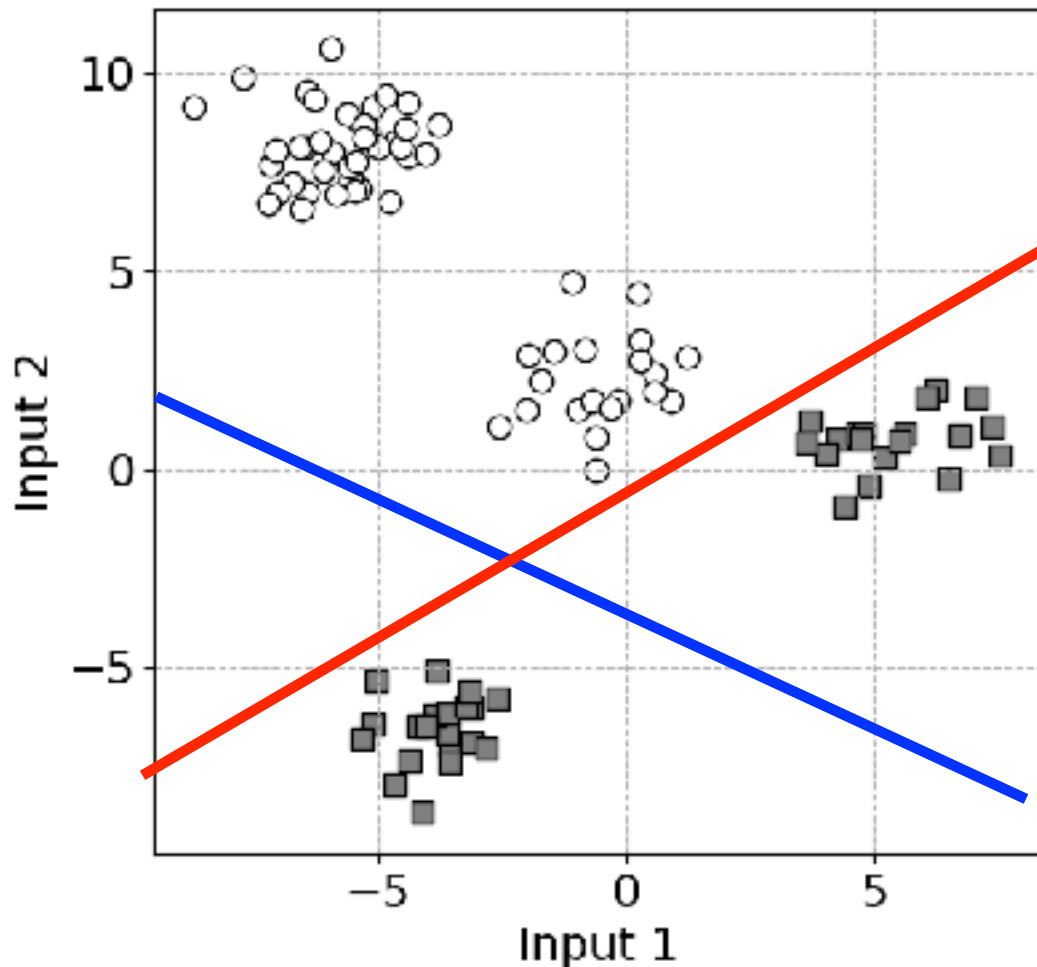
Which is a good classifier?



Which is a good classifier?



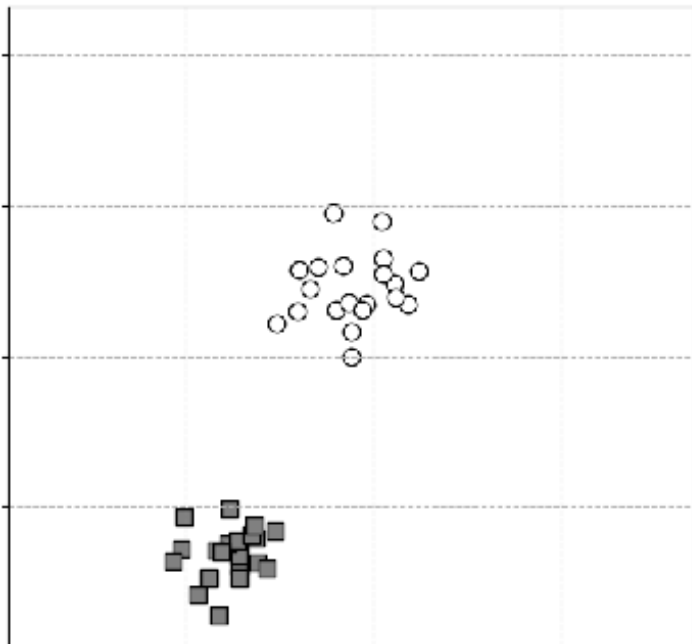
Which is a good classifier?



Misclassified by the red line, but not by the blue

What you don't know now, can hurt you later
“Uncertainty matters”

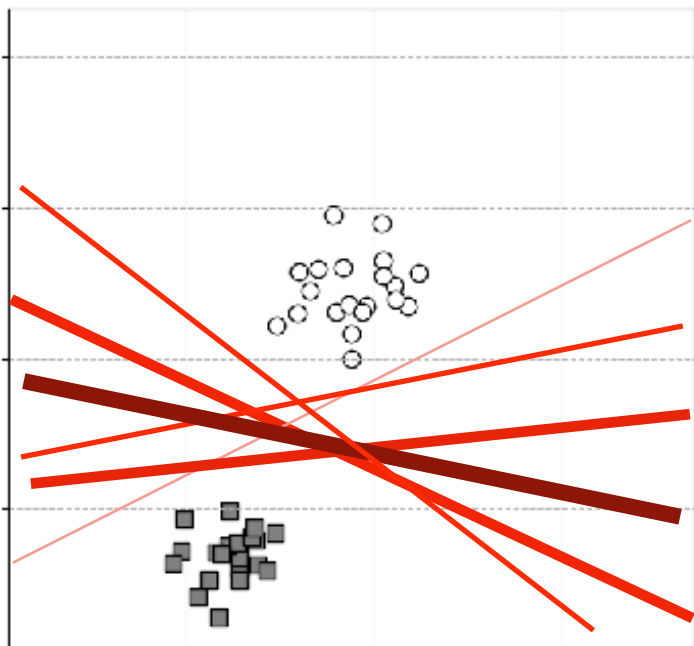
Bayesian Principles



(1) Keep your options open

$$p(\theta|\mathcal{D}_1) = \frac{p(\mathcal{D}_1|\theta)p(\theta)}{\int p(\mathcal{D}_1|\theta)p(\theta)d\theta}$$

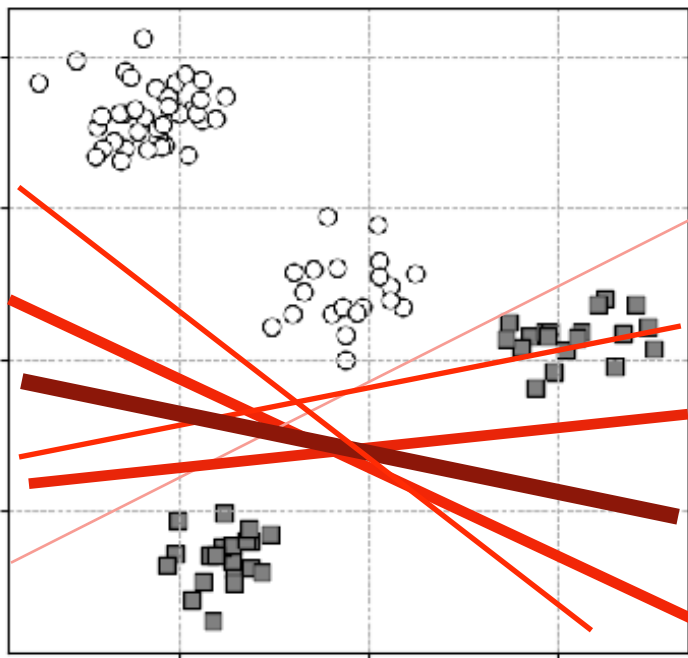
Bayesian Principles



(1) Keep your options open

$$p(\theta|\mathcal{D}_1) = \frac{p(\mathcal{D}_1|\theta)p(\theta)}{\int p(\mathcal{D}_1|\theta)p(\theta)d\theta}$$

Bayesian Principles

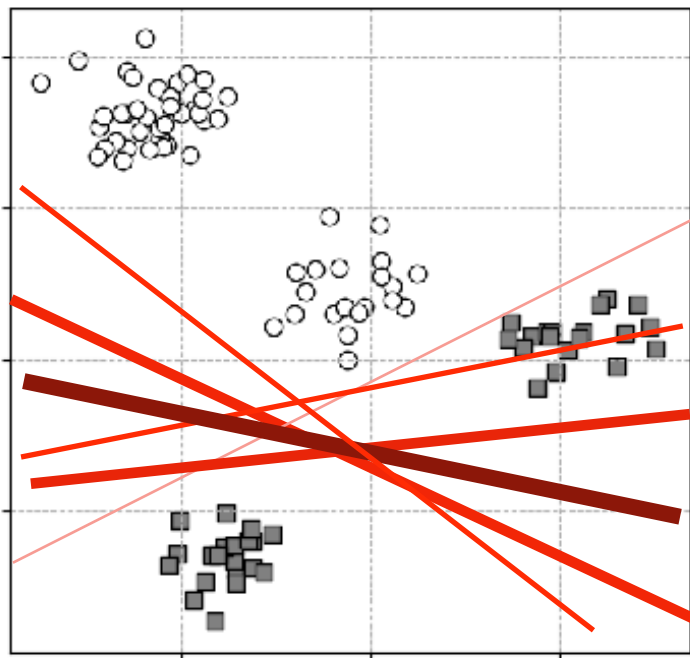


(1) Keep your options open

$$p(\theta|\mathcal{D}_1) = \frac{p(\mathcal{D}_1|\theta)p(\theta)}{\int p(\mathcal{D}_1|\theta)p(\theta)d\theta}$$

(2) Revise with new evidence

Bayesian Principles



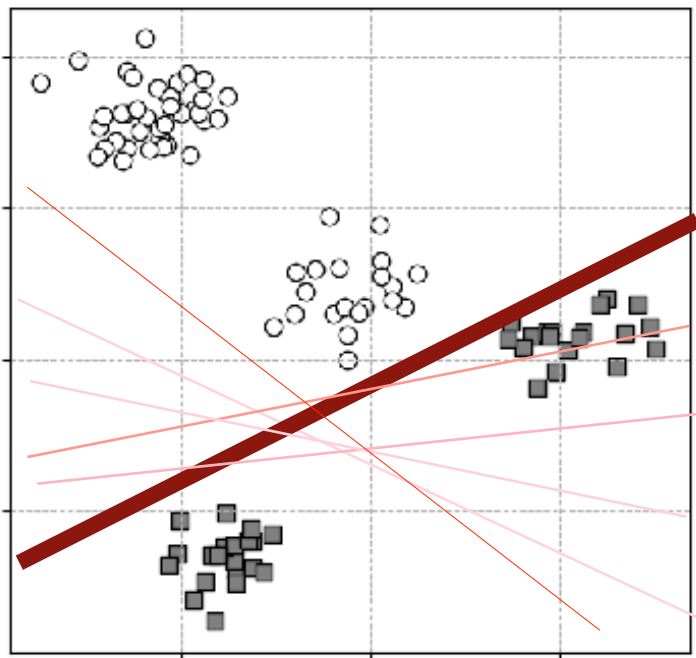
(1) Keep your options open

$$p(\theta|\mathcal{D}_1) = \frac{p(\mathcal{D}_1|\theta)p(\theta)}{\int p(\mathcal{D}_1|\theta)p(\theta)d\theta}$$

(2) Revise with new evidence

$$p(\theta|\mathcal{D}_2, \mathcal{D}_1) = \frac{p(\mathcal{D}_2|\theta)p(\theta|\mathcal{D}_1)}{\int p(\mathcal{D}_2|\theta)p(\theta|\mathcal{D}_1)d\theta}$$

Bayesian Principles



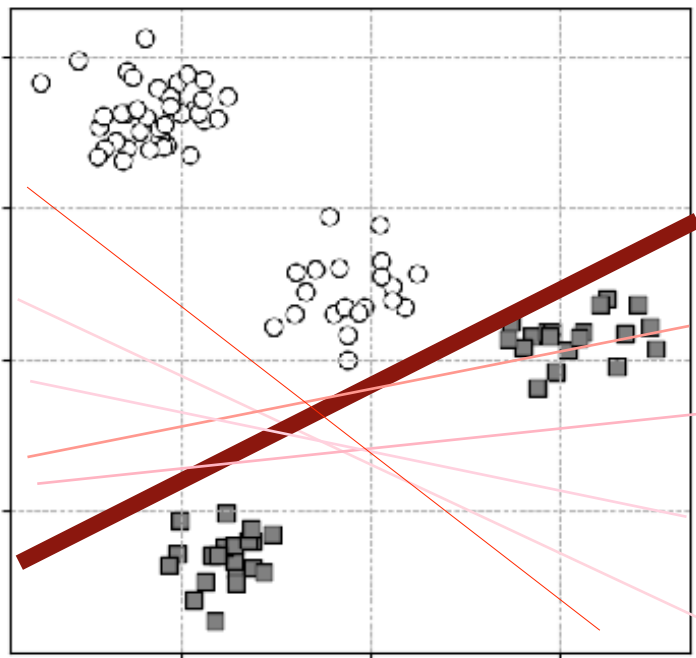
(1) Keep your options open

$$p(\theta|\mathcal{D}_1) = \frac{p(\mathcal{D}_1|\theta)p(\theta)}{\int p(\mathcal{D}_1|\theta)p(\theta)d\theta}$$

(2) Revise with new evidence

$$p(\theta|\mathcal{D}_2, \mathcal{D}_1) = \frac{p(\mathcal{D}_2|\theta)p(\theta|\mathcal{D}_1)}{\int p(\mathcal{D}_2|\theta)p(\theta|\mathcal{D}_1)d\theta}$$

Bayesian Principles



(1) Keep your options open

$$p(\theta|\mathcal{D}_1) = \frac{p(\mathcal{D}_1|\theta)p(\theta)}{\int p(\mathcal{D}_1|\theta)p(\theta)d\theta}$$

(2) Revise with new evidence

$$p(\theta|\mathcal{D}_2, \mathcal{D}_1) = \frac{p(\mathcal{D}_2|\theta)p(\theta|\mathcal{D}_1)}{\int p(\mathcal{D}_2|\theta)p(\theta|\mathcal{D}_1)d\theta}$$

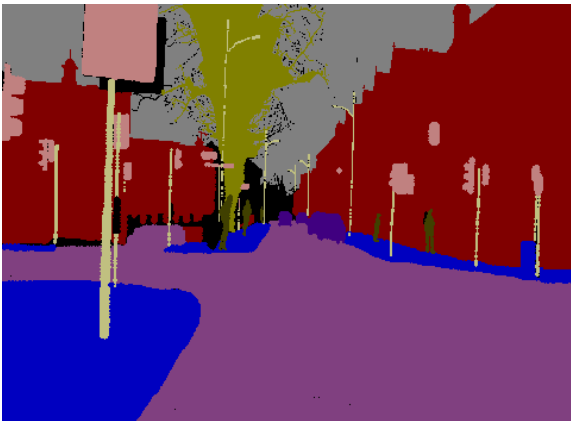
Similar ideas in sequential/online decision-making (uncertainty/randomization). **Computation is infeasible.**

Image Segmentation

Image



True Segments



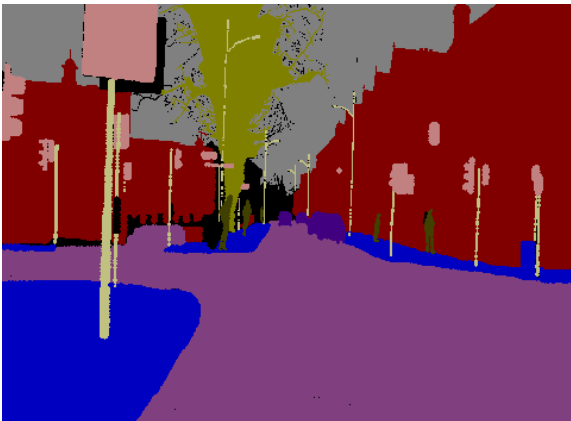
Kendall, Alex, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics." *CVPR*. 2018.

Image Segmentation

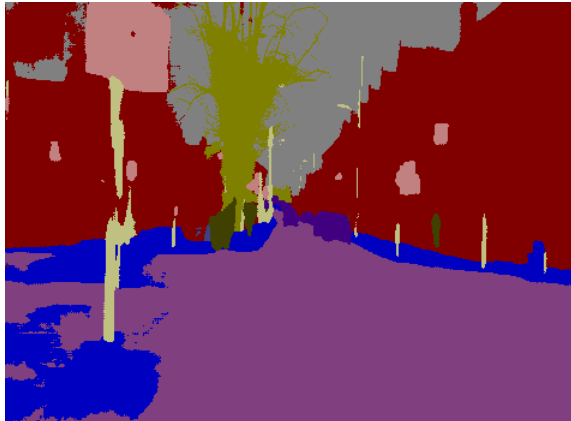
Image



True Segments



Prediction



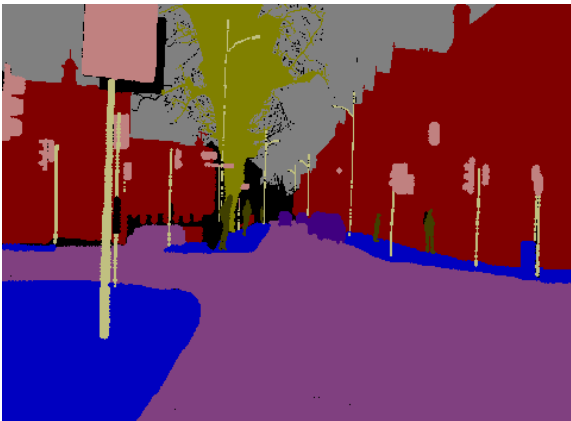
Kendall, Alex, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics." *CVPR*. 2018.

