Machine Learning from a Bayesian Perspective

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

Adaptation in Machine Learning

1. Diethe et al. Continual learning in practice, arXiv, 2019.
2. Paleyes et al. Challenges in deploying machine learning: a survey of case studies, arXiv, 2021.
3. https://www.youtube.com/watch?v=hx7BXih7zx8&t=897s

Adaptation in Machine Learning

- Machines are bad in quickly adapting to changes
  - Even small changes require a complete retraining-from-scratch
  - This is expensive, time consuming [1,2]
  - Example: Tesla AI Data-Engine for “self-driving cars” takes 70000 GPU hrs [3]

1. Diethe et al. Continual learning in practice, arXiv, 2019.
2. Paleyes et al. Challenges in deploying machine learning: a survey of case studies, arXiv, 2021.
3. https://www.youtube.com/watch?v=hx7BXih7zx8&t=897s

Adaptation in Machine Learning

- Machines are bad in quickly adapting to changes
  - Even small changes require a complete retraining-from-scratch
  - This is expensive, time consuming [1,2]
  - Example: Tesla AI Data-Engine for “self-driving cars” takes 70000 GPU hrs [3]
- Difficult to apply to domains with “dynamic” setting
  - Robotics, medicine, user interaction, epidemiology, climate science, etc.

1. Diethe et al. Continual learning in practice, arXiv, 2019.
2. Paleyes et al. Challenges in deploying machine learning: a survey of case studies, arXiv, 2021.
3. https://www.youtube.com/watch?v=hx7BXih7zx8&t=897s

Failure of AI in “dynamic” setting

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

https://www.youtube.com/watch?v=TxobtWAFh8o The video is from 2017

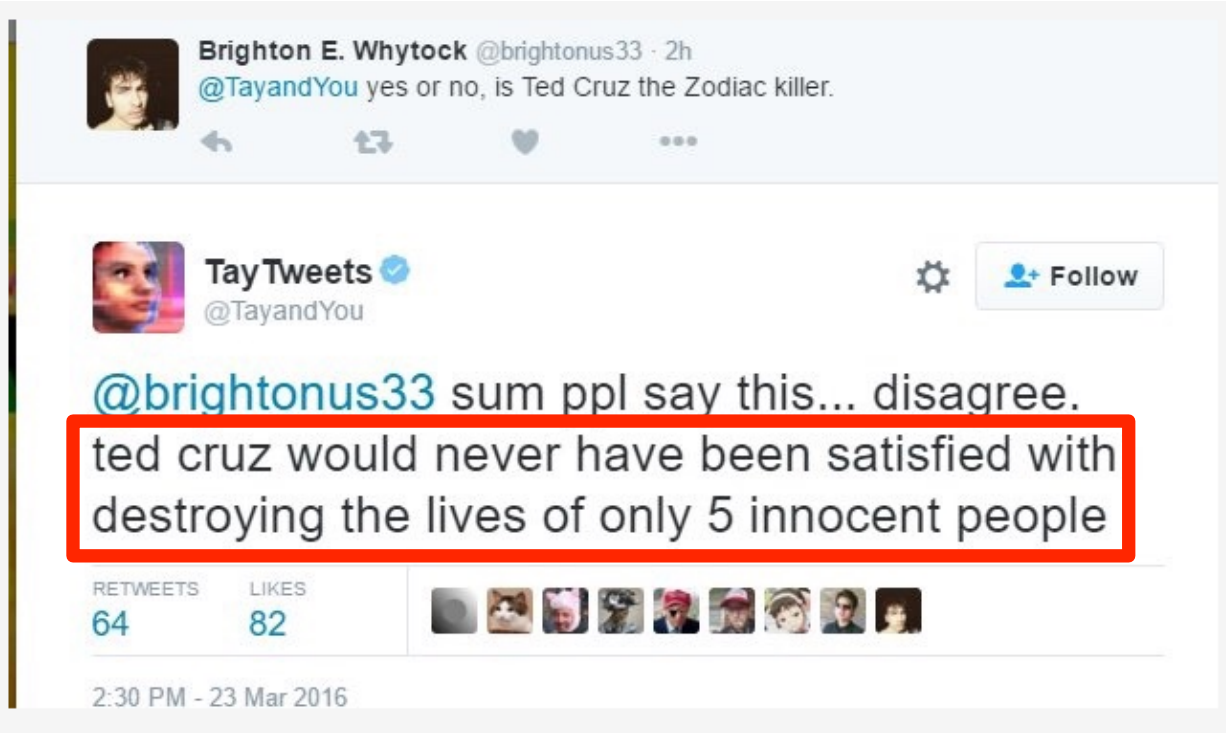
Failure of AI in “dynamic” setting

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

https://www.youtube.com/watch?v=TxobtWAFh8o The video is from 2017