Image Segmentation

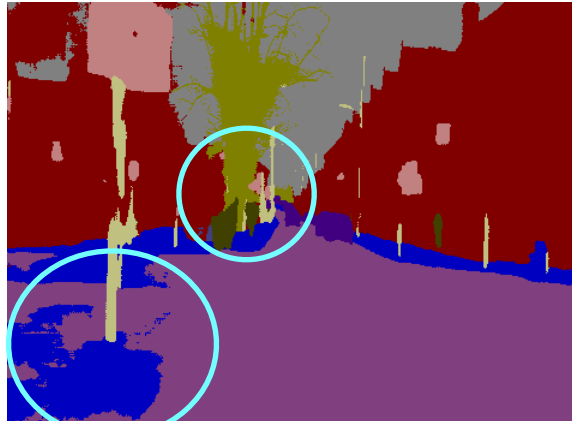
Image



True Segments



Prediction



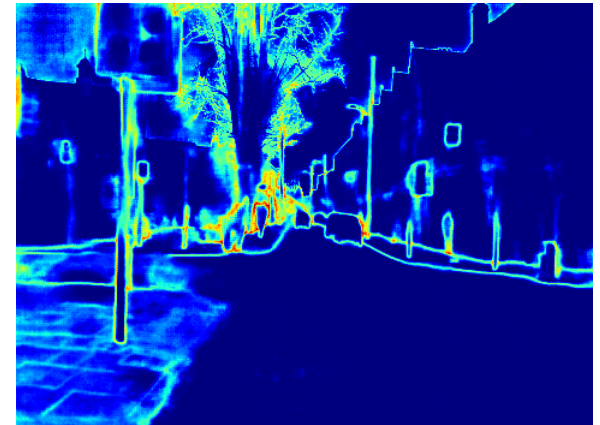
Kendall, Alex, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics." *CVPR*. 2018.

Image Segmentation

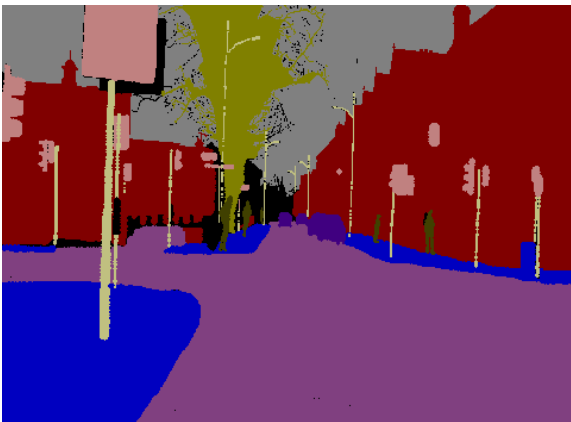
Image



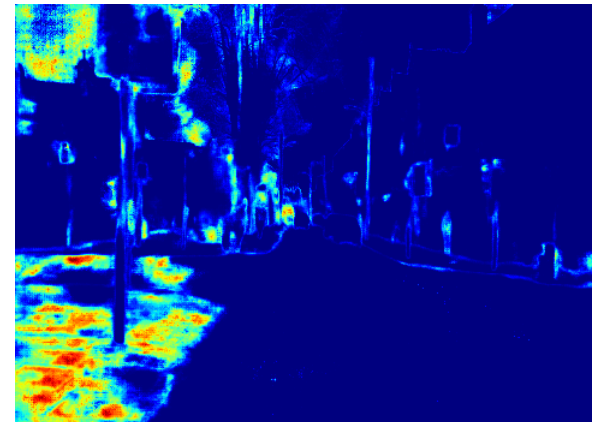
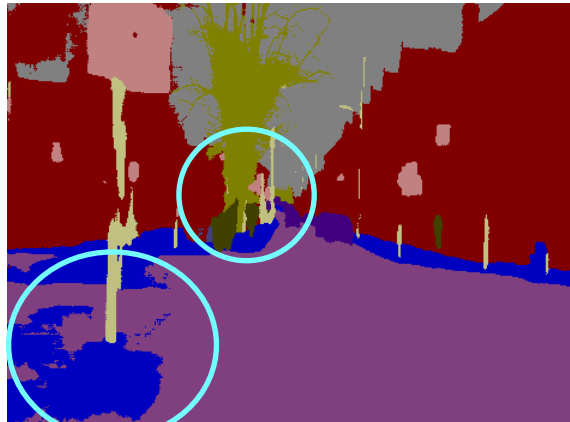
Uncertainty



True Segments



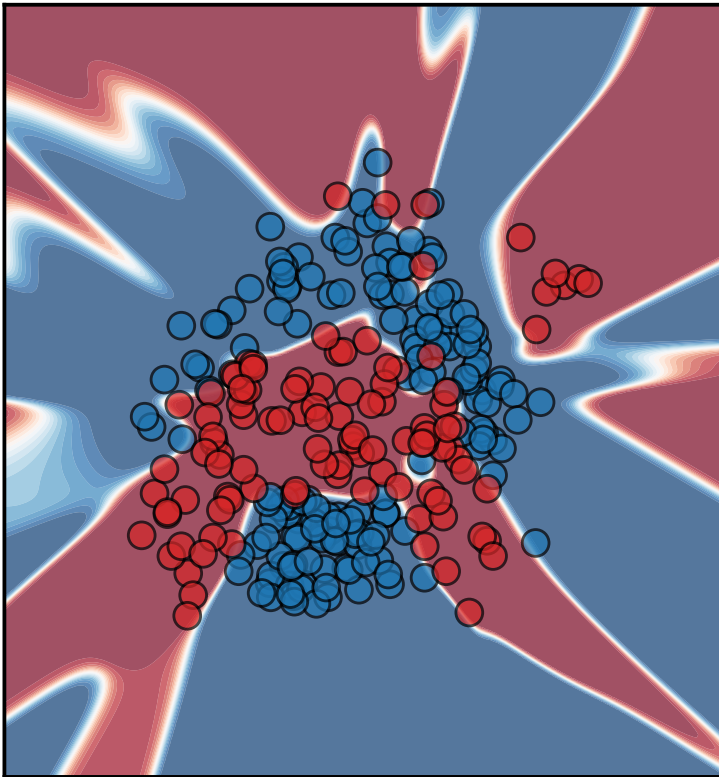
Prediction



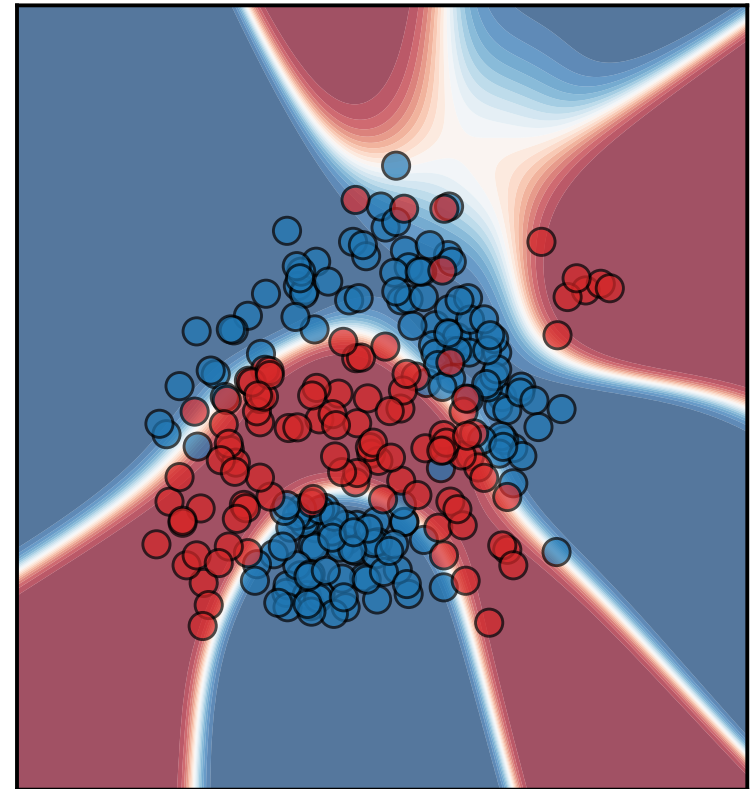
Kendall, Alex, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics." *CVPR*. 2018.

Reduce Overfitting

Standard DL



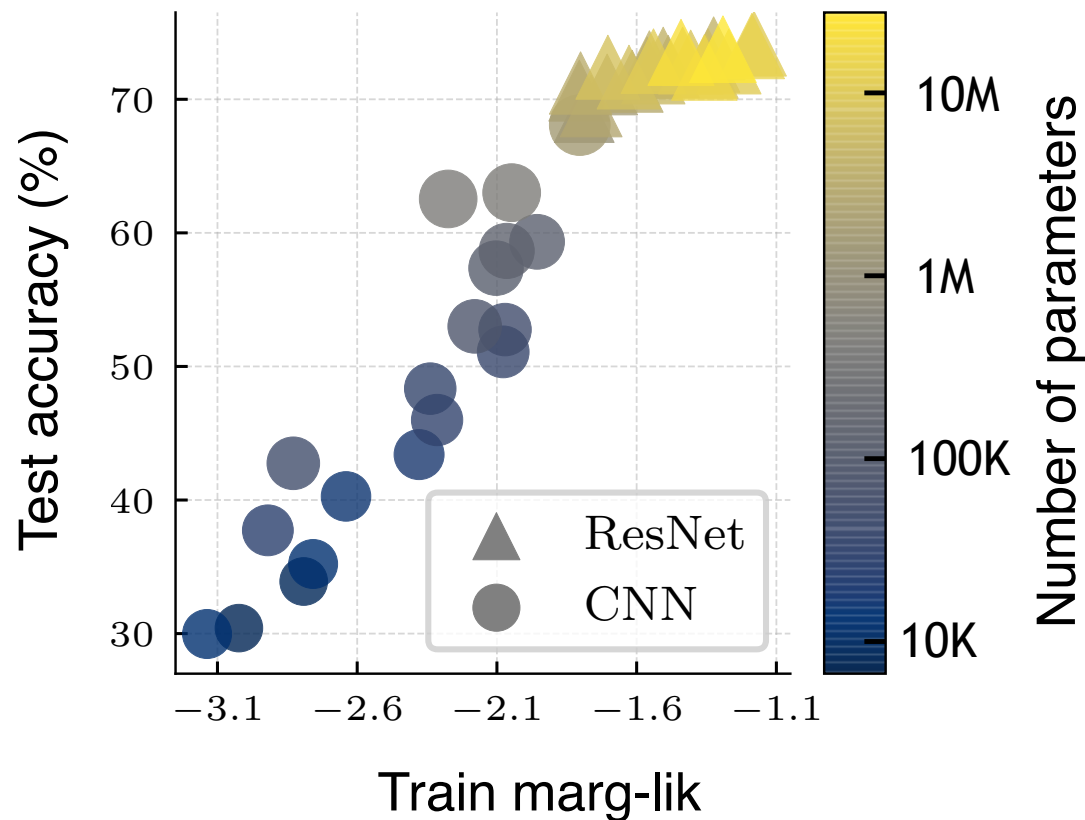
Bayesian DL



Left figure is cross-validation. Right figure is “Marginal Likelihood”.

Model selection without test set

The “training marginal-likelihood” can be used to select deep-nets, *without* requiring the test set.



Test-accuracy correlates with train marg-lik.

Both increase as the model size is increased.

On CIFAR-100, around 50 models are shown.

Bayesian learning

Not scalable

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

Deep learning

Scalable

$$\theta \leftarrow \theta - \rho H_{\theta}^{-1} \nabla_{\theta} \ell(\theta)$$

	Bayes	DL
Can handle large data and complex models?	✗	✓
Scalable training?	✗	✓
Can estimate uncertainty?	✓	✗
Can perform sequential / active /online / incremental learning?	✓	✗

Bayesian Principles

Inference as Optimization

Go beyond probabilistic models

To deep-learning models

Bayes Rule as Optimization

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

Bayes Rule as Optimization

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

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta)$$

Bayes Rule as Optimization

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

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta)$$

$$= \arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

All distribution
Distribution
Entropy

Bayes Rule as Optimization

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

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta)$$

$$= \arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

All distribution
Distribution
Entropy

$$= \mathbb{E}_q[\ell(\theta)] + \mathbb{E}_q[\log q(\theta)]$$

Bayes Rule as Optimization

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

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta)$$

$$= \arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

All distribution
Distribution
Entropy

$$= \mathbb{E}_q[\ell(\theta)] + \mathbb{E}_q[\log q(\theta)] = \mathbb{E}_q \left[\log \frac{q(\theta)}{e^{-\ell(\theta)}} \right]$$

Bayes Rule as Optimization

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

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta)$$

$$= \arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

All distribution
Distribution
Entropy

$$= \mathbb{E}_q[\ell(\theta)] + \mathbb{E}_q[\log q(\theta)] = \mathbb{E}_q \left[\log \frac{q(\theta)}{e^{-\ell(\theta)}} \right]$$

$$\implies q_*(\theta) \propto e^{-\ell(\theta)}$$

Bayes Rule as Optimization

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

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta)$$

$$= \arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

All distribution
Distribution
Entropy

$$= \mathbb{E}_q[\ell(\theta)] + \mathbb{E}_q[\log q(\theta)] = \mathbb{E}_q \left[\log \frac{q(\theta)}{e^{-\ell(\theta)}} \right]$$

$$\implies q_*(\theta) \propto e^{-\ell(\theta)} \propto p(\mathcal{D}|\theta)p(\theta)$$

Bayes Rule as Optimization

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

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta)$$

$$= \arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

All distribution
Distribution
Entropy

$$= \mathbb{E}_q[\ell(\theta)] + \mathbb{E}_q[\log q(\theta)] = \mathbb{E}_q \left[\log \frac{q(\theta)}{e^{-\ell(\theta)}} \right]$$

$$\implies q_*(\theta) \propto e^{-\ell(\theta)} \propto p(\mathcal{D}|\theta)p(\theta) \propto p(\theta|\mathcal{D})$$

Bayes Rule as Optimization

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

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta)$$

$$= \arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

All distribution
Distribution
Entropy

$$= \mathbb{E}_q[\ell(\theta)] + \mathbb{E}_q[\log q(\theta)] = \mathbb{E}_q \left[\log \frac{q(\theta)}{e^{-\ell(\theta)}} \right]$$


$$\implies q_*(\theta) \propto e^{-\ell(\theta)} \propto p(\mathcal{D}|\theta)p(\theta) \propto p(\theta|\mathcal{D})$$

Holds for any loss function (generalized-posterior)

Bayes Objective

$$\min_{\theta} \ell(\theta) \quad \text{vs} \quad \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q) \quad \text{Entropy}$$

Generalized-Posterior approximation

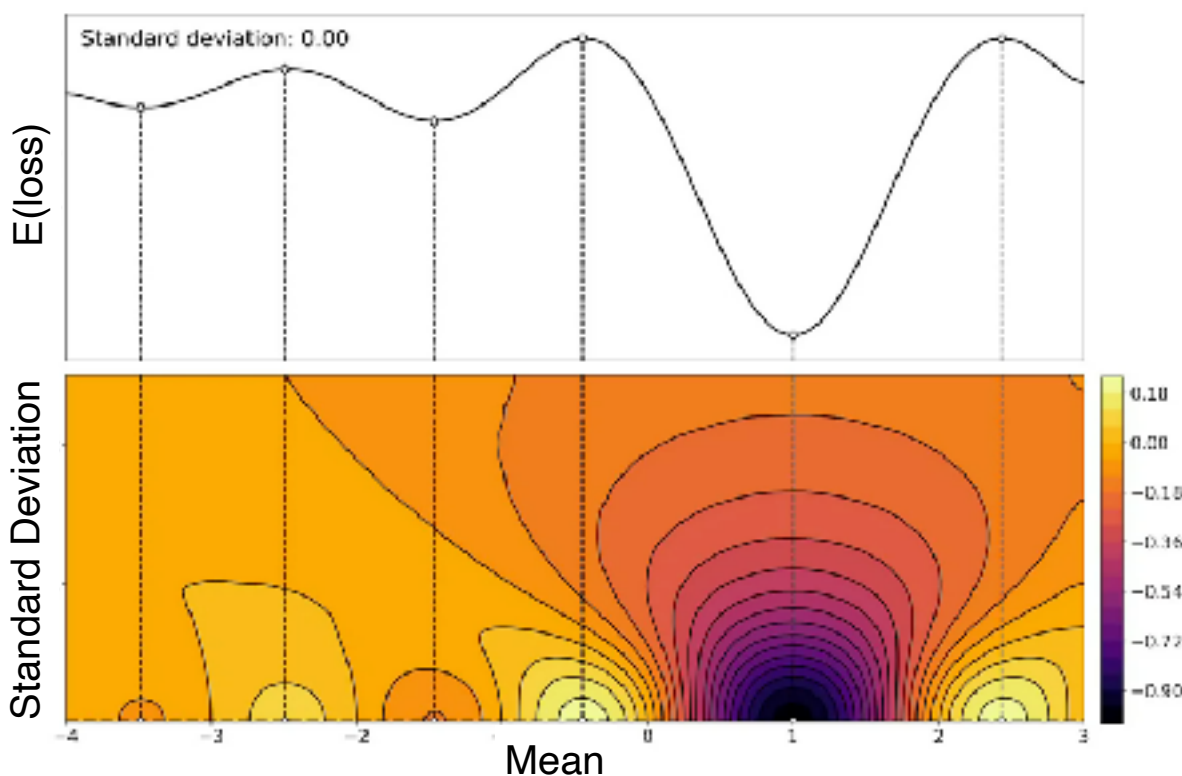


1. Zellner, A. "Optimal information processing and Bayes's theorem." *The American Statistician* (1988)

Bayes Objective

$$\min_{\theta} \ell(\theta) \quad \text{vs} \quad \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q) \quad \text{Entropy}$$

Generalized-Posterior approximation

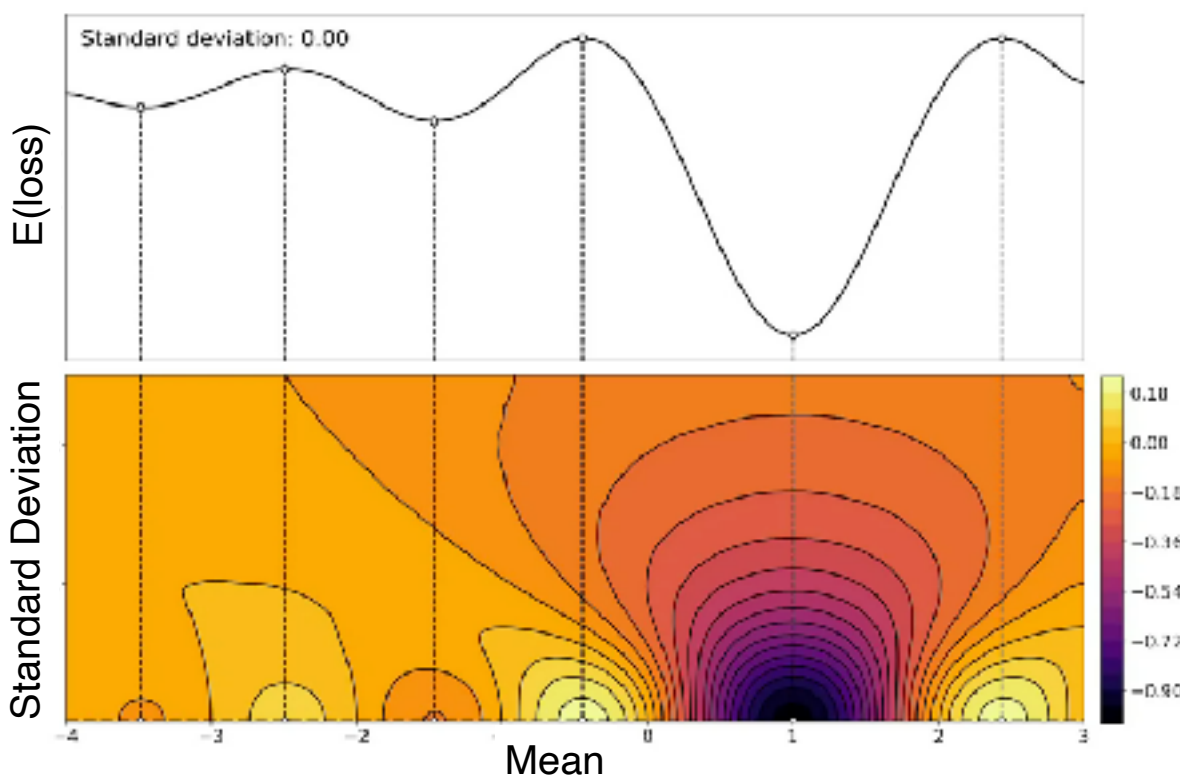


1. Zellner, A. "Optimal information processing and Bayes's theorem." *The American Statistician* (1988)
2. Huszar's blog, Evolution Strategies, Variational Optimisation and Natural ES (2017)
3. Khan et al. "Variational adaptive-Newton method for explorative learning." *arXiv* (2017).

Bayes Objective

$$\min_{\theta} \ell(\theta) \quad \text{vs} \quad \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q) \quad \text{Entropy}$$

Generalized-Posterior approximation

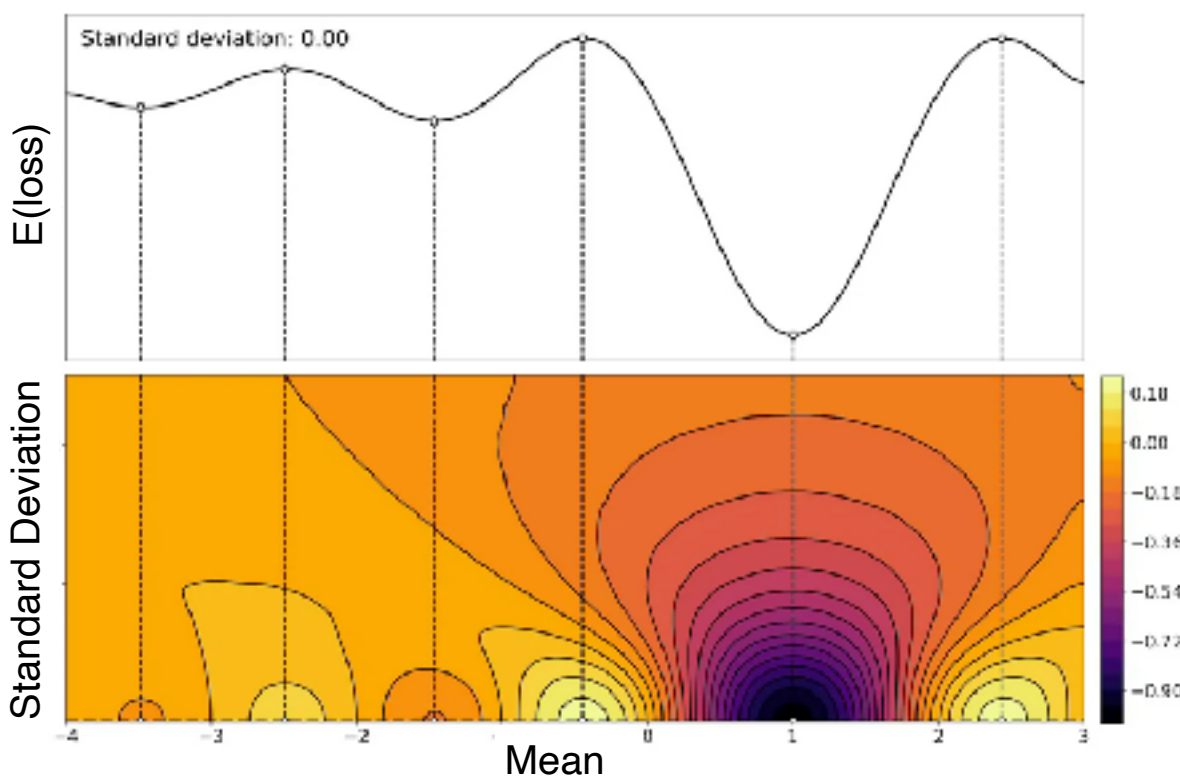


1. Zellner, A. "Optimal information processing and Bayes's theorem." *The American Statistician* (1988)
2. Huszar's blog, Evolution Strategies, Variational Optimisation and Natural ES (2017)
3. Khan et al. "Variational adaptive-Newton method for explorative learning." *arXiv* (2017).

Bayes Objective

$$\min_{\theta} \ell(\theta) \quad \text{vs} \quad \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q) \quad \text{Entropy}$$


Generalized-Posterior approximation

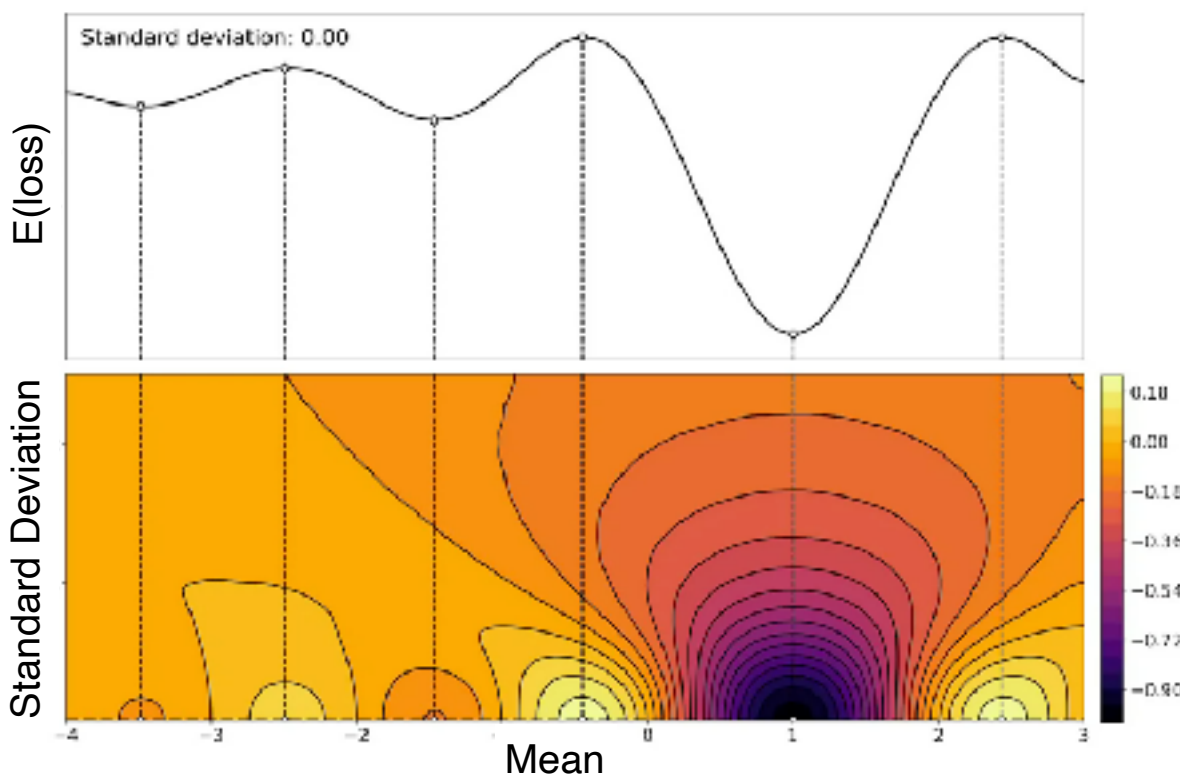


1. Zellner, A. "Optimal information processing and Bayes's theorem." *The American Statistician* (1988)
2. Huszar's blog, Evolution Strategies, Variational Optimisation and Natural ES (2017)
3. Khan et al. "Variational adaptive-Newton method for explorative learning." *arXiv* (2017).

Bayes Objective

$$\min_{\theta} \ell(\theta) \quad \text{vs} \quad \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q) \quad \text{Entropy}$$


 Generalized-Posterior approximation



Common in

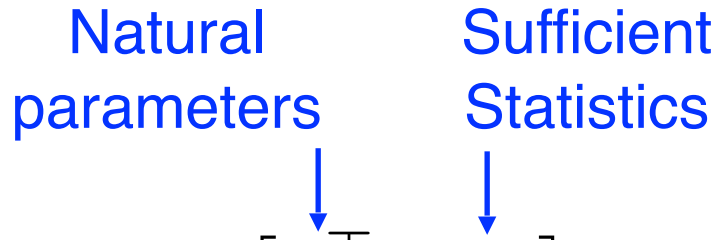
- Search
- Inference
- (Global) optimization
- Online learning
- Reinforcement learning

1. Zellner, A. "Optimal information processing and Bayes's theorem." *The American Statistician* (1988)
2. Huszar's blog, Evolution Strategies, Variational Optimisation and Natural ES (2017)
3. Khan et al. "Variational adaptive-Newton method for explorative learning." *arXiv* (2017).

Exponential Family Approximations

Natural
parameters

Sufficient
Statistics


$$q(\theta) \propto \exp [\lambda^{\top} T(\theta)]$$

Exponential Family Approximations

Natural
parameters

Sufficient
Statistics

$$q(\theta) \propto \exp \left[\lambda^\top T(\theta) \right]$$

$$\mathcal{N}(\theta|m, S^{-1}) \propto \exp \left[-\frac{1}{2}(\theta - m)^\top S(\theta - m) \right]$$

Exponential Family Approximations

Natural
parameters

Sufficient
Statistics

$$q(\theta) \propto \exp \left[\lambda^\top T(\theta) \right]$$

$$\begin{aligned} \mathcal{N}(\theta|m, S^{-1}) &\propto \exp \left[-\frac{1}{2}(\theta - m)^\top S(\theta - m) \right] \\ &\propto \exp \left[(Sm)^\top \theta + \text{Tr} \left(-\frac{S}{2} \theta \theta^\top \right) \right] \end{aligned}$$

Exponential Family Approximations

Natural
parameters

Sufficient
Statistics

Expectation
parameters

$$q(\theta) \propto \exp \left[\lambda^\top T(\theta) \right] \qquad \mu := \mathbb{E}_q[T(\theta)]$$

$$\begin{aligned} \mathcal{N}(\theta|m, S^{-1}) &\propto \exp \left[-\frac{1}{2}(\theta - m)^\top S(\theta - m) \right] \\ &\propto \exp \left[(Sm)^\top \theta + \text{Tr} \left(-\frac{S}{2} \theta \theta^\top \right) \right] \end{aligned}$$

Exponential Family Approximations

Natural
parameters

Sufficient
Statistics

Expectation
parameters

$$q(\theta) \propto \exp \left[\lambda^\top T(\theta) \right] \quad \mu := \mathbb{E}_q[T(\theta)]$$

$$\begin{aligned} \mathcal{N}(\theta|m, S^{-1}) &\propto \exp \left[-\frac{1}{2}(\theta - m)^\top S(\theta - m) \right] \\ &\propto \exp \left[(Sm)^\top \theta + \text{Tr} \left(-\frac{S}{2} \theta \theta^\top \right) \right] \end{aligned}$$

Gaussian distribution

$$q(\theta) := \mathcal{N}(\theta|m, S^{-1})$$

Natural parameters

$$\lambda := \{Sm, -S/2\}$$

Expectation parameters

$$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta \theta^\top)\}$$

Solutions of Bayes Objective

A fundamental equation

$$\nabla_{\mu} H(q_*) = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$$

Optimal approx Natural gradient
↓
Entropy

Solutions of Bayes Objective

A fundamental equation

$$\nabla_{\mu} \underset{\text{Entropy}}{H(q_*)} = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$$

Optimal approx Natural gradient

For minimal Exp-Family

$$-\lambda_* = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$$

Solutions of Bayes Objective

A fundamental equation

$$\nabla_{\mu} H(q_*) = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$$

Optimal approx Natural gradient
↓
Entropy

For minimal Exp-Family

$$-\lambda_* = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$$

Information matching due to the entropy term

1. Natural gradients contain essential higher-order information about the loss landscape
2. These are assigned to appropriate natural params

Solutions of Bayes Objective

A fundamental equation

$$\nabla_{\mu} H(q_*) = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$$

Optimal approx \downarrow Natural gradient
Entropy

For minimal Exp-Family

$$-\lambda_* = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$$

Information matching due to the entropy term

1. Natural gradients contain essential higher-order information about the loss landscape
2. These are assigned to appropriate natural params

The importance of this equation is “entirely missed in the Bayesian machine-learning community, including books, reviews, and tutorial on this topic”

A simple example

$$\nabla_{\mu} H(q_*) = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$$

Bayesian Learning Rule

Unify, generalize, and improve
machine-learning algorithms

Bayesian Learning Rule

$$\min_{\theta} \ell(\theta) \quad \text{vs} \quad \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

← Exponential-family Approx.

Deep Learning algo: $\theta \leftarrow \theta - \rho H_{\theta}^{-1} \nabla_{\theta} \ell(\theta)$

Bayesian Learning Rule

$$\min_{\theta} \ell(\theta) \quad \text{vs} \quad \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

Exponential-family Approx.

Deep Learning algo: $\theta \leftarrow \theta - \rho H_{\theta}^{-1} \nabla_{\theta} \ell(\theta)$

Bayes learning rule: $\lambda \leftarrow \lambda - \rho \nabla_{\mu} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$

An upward blue arrow points from the label "Current parameters" to the symbol λ. Another upward blue arrow points from the label "Step size" to the Greek letter ρ. A blue arrow points from the label "Natural Gradient" to the subscript μ.

Natural and Expectation parameters of an exponential family distribution q (natural-gradient descent & mirror descent)

Bayesian Learning Rule

$$\min_{\theta} \ell(\theta) \quad \text{vs} \quad \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

Exponential-family Approx.

Deep Learning algo: $\theta \leftarrow \theta - \rho H_{\theta}^{-1} \nabla_{\theta} \ell(\theta)$

Bayes learning rule: $\lambda \leftarrow \lambda - \rho \nabla_{\mu} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$

\uparrow \uparrow \nwarrow Natural Gradient

Natural and Expectation parameters of an
exponential family distribution q
(natural-gradient descent & mirror descent)

By changing \mathcal{Q} , we can recover DL algorithms (and more)

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation \longleftrightarrow Learning-Algorithm

Complex \longleftrightarrow Simple

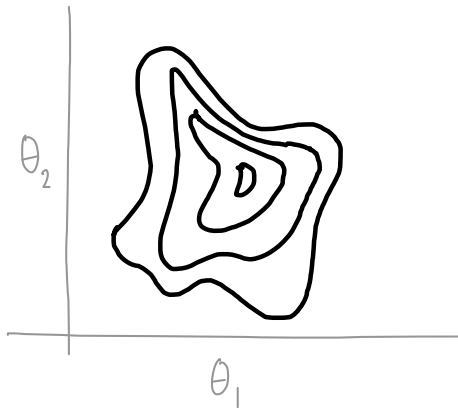
Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation \longleftrightarrow Learning-Algorithm

Complex



Simple



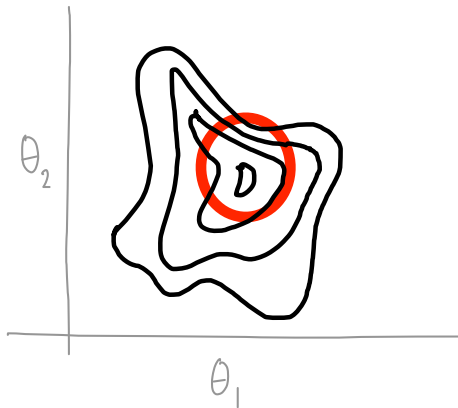
Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation \longleftrightarrow Learning-Algorithm

Complex



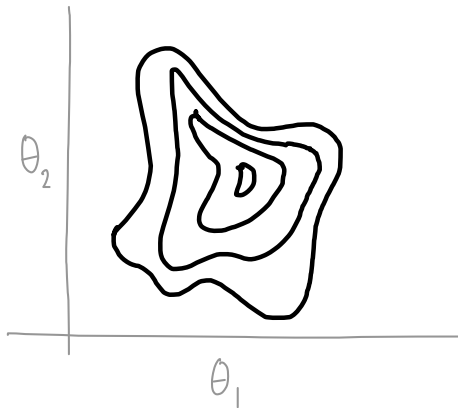
Simple



Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation \longleftrightarrow Learning-Algorithm

Complex



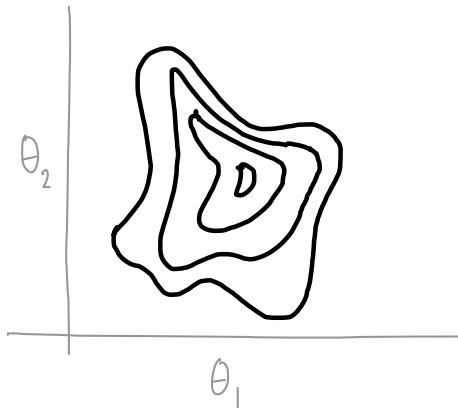
Simple



Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation \longleftrightarrow Learning-Algorithm

Complex



Simple

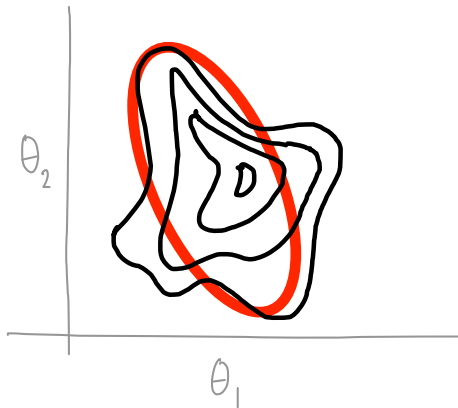


Gradient
Descent

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation \longleftrightarrow Learning-Algorithm

Complex



Simple

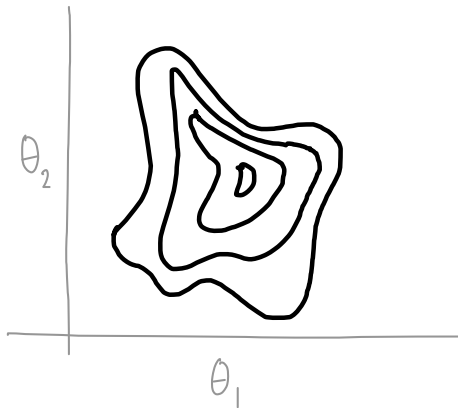


Gradient
Descent

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation \longleftrightarrow Learning-Algorithm

Complex



Simple

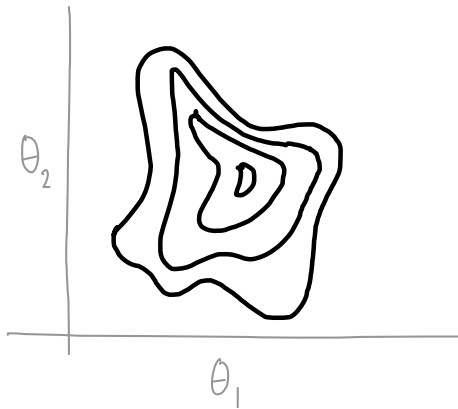


Gradient
Descent

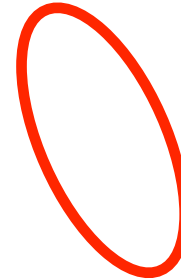
Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation \longleftrightarrow Learning-Algorithm

Complex



Simple

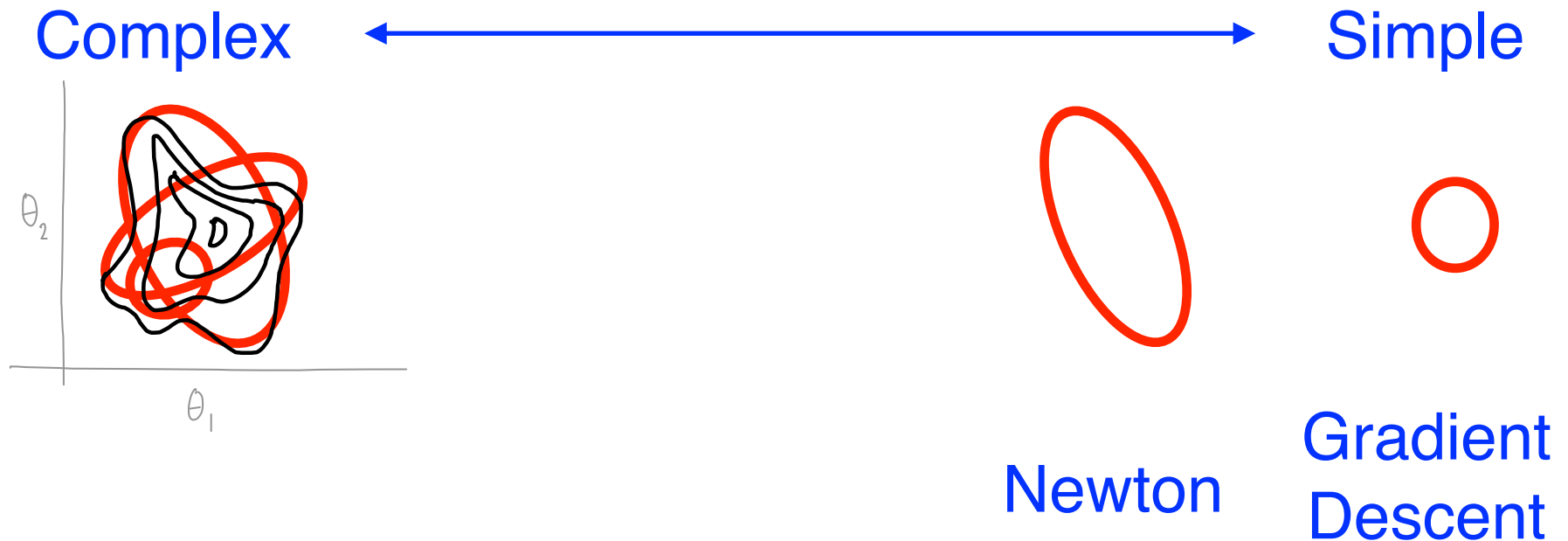


Newton

Gradient
Descent

Deriving Learning-Algorithms from the Bayesian Learning Rule

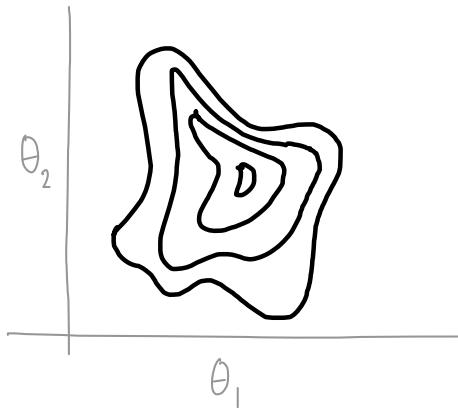
Posterior Approximation \longleftrightarrow Learning-Algorithm



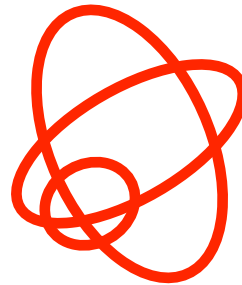
Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation \longleftrightarrow Learning-Algorithm

Complex



Simple

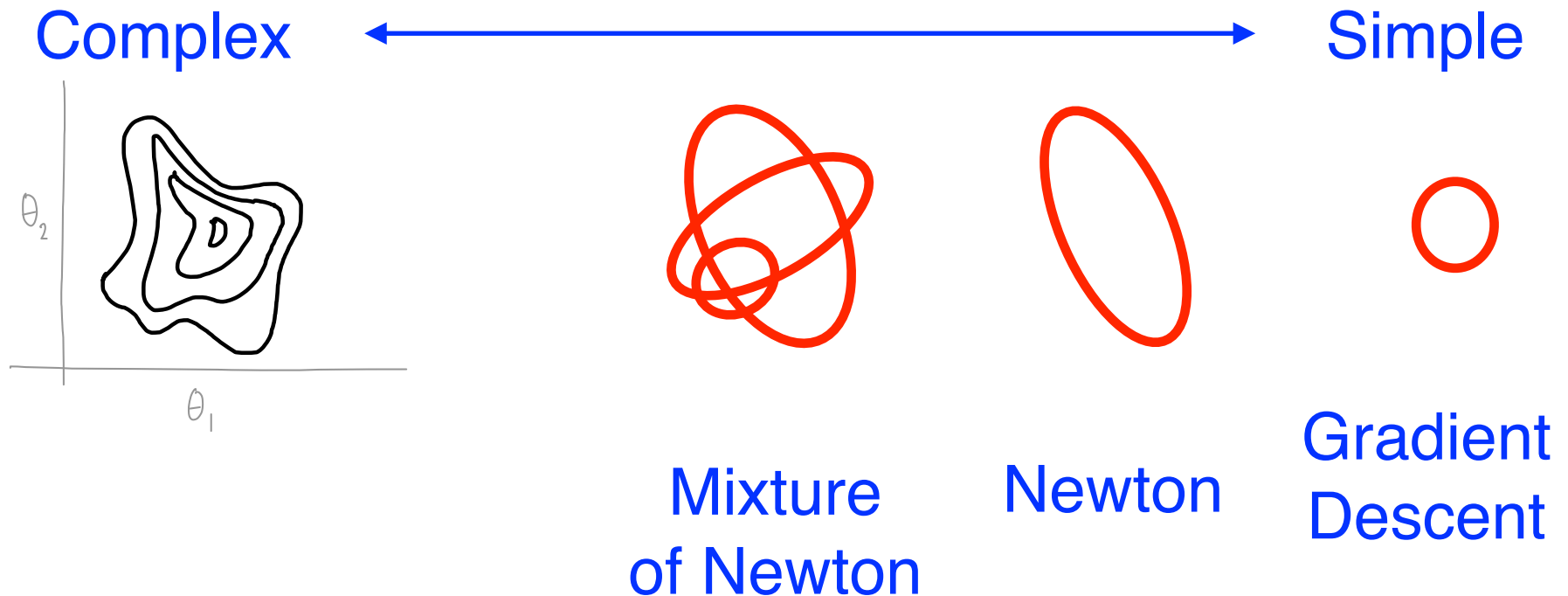


Newton

Gradient
Descent

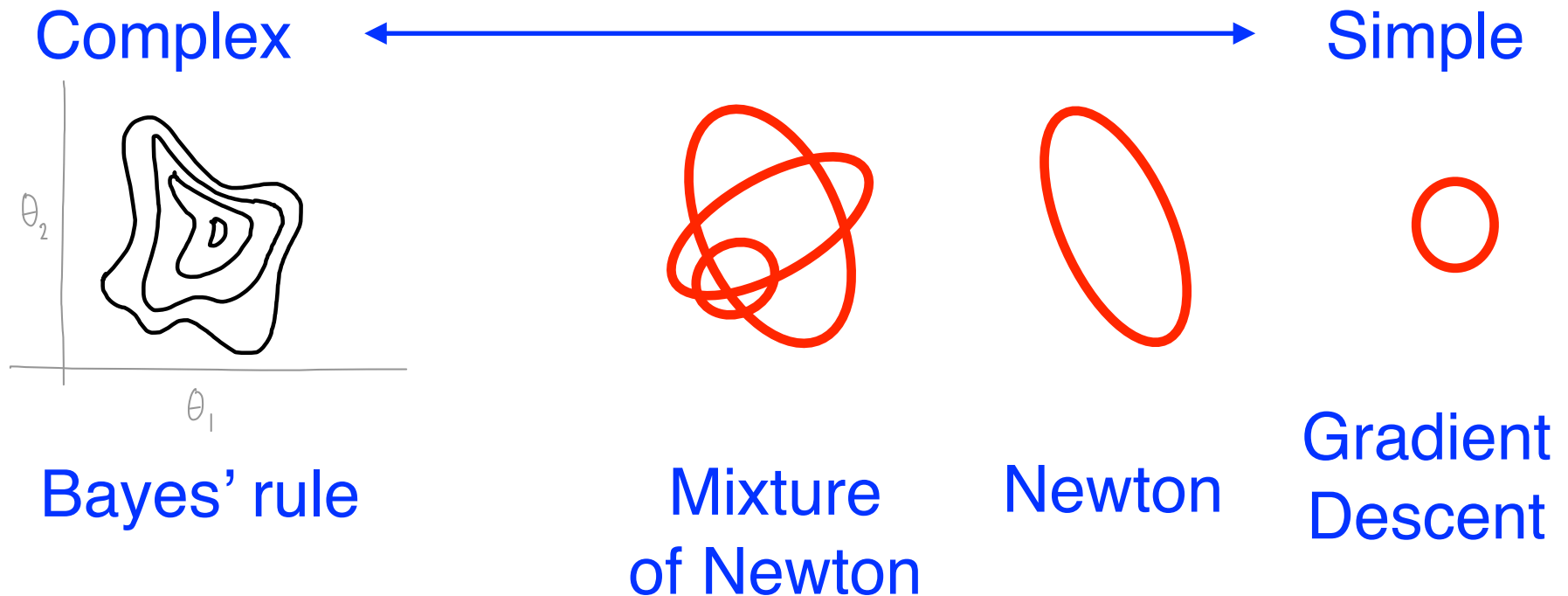
Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation \longleftrightarrow Learning-Algorithm



Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation \longleftrightarrow Learning-Algorithm



Gradient Descent from Bayes

Gradient descent: $\theta \leftarrow \theta - \rho \nabla_{\theta} \ell(\theta)$

Gradient Descent from Bayes

Gradient descent: $\theta \leftarrow \theta - \rho \nabla_{\theta} \ell(\theta)$

Derived by choosing **Gaussian with fixed covariance**

Gaussian distribution $q(\theta) := \mathcal{N}(m, 1)$

Natural parameters $\lambda := m$

Expectation parameters $\mu := \mathbb{E}_q[\theta] = m$

Entropy $\mathcal{H}(q) := \log(2\pi)/2$

Gradient Descent from Bayes

Gradient descent: $\theta \leftarrow \theta - \rho \nabla_{\theta} \ell(\theta)$

$$\lambda \leftarrow \lambda - \rho \nabla_{\mu} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$$

Derived by choosing **Gaussian with fixed covariance**

Gaussian distribution $q(\theta) := \mathcal{N}(m, 1)$

Natural parameters $\lambda := m$

Expectation parameters $\mu := \mathbb{E}_q[\theta] = m$

Entropy $\mathcal{H}(q) := \log(2\pi)/2$

Gradient Descent from Bayes

Gradient descent: $\theta \leftarrow \theta - \rho \nabla_{\theta} \ell(\theta)$

$$m \leftarrow m - \rho \nabla_{\textcolor{red}{m}} \mathbb{E}_q[\ell(\theta)]$$

$$\lambda \leftarrow \lambda - \rho \nabla_{\textcolor{red}{\mu}} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$$

Derived by choosing **Gaussian with fixed covariance**

Gaussian distribution $q(\theta) := \mathcal{N}(m, 1)$

Natural parameters $\lambda := m$

Expectation parameters $\mu := \mathbb{E}_q[\theta] = m$

Entropy $\mathcal{H}(q) := \log(2\pi)/2$

Gradient Descent from Bayes

Gradient descent: $\theta \leftarrow \theta - \rho \nabla_{\theta} \ell(\theta)$

“Global” to “local”
(the delta method)

$$\mathbb{E}_q[\ell(\theta)] \approx \ell(m)$$

$$m \leftarrow m - \rho \nabla_{\textcolor{red}{m}} \mathbb{E}_q[\ell(\theta)]$$

$$\lambda \leftarrow \lambda - \rho \nabla_{\textcolor{red}{\mu}} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$$

Derived by choosing **Gaussian with fixed covariance**

Gaussian distribution $q(\theta) := \mathcal{N}(m, 1)$

Natural parameters $\lambda := m$

Expectation parameters $\mu := \mathbb{E}_q[\theta] = m$

Entropy $\mathcal{H}(q) := \log(2\pi)/2$

Gradient Descent from Bayes

Gradient descent: $\theta \leftarrow \theta - \rho \nabla_{\theta} \ell(\theta)$

Bayes Learn Rule: $m \leftarrow m - \rho \nabla_m \ell(m)$

“Global” to “local”
(the delta method)

$$\mathbb{E}_q[\ell(\theta)] \approx \ell(m)$$

$$m \leftarrow m - \rho \nabla_{\textcolor{red}{m}} \mathbb{E}_q[\ell(\theta)]$$

$$\lambda \leftarrow \lambda - \rho \nabla_{\textcolor{red}{\mu}} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$$

Derived by choosing **Gaussian with fixed covariance**

Gaussian distribution $q(\theta) := \mathcal{N}(m, 1)$

Natural parameters $\lambda := m$

Expectation parameters $\mu := \mathbb{E}_q[\theta] = m$

Entropy $\mathcal{H}(q) := \log(2\pi)/2$

Bayesian learning rule: $\lambda \leftarrow \lambda - \rho \nabla_{\mu} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$

We can compute uncertainty using a variant of Adam.

Learning Algorithm	Posterior Approx.	Natural-Gradient Approx.	Sec.
Optimization Algorithms			
Gradient Descent	Gaussian (fixed cov.)	Delta method	1.3
Newton's method	Gaussian	—"—"	1.3
Multimodal optimization _(New)	Mixture of Gaussians	—"—"	3.2
Deep-Learning Algorithms			
Stochastic Gradient Descent	Gaussian (fixed cov.)	Delta method, stochastic approx.	4.1
RMSprop/Adam	Gaussian (diagonal cov.)	Delta method, stochastic approx., Hessian approx., square-root scaling, slow-moving scale vectors	4.2
Dropout	Mixture of Gaussians	Delta method, stochastic approx., responsibility approx.	4.3
STE	Bernoulli	Delta method, stochastic approx.	4.5
Online Gauss-Newton (OGN) _(New)	Gaussian (diagonal cov.)	Gauss-Newton Hessian approx. in Adam & no square-root scaling	4.4
Variational OGN _(New)	—"—"	Remove delta method from OGN	4.4
BayesBiNN _(New)	Bernoulli	Remove delta method from STE	4.5
Approximate Bayesian Inference Algorithms			
Conjugate Bayes	Exp-family	Set learning rate $\rho_t = 1$	5.1
Laplace's method	Gaussian	Delta method	4.4
Expectation-Maximization	Exp-Family + Gaussian	Delta method for the parameters	5.2
Stochastic VI (SVI)	Exp-family (mean-field)	Stochastic approx., local $\rho_t = 1$	5.3
VMP	—"—"	$\rho_t = 1$ for all nodes	5.3
Non-Conjugate VMP	—"—"	—"—"	5.3
Non-Conjugate VI _(New)	Mixture of Exp-family	None	5.4

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. "Practical Deep Learning with Bayesian Principles." *NeurIPS* (2019).

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

Derived by choosing a **multivariate Gaussian**

Gaussian distribution $q(\theta) := \mathcal{N}(\theta|m, S^{-1})$

Natural parameters $\lambda := \{Sm, -S/2\}$

Expectation parameters $\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^{\top})\}$

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

$$\lambda \leftarrow \lambda - \rho \nabla_{\mu} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$$

Derived by choosing a **multivariate Gaussian**

Gaussian distribution $q(\theta) := \mathcal{N}(\theta|m, S^{-1})$

Natural parameters $\lambda := \{Sm, -S/2\}$

Expectation parameters $\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^{\top})\}$

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

$$\lambda \leftarrow \lambda - \rho \nabla_{\mu} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q)) \quad \boxed{-\nabla_{\mu} \mathcal{H}(q) = \lambda}$$

Derived by choosing a **multivariate Gaussian**

Gaussian distribution $q(\theta) := \mathcal{N}(\theta|m, S^{-1})$

Natural parameters $\lambda := \{Sm, -S/2\}$

Expectation parameters $\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^{\top})\}$

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

$$\lambda \leftarrow \lambda - \rho (\nabla_{\mu} \mathbb{E}_q[\ell(\theta)] + \lambda) \quad \boxed{-\nabla_{\mu} \mathcal{H}(q) = \lambda}$$

Derived by choosing a **multivariate Gaussian**

Gaussian distribution $q(\theta) := \mathcal{N}(\theta|m, S^{-1})$

Natural parameters $\lambda := \{Sm, -S/2\}$

Expectation parameters $\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^{\top})\}$

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

$$\lambda \leftarrow (1 - \rho)\lambda - \rho \nabla_{\mu} \mathbb{E}_q[\ell(\theta)]$$

$$-\nabla_{\mu} \mathcal{H}(q) = \lambda$$

Derived by choosing a **multivariate Gaussian**

Gaussian distribution $q(\theta) := \mathcal{N}(\theta|m, S^{-1})$

Natural parameters $\lambda := \{Sm, -S/2\}$

Expectation parameters $\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^{\top})\}$

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

$$Sm \leftarrow (1 - \rho)Sm - \rho \nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)]$$

$$\lambda \leftarrow (1 - \rho)\lambda - \rho \nabla_{\mu} \mathbb{E}_q[\ell(\theta)]$$

$$-\nabla_{\mu} \mathcal{H}(q) = \lambda$$

Derived by choosing a **multivariate Gaussian**

Gaussian distribution $q(\theta) := \mathcal{N}(\theta|m, S^{-1})$

Natural parameters $\lambda := \{Sm, -S/2\}$

Expectation parameters $\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^{\top})\}$

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

$$\begin{aligned} Sm &\leftarrow (1 - \rho)Sm - \rho \nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)] \\ -\frac{1}{2}S &\leftarrow -(1 - \rho)\frac{1}{2}S + \rho \nabla_{\mathbb{E}_q(\theta\theta^{\top})} \mathbb{E}_q[\ell(\theta)] \end{aligned}$$

$$\lambda \leftarrow (1 - \rho)\lambda - \rho \nabla_{\mu} \mathbb{E}_q[\ell(\theta)]$$

$$-\nabla_{\mu} \mathcal{H}(q) = \lambda$$

Derived by choosing a **multivariate Gaussian**

Gaussian distribution $q(\theta) := \mathcal{N}(\theta|m, S^{-1})$

Natural parameters $\lambda := \{Sm, -S/2\}$

Expectation parameters $\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^{\top})\}$

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

$$Sm \leftarrow (1 - \rho)Sm - \rho \nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)]$$

$$S \leftarrow (1 - \rho)S - \rho 2 \nabla_{\mathbb{E}_q(\theta\theta^{\top})} \mathbb{E}_q[\ell(\theta)]$$

$$\lambda \leftarrow (1 - \rho)\lambda - \rho \nabla_{\mu} \mathbb{E}_q[\ell(\theta)]$$

$$-\nabla_{\mu} \mathcal{H}(q) = \lambda$$

Derived by choosing a **multivariate Gaussian**

Gaussian distribution $q(\theta) := \mathcal{N}(\theta|m, S^{-1})$

Natural parameters $\lambda := \{Sm, -S/2\}$

Expectation parameters $\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^{\top})\}$

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

$$\begin{aligned} S_m &\leftarrow (1 - \rho) S_m - \rho \nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)] \\ S &\leftarrow (1 - \rho) S - \rho 2 \nabla_{\mathbb{E}_q(\theta \theta^{\top})} \mathbb{E}_q[\ell(\theta)] \end{aligned}$$

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

Express in terms of gradient and Hessian of loss:

$$\nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)] = \mathbb{E}_q[\nabla_{\theta} \ell(\theta)] - 2\mathbb{E}_q[H_{\theta}]m$$

$$\nabla_{\mathbb{E}_q(\theta\theta^{\top})} \mathbb{E}_q[\ell(\theta)] = \mathbb{E}_q[H_{\theta}]$$

$$\begin{aligned} Sm &\leftarrow (1 - \rho)Sm - \rho \nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)] \\ S &\leftarrow (1 - \rho)S - \rho 2 \nabla_{\mathbb{E}_q(\theta\theta^{\top})} \mathbb{E}_q[\ell(\theta)] \end{aligned}$$

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

Delta Method

$$\mathbb{E}_q[\ell(\theta)] \approx \ell(m)$$

Express in terms of gradient and Hessian of loss:

$$\nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)] = \mathbb{E}_q[\nabla_{\theta} \ell(\theta)] - 2\mathbb{E}_q[H_{\theta}]m$$

$$\nabla_{\mathbb{E}_q(\theta\theta^{\top})} \mathbb{E}_q[\ell(\theta)] = \mathbb{E}_q[H_{\theta}]$$

$$\begin{aligned} Sm &\leftarrow (1 - \rho)Sm - \rho \nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)] \\ S &\leftarrow (1 - \rho)S - \rho 2 \nabla_{\mathbb{E}_q(\theta\theta^{\top})} \mathbb{E}_q[\ell(\theta)] \end{aligned}$$

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

$$\begin{aligned} m &\leftarrow m - \rho \mathbf{S}^{-1} \nabla_m \ell(m) \\ \mathbf{S} &\leftarrow (1 - \rho) \mathbf{S} + \rho \mathbf{H}_m \end{aligned}$$

Delta Method
 $\mathbb{E}_q[\ell(\theta)] \approx \ell(m)$

Express in terms of gradient and Hessian of loss:

$$\nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)] = \mathbb{E}_q[\nabla_{\theta} \ell(\theta)] - 2\mathbb{E}_q[\mathbf{H}_{\theta}]m$$

$$\nabla_{\mathbb{E}_q(\theta\theta^{\top})} \mathbb{E}_q[\ell(\theta)] = \mathbb{E}_q[\mathbf{H}_{\theta}]$$

$$\begin{aligned} \mathbf{S}m &\leftarrow (1 - \rho) \mathbf{S}m - \rho \nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)] \\ \mathbf{S} &\leftarrow (1 - \rho) \mathbf{S} - \rho 2 \nabla_{\mathbb{E}_q(\theta\theta^{\top})} \mathbb{E}_q[\ell(\theta)] \end{aligned}$$

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_{\theta}^{-1} [\nabla_{\theta} \ell(\theta)]$

Set $\rho=1$ to get $m \leftarrow m - H_m^{-1} [\nabla_m \ell(m)]$

$$m \leftarrow m - \rho \mathbf{S}^{-1} \nabla_m \ell(m)$$

$$\mathbf{S} \leftarrow (1 - \rho) \mathbf{S} + \rho \mathbf{H}_m$$

Delta Method

$$\mathbb{E}_q[\ell(\theta)] \approx \ell(m)$$

Express in terms of gradient and Hessian of loss:

$$\nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)] = \mathbb{E}_q[\nabla_{\theta} \ell(\theta)] - 2\mathbb{E}_q[\mathbf{H}_{\theta}]m$$

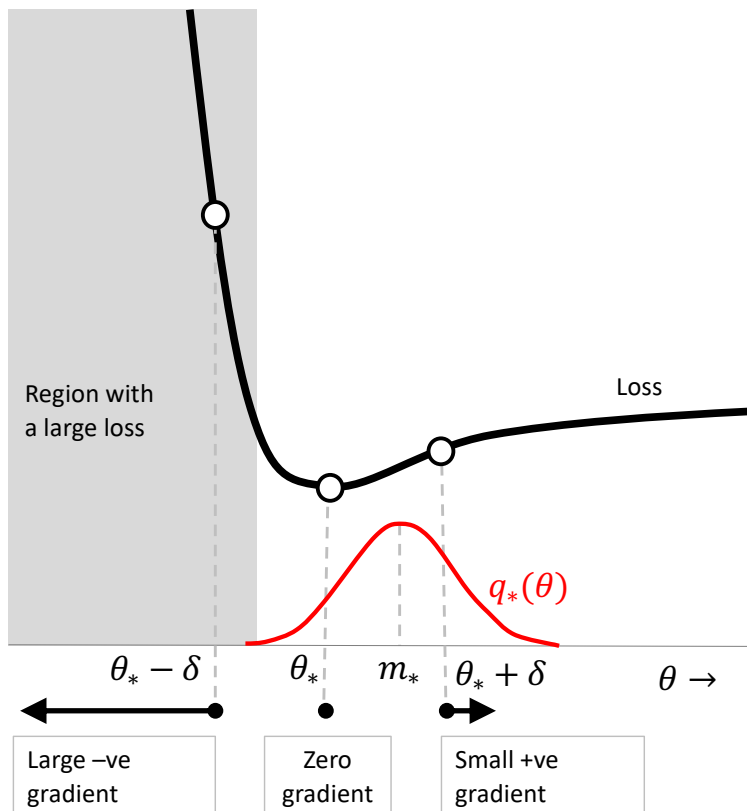
$$\nabla_{\mathbb{E}_q(\theta\theta^{\top})} \mathbb{E}_q[\ell(\theta)] = \mathbb{E}_q[\mathbf{H}_{\theta}]$$

$$Sm \leftarrow (1 - \rho)Sm - \rho \nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)]$$

$$\mathbf{S} \leftarrow (1 - \rho)\mathbf{S} - \rho 2 \nabla_{\mathbb{E}_q(\theta\theta^{\top})} \mathbb{E}_q[\ell(\theta)]$$

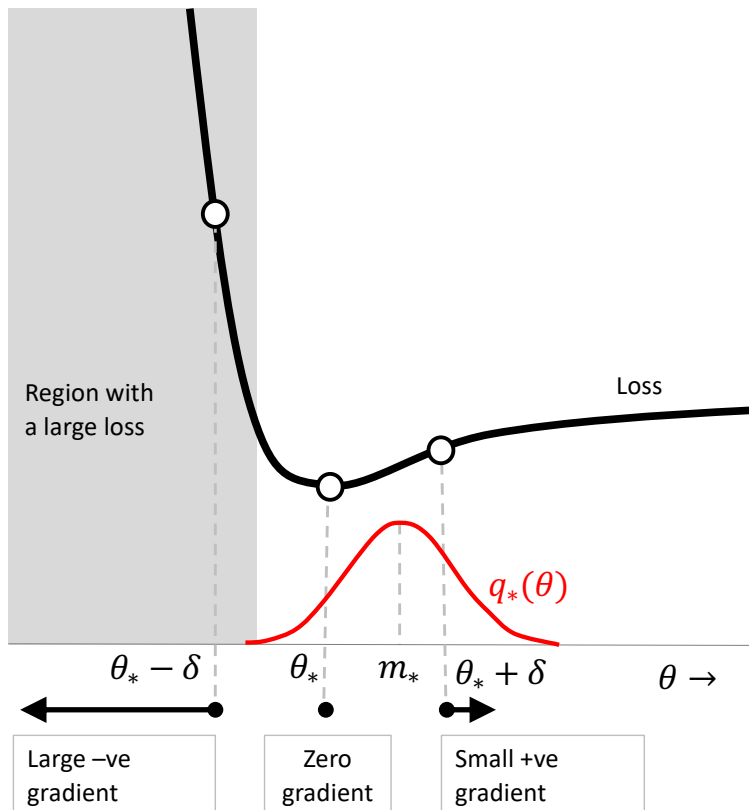
Bayes leads to robust solutions

Avoiding large losses

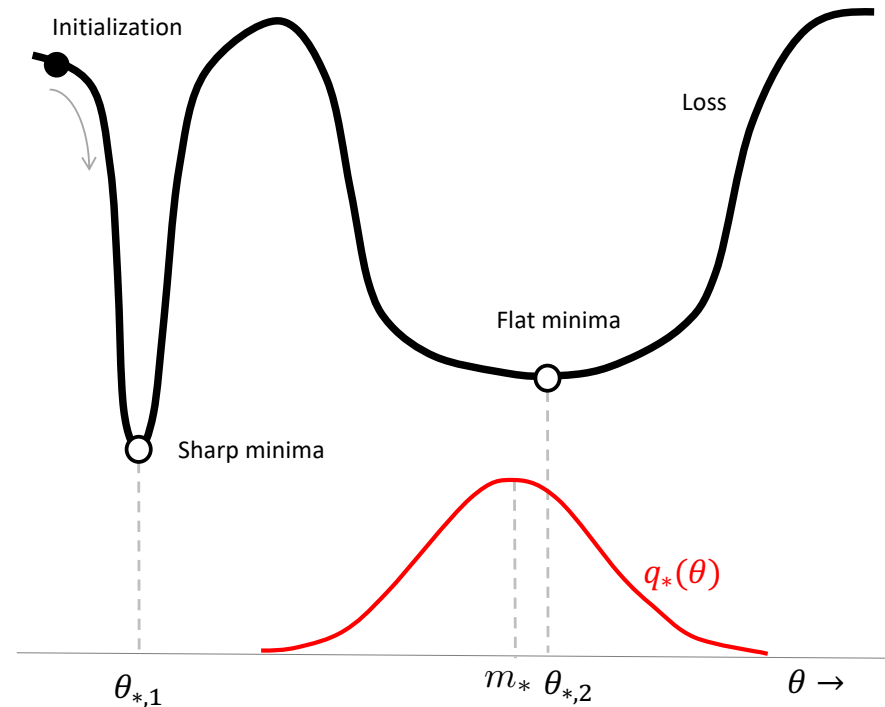


Bayes leads to robust solutions

Avoiding large losses



Avoiding sharp minima



(Some) Bayesian Deep Learning Methods

1. Gal and Ghahramani. "Dropout as a bayesian approximation..." *ICML*. 2016.
2. Maddox, Wesley, et al. "A simple baseline for bayesian uncertainty in deep learning." *arXiv* (2019).
3. Ritter et al. "A scalable laplace approximation for neural networks." (2018).
4. Graves, Alex. "Practical variational inference for neural networks." *NeurIPS* (2011).
5. Blundell, Charles, et al. "Weight uncertainty in neural networks." *ICML* (2015).

(Some) Bayesian Deep Learning Methods

- SGD based (MC-dropout [1], SWAG [2], Laplace [3])
 - Pros: Scales well to large problems
 - Cons: Not flexible

1. Gal and Ghahramani. "Dropout as a bayesian approximation..." *ICML*. 2016.
2. Maddox, Wesley, et al. "A simple baseline for bayesian uncertainty in deep learning." *arXiv* (2019).
3. Ritter et al. "A scalable laplace approximation for neural networks." (2018).
4. Graves, Alex. "Practical variational inference for neural networks." *NeurIPS* (2011).
5. Blundell, Charles, et al. "Weight uncertainty in neural networks." *ICML* (2015).

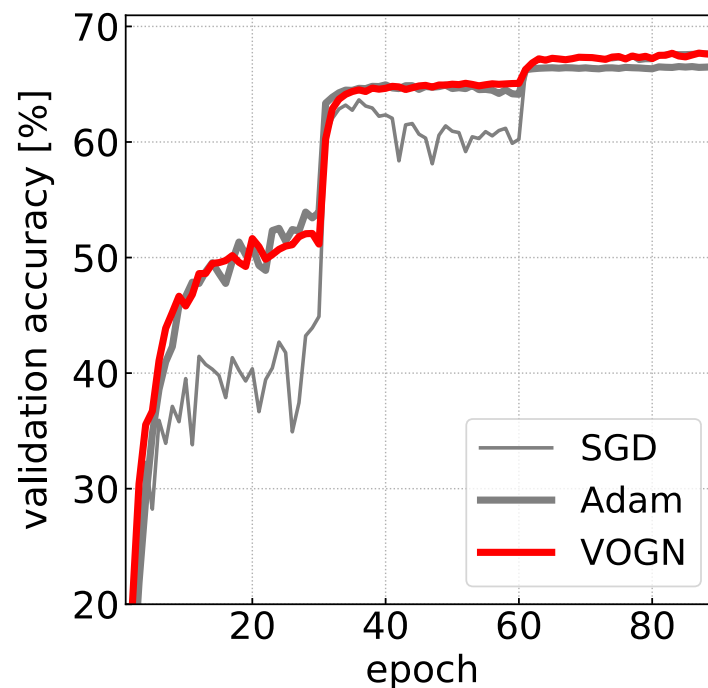
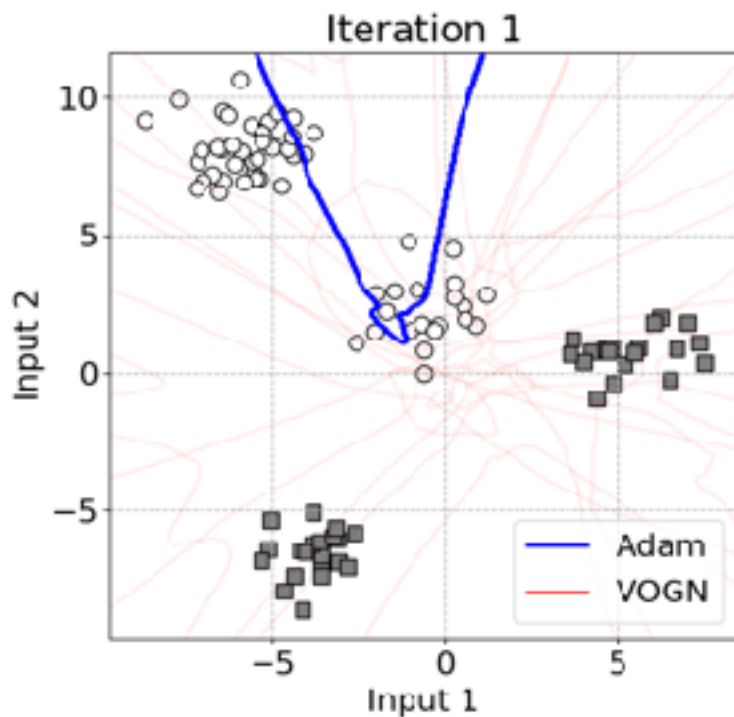
(Some) Bayesian Deep Learning Methods

- SGD based (MC-dropout [1], SWAG [2], Laplace [3])
 - Pros: Scales well to large problems
 - Cons: Not flexible
- Variational inference methods [4,5]
$$\lambda \leftarrow \lambda - \rho \nabla_{\lambda} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$$
 - Pros: Enable flexible distributions
 - Cons: Do not scale to large problems (ImageNet)

1. Gal and Ghahramani. "Dropout as a bayesian approximation..." *ICML*. 2016.
2. Maddox, Wesley, et al. "A simple baseline for bayesian uncertainty in deep learning." *arXiv* (2019).
3. Ritter et al. "A scalable laplace approximation for neural networks." (2018).
4. Graves, Alex. "Practical variational inference for neural networks." *NeurIPS* (2011).
5. Blundell, Charles, et al. "Weight uncertainty in neural networks." *ICML* (2015).

Uncertainty of Deep Nets

VOGN: A modification of Adam but match the performance on ImageNet

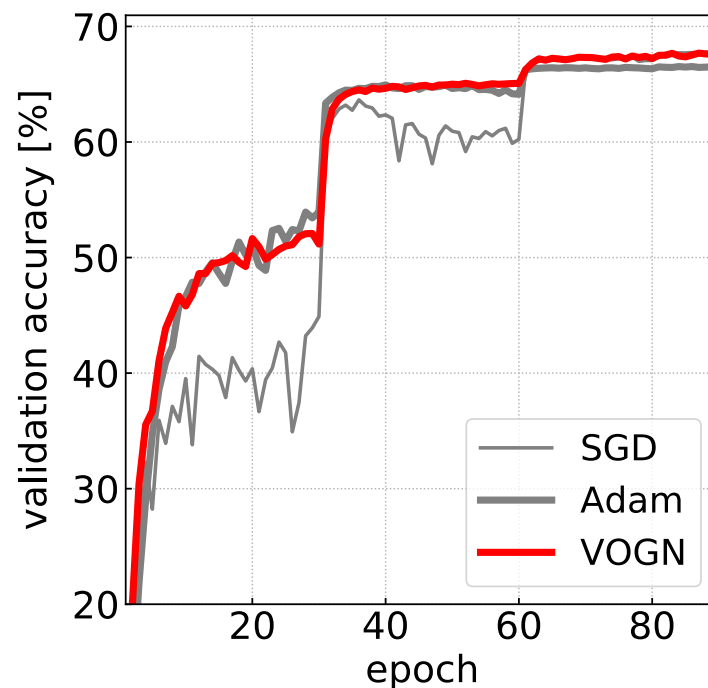
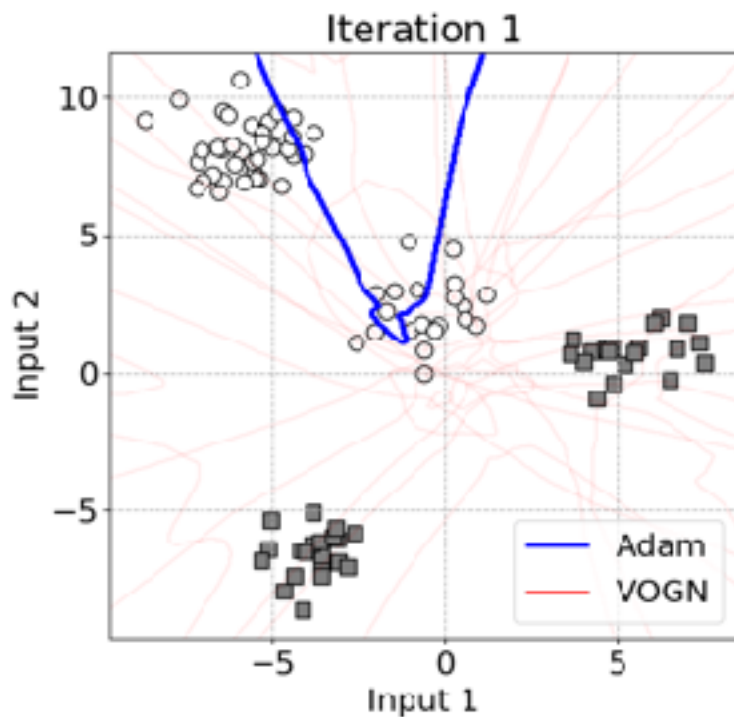


Code available at <https://github.com/team-approx-bayes/dl-with-bayes>

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. "Practical Deep Learning with Bayesian Principles." *NeurIPS* (2019).

Uncertainty of Deep Nets

VOGN: A modification of Adam but match the performance on ImageNet



Code available at <https://github.com/team-approx-bayes/dl-with-bayes>

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. "Practical Deep Learning with Bayesian Principles." *NeurIPS* (2019).

RMSprop/Adam from Bayes

RMSprop

$$\begin{aligned}s &\leftarrow (1 - \rho)s + \rho[\hat{\nabla}\ell(\theta)]^2 \\ \theta &\leftarrow \theta - \alpha(\sqrt{s} + \delta)^{-1}\hat{\nabla}\ell(\theta)\end{aligned}$$

BLR for Gaussian approx

$$\begin{aligned}S &\leftarrow (1 - \rho)S + \rho(\textcolor{red}{H}_{\theta}) \\ m &\leftarrow m - \alpha\textcolor{red}{S}^{-1}\textcolor{red}{\nabla}_{\theta}\ell(\theta)\end{aligned}$$

RMSprop/Adam from Bayes

RMSprop

$$\begin{aligned}s &\leftarrow (1 - \rho)s + \rho[\hat{\nabla}\ell(\theta)]^2 \\ \theta &\leftarrow \theta - \alpha(\sqrt{s} + \delta)^{-1}\hat{\nabla}\ell(\theta)\end{aligned}$$

BLR for Gaussian approx

$$\begin{aligned}S &\leftarrow (1 - \rho)S + \rho(\textcolor{red}{H}_\theta) \\ m &\leftarrow m - \alpha\textcolor{red}{S}^{-1}\nabla_\theta\ell(\theta)\end{aligned}$$

To get RMSprop, make the following choices

- Restrict covariance to be diagonal
- Replace Hessian by square of gradients
- Add square root for scaling vector

RMSprop/Adam from Bayes

RMSprop

$$\begin{aligned}s &\leftarrow (1 - \rho)s + \rho[\hat{\nabla}\ell(\theta)]^2 \\ \theta &\leftarrow \theta - \alpha(\sqrt{s} + \delta)^{-1}\hat{\nabla}\ell(\theta)\end{aligned}$$

BLR for Gaussian approx

$$\begin{aligned}S &\leftarrow (1 - \rho)S + \rho(\textcolor{red}{H}_{\theta}) \\ m &\leftarrow m - \alpha\textcolor{red}{S}^{-1}\textcolor{red}{\nabla}_{\theta}\ell(\theta)\end{aligned}$$

To get RMSprop, make the following choices

- Restrict covariance to be diagonal
- Replace Hessian by square of gradients
- Add square root for scaling vector

For Adam, use a Heavy-ball term with KL divergence as momentum (Appendix E in [1])

Variational Online Gauss-Newton

RMSprop

$$g \leftarrow \hat{\nabla} \ell(\theta)$$

$$s \leftarrow (1 - \rho)s + \rho g^2$$

$$\theta \leftarrow \theta - \alpha(\sqrt{s} + \delta)^{-1}g$$

VOGN

$$g \leftarrow \hat{\nabla} \ell(\theta), \text{ where } \theta \sim \mathcal{N}(m, \sigma^2)$$

$$s \leftarrow (1 - \rho)s + \rho(\sum_i g_i^2)$$

$$m \leftarrow m - \alpha(s + \gamma)^{-1} \nabla_{\theta} \ell(\theta)$$

$$\sigma^2 \leftarrow (s + \gamma)^{-1}$$

```
import torch
+import torchsso

train_loader = torch.utils.data.DataLoader(train_dataset)
model = MLP()

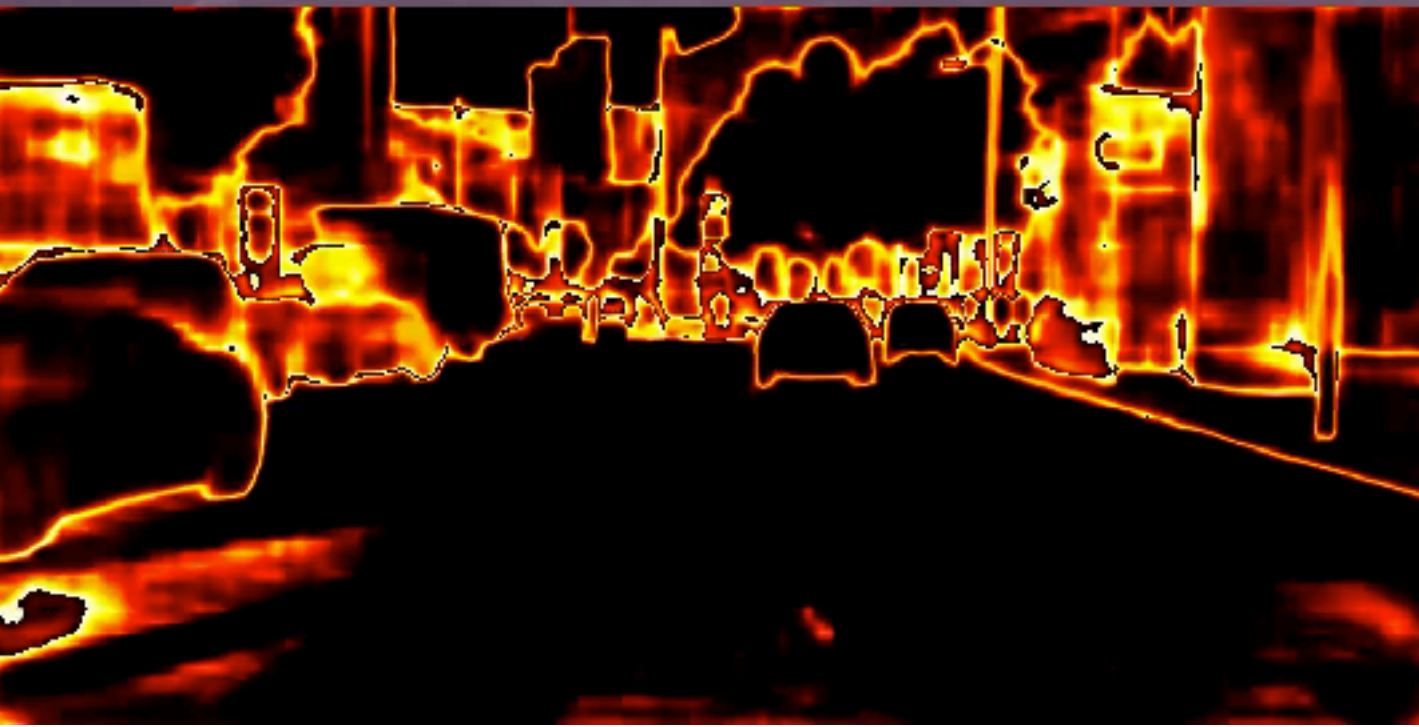
-optimizer = torch.optim.Adam(model.parameters())
+optimizer = torchsso.optim.VOGN(model, dataset_size=len(train_loader.dataset))
```

Available at <https://github.com/team-approx-bayes/dl-with-bayes>

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. "Practical Deep Learning with Bayesian Principles." *NeurIPS* (2019).



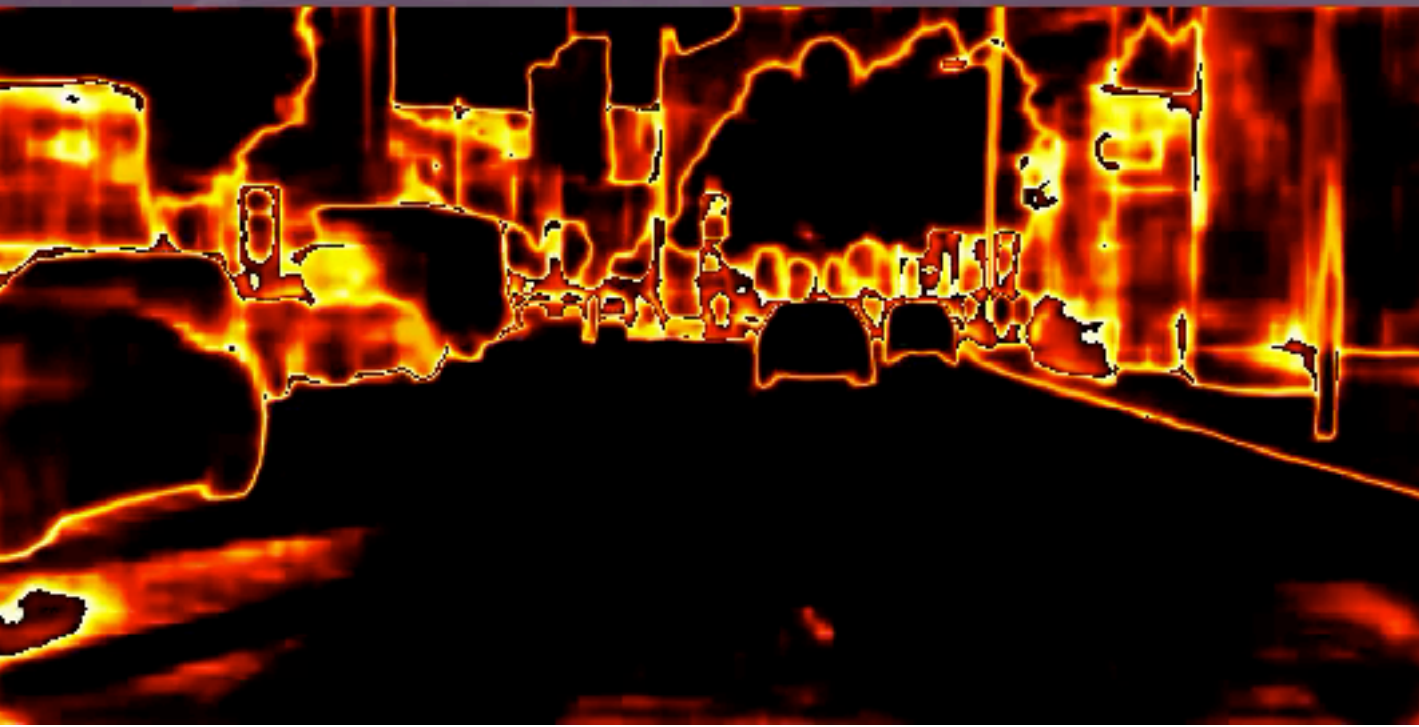
Image
Segmentation



Uncertainty
(entropy of
class probs)



Image
Segmentation

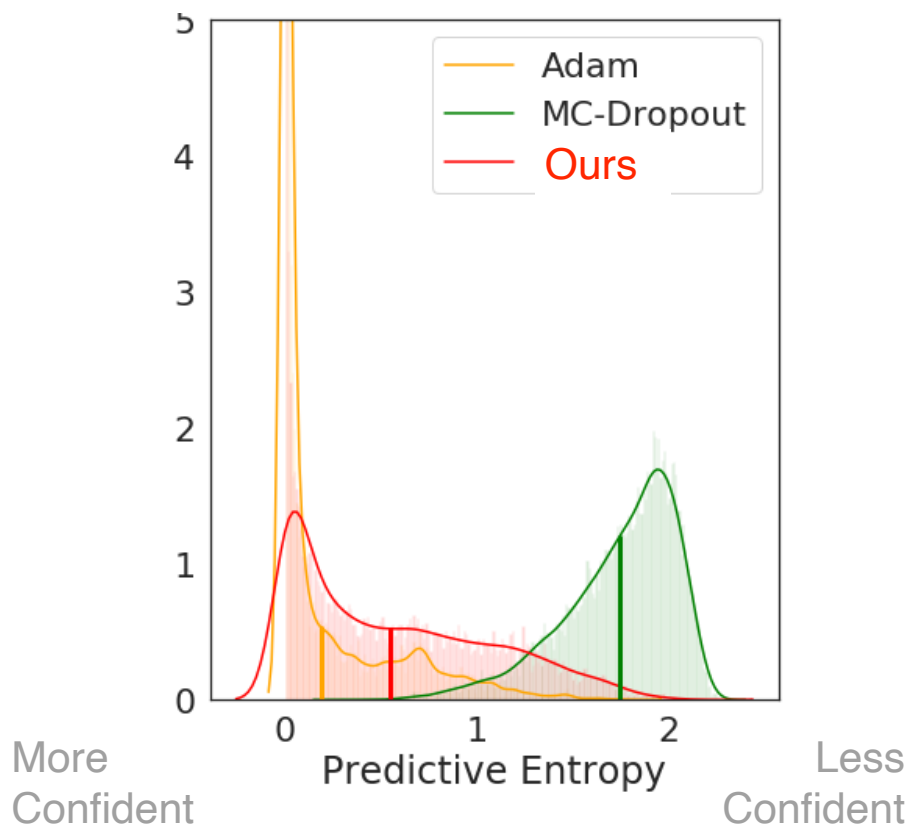


Uncertainty
(entropy of
class probs)

Out-of-Distributions Test

Our method (in red) is confident on “in-distribution” data, and not overconfident on “out-of-distribution” data.

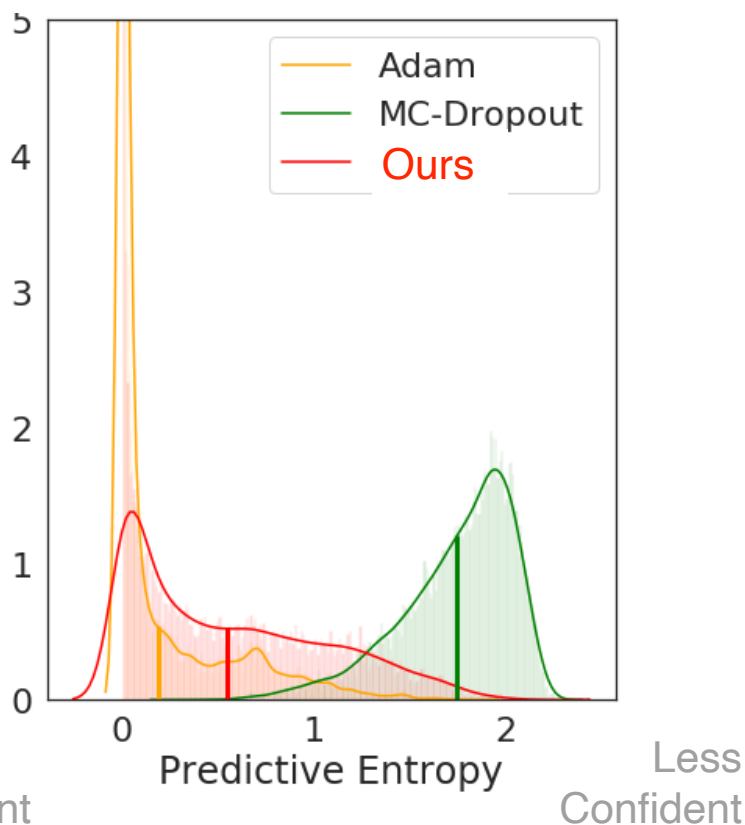
In-Distribution (CIFAR-10)



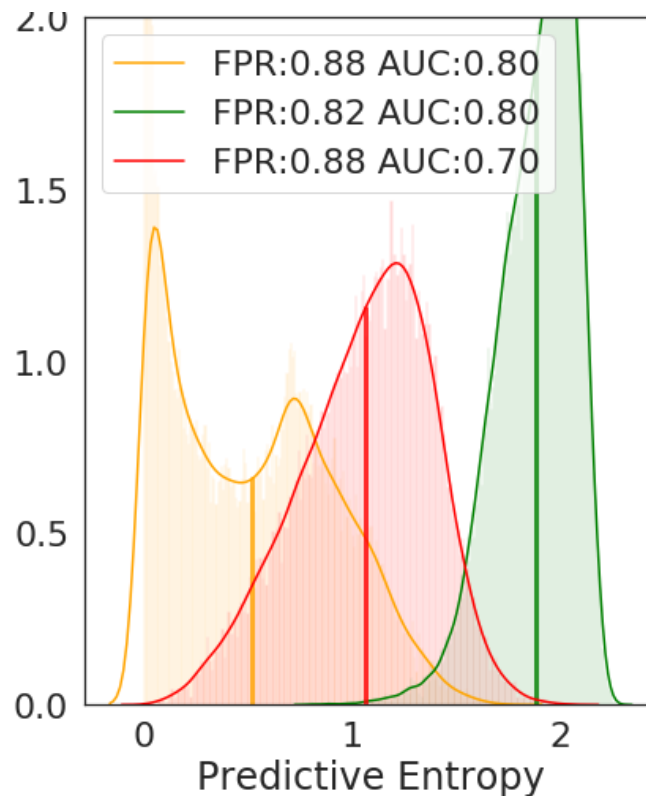
Out-of-Distributions Test

Our method (in red) is confident on “in-distribution” data, and not overconfident on “out-of-distribution” data.

In-Distribution (CIFAR-10)



Out-of-Distribution (SVHN)

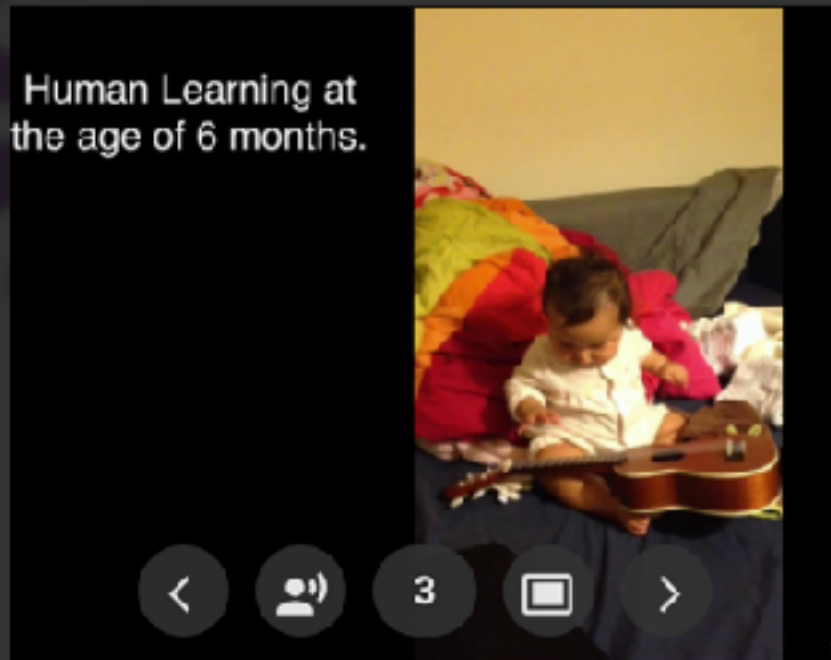
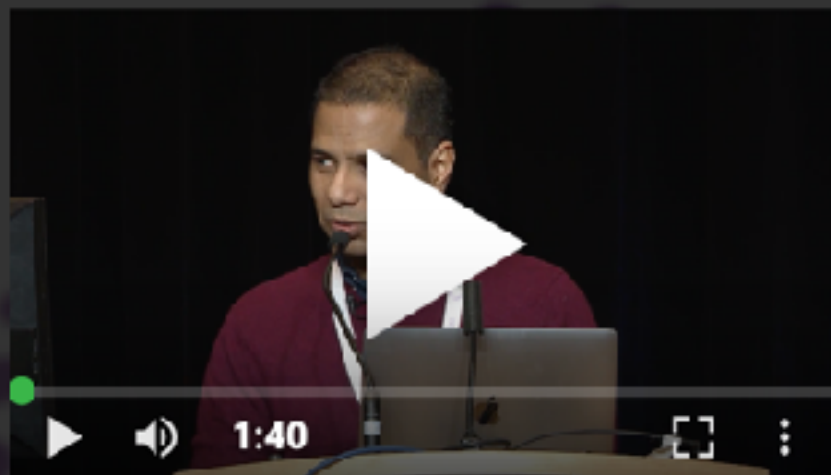


Tuning VOGN

The trick is to mimic Adam's trajectory as closely as possible

Tuning VOGN: Currently, there is no common recipe for tuning the algorithmic hyperparameters for VI, especially for large-scale tasks like ImageNet classification. One key idea we use in our experiments is to start with Adam hyperparameters and then make sure that VOGN training closely follows an Adam-like trajectory in the beginning of training. To achieve this, we divide the tuning into an *optimisation part* and a *regularisation part*. In the *optimisation part*, we first tune the hyperparameters of a deterministic version of VOGN, called the online Gauss-Newton (OGN) method. This method, described in Appendix C, is more stable than VOGN since it does not require MC sampling, and can be used as a stepping stone when moving from Adam/SGD to VOGN. After reaching a competitive performance to Adam/SGD by OGN, we move to the *regularisation part*, where we tune the prior precision δ , the tempering parameter τ , and the number of MC samples K for VOGN. We initialise our search by setting the prior precision δ using the L2-regularisation parameter used for OGN, as well as the dataset size N . Another technique is to warm-up the parameter τ towards $\tau = 1$ (also see the “momentum and initialisation” part). Setting τ to smaller values usually stabilises the training, and increasing it slowly also helps during tuning. We also add an *external damping factor* $\gamma > 0$ to the moving average s_t . This increases the lower bound of the eigenvalues of the diagonal covariance Σ_t and prevents the noise and the step size from becoming too large. We find that a mix of these techniques works well for the problems we considered.

NeurIPS 2019 Tutorial



Deep Learning with Bayesian Principles

by **Mohammad Emtiyaz Khan** · Dec 9, 2019 ·

#NeurIPS 2019

Follow

Views 151 807

Presentations 263

Followers 200

Latest

Popular

...



From System 1 Deep Learning to System 2 Deep Learning

by [Yoshua Bengio](#)

17,953 views · Dec 11, 2019



NeurIPS Workshop on Machine Learning for Creativity and Design...

by [Aaron Hertzmann](#) [Adam Roberts](#) ...

9,654 views · Dec 14, 2019



Deep Learning with Bayesian Principles

by [Mohammad Emtiyaz Khan](#)

4,084 views · Dec 5, 2019



Efficient Processing of Deep Neural Network: from Algorithms to...

by [Vivienne Sze](#)

7,162 views · Dec 9, 2019

Past and New Work

- Natural Gradient Variational Inference

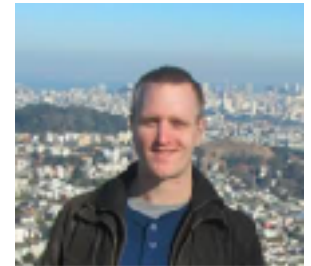
1. Khan and Lin. "Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models." *Alstats* (2017).
2. Khan and Nielsen. "Fast yet simple natural-gradient descent for variational inference in complex models." (2018) *ISITA*.



Wu Lin (UBC)

- Mixture of Exponential family

3. Lin et al. "Fast and Simple Natural-Gradient Variational Inference with Mixture of Exponential-family Approximations," *ICML* (2019).



Mark Schmidt (UBC)

- Generalization of natural gradients

4. Lin et al. "Handling the Positive-Definite Constraint in the Bayesian Learning Rule", *ICML* (2020)
5. Lin et al. "Tractable structured natural gradient descent using local parameterizations", *ICML*, (2021)

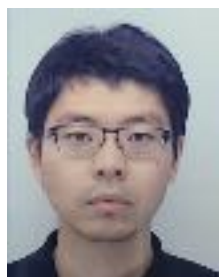


Frank Nielsen (Sony)

- Gaussian approx \Leftrightarrow Newton-variants

Gaussian Approximation and DL

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Mishkin et al. "SLANG: Fast Structured Covariance Approximations for Bayesian Deep Learning with Natural Gradient" *NeurIPS* (2018).
3. Osawa et al. "Practical Deep Learning with Bayesian Principles." *NeurIPS* (2019).



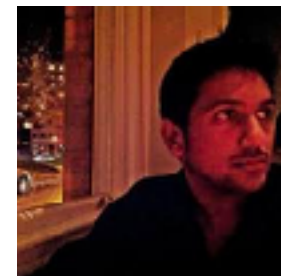
Voot Tangkaratt
(Postdoc, RIKEN-AIP)



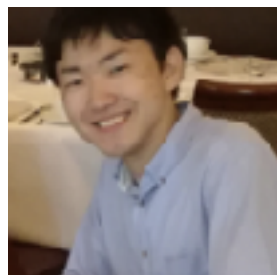
Aaron Mishkin
(Intern From UBC)
Frederik Kunstner
(Intern From EPFL)
Didrik Nielsen
(Past: RA)



Yarin Gal
(UOxford)



Akash Srivastava
(UEdinburgh)



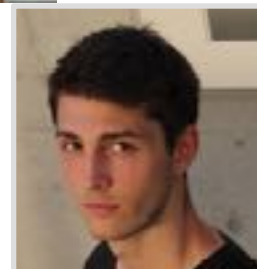
Kazuki Osawa
(Tokyo Tech)



Rio Yokota
(Tokyo Tech)



Anirudh Jain
(Intern from
IIT-ISM, India)



Runa Eschenhagen
(Intern from
U Osnabruck)



Siddharth Swaroop
(UCambridge)



Rich Turner
(UCambridge)

Extensions

- Binary Neural Networks (Bernoulli approx)

1. Meng, et al. "Training Binary Neural Networks using the Bayesian Learning Rule." *ICML* (2020).

- Gaussian Process

2. Chang et al. "Fast Variational Learning in State-Space GP Models", *MLSP* (2020)

– For sparse GPs, BLR is a generalization of [1]



Roman
Bachmann
(Intern from EPFL)



Xiangming
Meng
(RIKEN-AIP)



Paul Chang
(Aalto University)



W. J. Wilkinson
(Aalto University)



Arno Solin
(Aalto University)

How to design AI that learn like us?

How to design AI that learn like us?

- Uncertainty -> Learning -> Knowledge

How to design AI that learn like us?

- Uncertainty -> Learning -> Knowledge
- Three questions

How to design AI that learn like us?

- Uncertainty -> Learning -> Knowledge
- Three questions
 - Q1: What do we know? (model)

How to design AI that learn like us?

- Uncertainty -> Learning -> Knowledge
- Three questions
 - Q1: What do we know? (model)
 - Q2: What do we not know? (uncertainty)

How to design AI that learn like us?

- Uncertainty -> Learning -> Knowledge
- Three questions
 - Q1: What do we know? (model)
 - Q2: What do we not know? (uncertainty)
 - Q3: What do we need to know? (action & exploration)

How to design AI that learn like us?

- Uncertainty -> Learning -> Knowledge
- Three questions
 - Q1: What do we know? (model)
 - Q2: What do we not know? (uncertainty)
 - Q3: What do we need to know? (action & exploration)
- Posterior approximation is the key

How to design AI that learn like us?

- Uncertainty -> Learning -> Knowledge
- Three questions
 - Q1: What do we know? (model)
 - Q2: What do we not know? (uncertainty)
 - Q3: **What do we need to know? (action & exploration)**
- Posterior approximation is the key
 - (Q1) Models == representation of the world

How to design AI that learn like us?

- Uncertainty -> Learning -> Knowledge
- Three questions
 - Q1: What do we know? (model)
 - Q2: What do we not know? (uncertainty)
 - Q3: **What do we need to know? (action & exploration)**
- Posterior approximation is the key
 - (Q1) Models == representation of the world
 - (Q2) Posterior approximations == representation of the model

How to design AI that learn like us?

- Uncertainty -> Learning -> Knowledge
- Three questions
 - Q1: What do we know? (model)
 - Q2: What do we not know? (uncertainty)
 - Q3: What do we need to know? (action & exploration)
- Posterior approximation is the key
 - (Q1) Models == representation of the world
 - (Q2) Posterior approximations == representation of the model
 - (Q3) Use posterior approximations for knowledge representation, transfer, and collection.

Approximate Bayesian Inference Team



Emtiyaz Khan

Team Leader



Pierre Alquier

Research Scientist



Gian Maria Marconi

Postdoc



Thomas Möllenhoff

Postdoc

<https://team-approx-bayes.github.io/>



Wu Lin

PhD Student
University of British Columbia



Dharmesh Tailor

Research Assistant



Fariz Ikhwantri

Part-time Student
Tokyo Institute of Technology



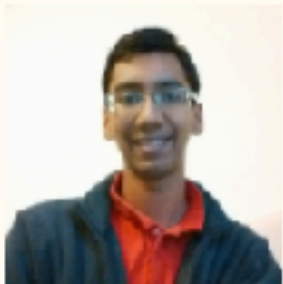
Happy Buzaaba

Part-time Student
University of Tsukuba



Evgenii Egorov

Remote Collaborator
Skoltech



Siddharth Swaroop

Remote Collaborator
University of Cambridge



Dimitri Meunier

Remote Collaborator
FNSAF Paris



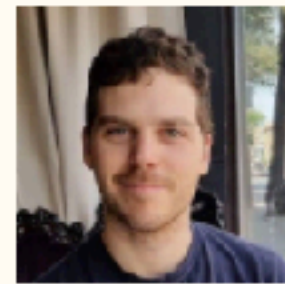
Peter Nickl

Remote Collaborator
TU Darmstadt



Erik Daxberger

Remote Collaborator
University of Cambridge



Alexandre Piché

Remote Collaborator
MILA

References for Bayes as Optimization

$$\arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

References for Bayes as Optimization

$$\arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)} [\ell(\theta)] - \mathcal{H}(q)$$

- Bayesian statistics

1. Jaynes, Edwin T. "Information theory and statistical mechanics." *Physical review* (1957)
2. Zellner, A. "Optimal information processing and Bayes's theorem." *The American Statistician* (1988)
3. Bissiri, Pier Giovanni, Chris C. Holmes, and Stephen G. Walker. "A general framework for updating belief distributions." *RSS: Series B (Statistical Methodology)* (2016)

References for Bayes as Optimization

$$\arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)} [\ell(\theta)] - \mathcal{H}(q)$$

- Bayesian statistics

1. Jaynes, Edwin T. "Information theory and statistical mechanics." *Physical review* (1957)
2. Zellner, A. "Optimal information processing and Bayes's theorem." *The American Statistician* (1988)
3. Bissiri, Pier Giovanni, Chris C. Holmes, and Stephen G. Walker. "A general framework for updating belief distributions." *RSS: Series B (Statistical Methodology)* (2016)

- PAC-Bayes

4. Shawe-Taylor, John, and Robert C. Williamson. "A PAC analysis of a Bayesian estimator." COLT 1997.
5. Alquier, Pierre. "PAC-Bayesian bounds for randomized empirical risk minimizers." *Mathematical Methods of Statistics* 17.4 (2008): 279-304.

References for Bayes as Optimization

$$\arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)} [\ell(\theta)] - \mathcal{H}(q)$$

- Bayesian statistics

1. Jaynes, Edwin T. "Information theory and statistical mechanics." *Physical review* (1957)
2. Zellner, A. "Optimal information processing and Bayes's theorem." *The American Statistician* (1988)
3. Bissiri, Pier Giovanni, Chris C. Holmes, and Stephen G. Walker. "A general framework for updating belief distributions." *RSS: Series B (Statistical Methodology)* (2016)

- PAC-Bayes

4. Shawe-Taylor, John, and Robert C. Williamson. "A PAC analysis of a Bayesian estimator." COLT 1997.
5. Alquier, Pierre. "PAC-Bayesian bounds for randomized empirical risk minimizers." *Mathematical Methods of Statistics* 17.4 (2008): 279-304.

- Online-learning (Exponential Weight Aggregate)

6. Cesa-Bianchi, Nicolo, and Gabor Lugosi. *Prediction, learning, and games*. 2006.

References for Bayes as Optimization

$$\arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)} [\ell(\theta)] - \mathcal{H}(q)$$

- Bayesian statistics

1. Jaynes, Edwin T. "Information theory and statistical mechanics." *Physical review* (1957)
2. Zellner, A. "Optimal information processing and Bayes's theorem." *The American Statistician* (1988)
3. Bissiri, Pier Giovanni, Chris C. Holmes, and Stephen G. Walker. "A general framework for updating belief distributions." *RSS: Series B (Statistical Methodology)* (2016)

- PAC-Bayes

4. Shawe-Taylor, John, and Robert C. Williamson. "A PAC analysis of a Bayesian estimator." COLT 1997.
5. Alquier, Pierre. "PAC-Bayesian bounds for randomized empirical risk minimizers." *Mathematical Methods of Statistics* 17.4 (2008): 279-304.

- Online-learning (Exponential Weight Aggregate)

6. Cesa-Bianchi, Nicolo, and Gabor Lugosi. *Prediction, learning, and games*. 2006.

- Free-energy principle

7. Friston, K. "The free-energy principle: a unified brain theory?." *Nature neuroscience* (2010)

Bayes with Approximate Posterior

$$\arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)} [\ell(\theta)] - \mathcal{H}(q)$$

All distribution Distribution Entropy

Restrict the set of distribution from P to Q

$$\arg \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)} [\ell(\theta)] - \mathcal{H}(q)$$

Bayes with Approximate Posterior

$$\arg \min_{q \in \mathcal{P}} \mathbb{E}_{q(\theta)} [\ell(\theta)] - \mathcal{H}(q)$$

All distribution Distribution Entropy

Restrict the set of distribution from P to Q

$$\arg \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)} [\ell(\theta)] - \mathcal{H}(q)$$

This is known as **Variational Inference**, but along with the Bayesian learning rule, it enables us to derive many more algorithms (including Bayes' rule). So this is not just a method, but a principle.

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta)$$

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta) = -\lambda_{\mathcal{D}}^{\top}T(\theta)$$

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta) = -\lambda_{\mathcal{D}}^{\top} T(\theta) \leftarrow \text{Sufficient statistics of } q$$

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta) = -\lambda_{\mathcal{D}}^{\top} T(\theta) \leftarrow \text{Sufficient statistics of } q$$

$$\ell(\theta) := (y - X\theta)^{\top} (y - X\theta) + \gamma\theta^{\top} \theta$$

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta) = -\lambda_{\mathcal{D}}^{\top} T(\theta) \leftarrow \text{Sufficient statistics of } q$$

$$\begin{aligned}\ell(\theta) &:= (y - X\theta)^{\top} (y - X\theta) + \gamma\theta^{\top} \theta \\ &= -2\theta^{\top} (X^{\top} y) + \text{Tr} [\theta\theta^{\top} (X^{\top} X + \gamma I)] + \text{cnst}\end{aligned}$$

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta) = -\lambda_{\mathcal{D}}^{\top} T(\theta) \leftarrow \text{Sufficient statistics of } q$$

$$\begin{aligned}\ell(\theta) &:= (y - X\theta)^{\top} (y - X\theta) + \gamma\theta^{\top} \theta \\ &= -2\theta^{\top} (X^{\top} y) + \text{Tr} [\theta\theta^{\top} (X^{\top} X + \gamma I)] + \text{cnst}\end{aligned}$$

$$\implies \mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}\mu$$

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta) = -\lambda_{\mathcal{D}}^{\top} T(\theta) \leftarrow \text{Sufficient statistics of } q$$

$$\begin{aligned}\ell(\theta) &:= (y - X\theta)^{\top} (y - X\theta) + \gamma\theta^{\top} \theta \\ &= -2\theta^{\top} (X^{\top} y) + \text{Tr} [\theta\theta^{\top} (X^{\top} X + \gamma I)] + \text{cnst}\end{aligned}$$

$$\implies \mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}^{\top} \mu \implies \nabla_{\mu} \mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}$$

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta) = -\lambda_{\mathcal{D}}^{\top} T(\theta) \leftarrow \text{Sufficient statistics of } q$$

$$\begin{aligned}\ell(\theta) &:= (y - X\theta)^{\top} (y - X\theta) + \gamma\theta^{\top}\theta \\ &= -2\theta^{\top} (X^{\top}y) + \text{Tr} [\theta\theta^{\top} (X^{\top}X + \gamma I)] + \text{cnst}\end{aligned}$$

$$\implies \mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}\mu \implies \nabla_{\mu}\mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}$$

$$\lambda \leftarrow \lambda - \rho \nabla_{\mu} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$$

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta) = -\lambda_{\mathcal{D}}^{\top} T(\theta) \leftarrow \text{Sufficient statistics of } q$$

$$\begin{aligned}\ell(\theta) &:= (y - X\theta)^{\top} (y - X\theta) + \gamma\theta^{\top} \theta \\ &= -2\theta^{\top} (X^{\top} y) + \text{Tr} [\theta\theta^{\top} (X^{\top} X + \gamma I)] + \text{cnst}\end{aligned}$$

$$\implies \mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}\mu \implies \nabla_{\mu}\mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}$$

$$\lambda \leftarrow \lambda - \rho (\nabla_{\mu}\mathbb{E}_q[\ell(\theta)] + \lambda)$$

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta) = -\lambda_{\mathcal{D}}^{\top} T(\theta) \leftarrow \text{Sufficient statistics of } q$$

$$\begin{aligned}\ell(\theta) &:= (y - X\theta)^{\top} (y - X\theta) + \gamma\theta^{\top} \theta \\ &= -2\theta^{\top} (X^{\top} y) + \text{Tr} [\theta\theta^{\top} (X^{\top} X + \gamma I)] + \text{cnst}\end{aligned}$$

$$\implies \mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}\mu \implies \nabla_{\mu} \mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}$$

$$\lambda \leftarrow \lambda - \rho(-\lambda_{\mathcal{D}} + \lambda)$$

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta) = -\lambda_{\mathcal{D}}^{\top} T(\theta) \leftarrow \text{Sufficient statistics of } q$$

$$\begin{aligned}\ell(\theta) &:= (y - X\theta)^{\top} (y - X\theta) + \gamma\theta^{\top} \theta \\ &= -2\theta^{\top} (X^{\top} y) + \text{Tr} [\theta\theta^{\top} (X^{\top} X + \gamma I)] + \text{cnst}\end{aligned}$$

$$\implies \mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}\mu \implies \nabla_{\mu} \mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}$$

$$\lambda \leftarrow \lambda - \rho(-\lambda_{\mathcal{D}} + \lambda) \implies \lambda_* = \lambda_{\mathcal{D}}$$

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta) = -\lambda_{\mathcal{D}}^{\top} T(\theta) \leftarrow \text{Sufficient statistics of } q$$

$$\begin{aligned}\ell(\theta) &:= (y - X\theta)^{\top} (y - X\theta) + \gamma\theta^{\top}\theta \\ &= -2\theta^{\top} (X^{\top}y) + \text{Tr} [\theta\theta^{\top} (X^{\top}X + \gamma I)] + \text{cnst}\end{aligned}$$

$$\implies \mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}\mu \implies \nabla_{\mu}\mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}$$

$$\lambda \leftarrow \lambda - \rho(-\lambda_{\mathcal{D}} + \lambda) \implies \lambda_* = \lambda_{\mathcal{D}}$$

$$S_* = X^{\top}X + \gamma I$$

Conjugate Bayesian Inference from Bayesian Principles

Ex: Linear model, Kalman filters, HMM, etc.

$$\ell(\theta) := -\log p(\mathcal{D}|\theta)p(\theta) = -\lambda_{\mathcal{D}}^{\top} T(\theta) \leftarrow \text{Sufficient statistics of } q$$

$$\begin{aligned}\ell(\theta) &:= (y - X\theta)^{\top} (y - X\theta) + \gamma\theta^{\top}\theta \\ &= -2\theta^{\top} (X^{\top}y) + \text{Tr} [\theta\theta^{\top} (X^{\top}X + \gamma I)] + \text{cnst}\end{aligned}$$

$$\implies \mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}\mu \implies \nabla_{\mu}\mathbb{E}_q[\ell(\theta)] = -\lambda_{\mathcal{D}}$$

$$\lambda \leftarrow \lambda - \rho(-\lambda_{\mathcal{D}} + \lambda) \implies \lambda_* = \lambda_{\mathcal{D}}$$

$$S_* = X^{\top}X + \gamma I \quad m_* = (X^{\top}X + \gamma I)^{-1}X^{\top}y$$

Conjugate Bayesian Inference from Bayesian Principles

The following algorithms can be obtained by setting $\lambda_* = \lambda_{\mathcal{D}}$

- Forward-backward algorithm [2]
 - Kalman filters, HMM etc.
- Stochastic Variational Inference [3]
- Variational message passing [4]

1. Khan and Lin. "Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models." *Alstats* (2017).
2. Binder et al.. Space-Efficient Inference in Dynamic Probabilistic Networks. *IJCAI* (1997).
3. Hoffman et al. Stochastic variational inference. *JMLR* (2013)
4. Winn and Bishop. "Variational message passing." *JMLR* (2005)

Laplace Approximation

Derived by choosing a multivariate Gaussian, then running the following Newton's update

$$m \leftarrow m - \rho \mathbf{S}^{-1} \nabla_m \ell(m)$$

$$\mathbf{S} \leftarrow (1 - \rho) \mathbf{S} + \rho \mathbf{H}_m \leftarrow \text{Hessian at } m$$

Laplace Approximation

Derived by choosing a multivariate Gaussian, then running the following Newton's update

$$\begin{aligned} m &\leftarrow m - \rho \mathbf{S}^{-1} \nabla_m \ell(m) \\ \mathbf{S} &\leftarrow (1 - \rho) \mathbf{S} + \rho \mathbf{H}_m \end{aligned}$$

← Hessian at m

Bayesian principles we discussed are general principles to derive learning algorithms

Laplace Approximation

Derived by choosing a multivariate Gaussian, then running the following Newton's update

$$\begin{aligned} m &\leftarrow m - \rho \mathbf{S}^{-1} \nabla_m \ell(m) \\ \mathbf{S} &\leftarrow (1 - \rho) \mathbf{S} + \rho \mathbf{H}_m \end{aligned}$$

← Hessian at m

Bayesian principles we discussed are general principles to derive learning algorithms

Calling them variational inference limits their scope!

References for Posterior Approximations

$$\arg \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

References for Posterior Approximations

$$\arg \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)} [\ell(\theta)] - \mathcal{H}(q)$$

- Variational inference

1. Hinton, Geoffrey, and Drew Van Camp. "Keeping neural networks simple by minimizing the description length of the weights." *COLT* 1993.
2. Jordan, Michael I., et al. "An introduction to variational methods for graphical models." *Machine learning* 37.2 (1999): 183-233.

References for Posterior Approximations

$$\arg \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)} [\ell(\theta)] - \mathcal{H}(q)$$

- Variational inference

1. Hinton, Geoffrey, and Drew Van Camp. "Keeping neural networks simple by minimizing the description length of the weights." *COLT* 1993.
2. Jordan, Michael I., et al. "An introduction to variational methods for graphical models." *Machine learning* 37.2 (1999): 183-233.

- Entropy-regularized / Maximum-entropy RL

3. Williams, Ronald J., and Jing Peng. "Function optimization using connectionist reinforcement learning algorithms." *Connection Science* 3.3 (1991): 241-268.
4. Ziebart, Brian D. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. Diss. figshare, 2010. (see chapter 5)

References for Posterior Approximations

$$\arg \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)} [\ell(\theta)] - \mathcal{H}(q)$$

- Variational inference

1. Hinton, Geoffrey, and Drew Van Camp. "Keeping neural networks simple by minimizing the description length of the weights." *COLT* 1993.
2. Jordan, Michael I., et al. "An introduction to variational methods for graphical models." *Machine learning* 37.2 (1999): 183-233.

- Entropy-regularized / Maximum-entropy RL

3. Williams, Ronald J., and Jing Peng. "Function optimization using connectionist reinforcement learning algorithms." *Connection Science* 3.3 (1991): 241-268.
4. Ziebart, Brian D. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. Diss. figshare, 2010. (see chapter 5)

- Parameter-Space Exploration in RL

5. Rückstieß, Thomas, et al. "Exploring parameter space in reinforcement learning." *Paladyn, Journal of Behavioral Robotics* 1.1 (2010): 14-24.
6. Plappert, Matthias, et al. "Parameter space noise for exploration." *arXiv preprint arXiv:1706.01905* (2017)
7. Fortunato, Meire, et al. "Noisy networks for exploration." *arXiv preprint arXiv:1706.10295* (2017).

More References for Posterior Approximations

- Evolution strategy $\arg \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)} [\ell(\theta)]$
 1. Ingo Rechenberg, *Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution* (PhD thesis) 1971.
- Gaussian Homotopy
 2. Mobahi, Hossein, and John W. Fisher III. "A theoretical analysis of optimization by Gaussian continuation." *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.
- Smoothing-based Optimization
 3. Leordeanu, Marius, and Martial Hebert. "Smoothing-based optimization." *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008.
- Graduated Optimization
 4. Hazan, Elad, Kfir Yehuda Levy, and Shai Shalev-Shwartz. "On graduated optimization for stochastic non-convex problems." *International conference on machine learning*. 2016.
- Stochastic Search
 5. Zhou, Enlu, and Jiaqiao Hu. "Gradient-based adaptive stochastic search for non-differentiable optimization." *IEEE Transactions on Automatic Control* 59.7 (2014): 1818-1832.

Black-Box VI & Bayesian Learning rule

Bayes learning rule: $\lambda \leftarrow \lambda - \rho \nabla_{\mu} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$

Black-Box VI [1]: $\lambda \leftarrow \lambda - \rho \nabla_{\lambda} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$

Black-box VI is more generally applicable (beyond exponential-family), but we cannot derive learning-algorithms from it (even for conjugate Bayesian models)

Bayesian Learning Rule and Related Works

$$\min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

Bayes learning rule: $\lambda \leftarrow \lambda - \rho \nabla_{\mu} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$

Natural-Gradient VI: $\lambda \leftarrow \lambda - \rho F_q^{-1} \nabla_{\lambda} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$

 Fisher Information Matrix

1. Khan and Lin. "Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models." *Alstats* (2017).
2. Raskutti, Garvesh, and Sayan Mukherjee. "The information geometry of mirror descent." *IEEE Transactions on Information Theory* 61.3 (2015): 1451-1457.

Bayesian Learning Rule and Related Works

$$\min_{q \in \mathcal{Q}} \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

Bayes learning rule: $\lambda \leftarrow \lambda - \rho \nabla_{\mu} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$

Natural-Gradient VI: $\lambda \leftarrow \lambda - \rho F_q^{-1} \nabla_{\lambda} (\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q))$

 Fisher Information Matrix

Also equivalent to a mirror-descent algorithm. The Geometry of the mirror-descent is defined by the log partition function of the posterior approximation.

1. Khan and Lin. "Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models." *Alstats* (2017).
2. Raskutti, Garvesh, and Sayan Mukherjee. "The information geometry of mirror descent." *IEEE Transactions on Information Theory* 61.3 (2015): 1451-1457.

References for Step C: Natural-Gradient VI

1. Sato, Masa-aki. "Fast learning of on-line EM algorithm." Technical Report, ATR Human Information Processing Research Laboratories (1999).
2. Sato, Masa-Aki. "Online model selection based on the variational Bayes." *Neural computation* 13.7 (2001): 1649-1681.
3. Winn, John, and Christopher M. Bishop. "Variational message passing." *Journal of Machine Learning Research* 6.Apr (2005): 661-694.
4. Honkela, Antti, et al. "Approximate Riemannian conjugate gradient learning for fixed-form variational Bayes." *Journal of Machine Learning Research* 11.Nov (2010): 3235-3268.
5. Knowles, David A., and Tom Minka. "Non-conjugate variational message passing for multinomial and binary regression." *NeurIPS*. (2011).
6. Hoffman, Matthew D., et al. "Stochastic variational inference." *JMLR* (2013).
7. Salimans, Tim, and David A. Knowles. "Fixed-form variational posterior approximation through stochastic linear regression." *Bayesian Analysis* 8.4 (2013): 837-882.
8. Sheth, Rishit, and Roni Khardon. "Monte Carlo Structured SVI for Two-Level Non-Conjugate Models." *arXiv preprint arXiv:1612.03957* (2016).
9. Khan and Lin. "Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models." *Alstats* (2017).
10. Khan and Nielsen. "Fast yet simple natural-gradient descent for variational inference in complex models." (2018) *ISITA*.
11. Zhang, Guodong, et al. "Noisy natural gradient as variational inference." *ICML* (2018).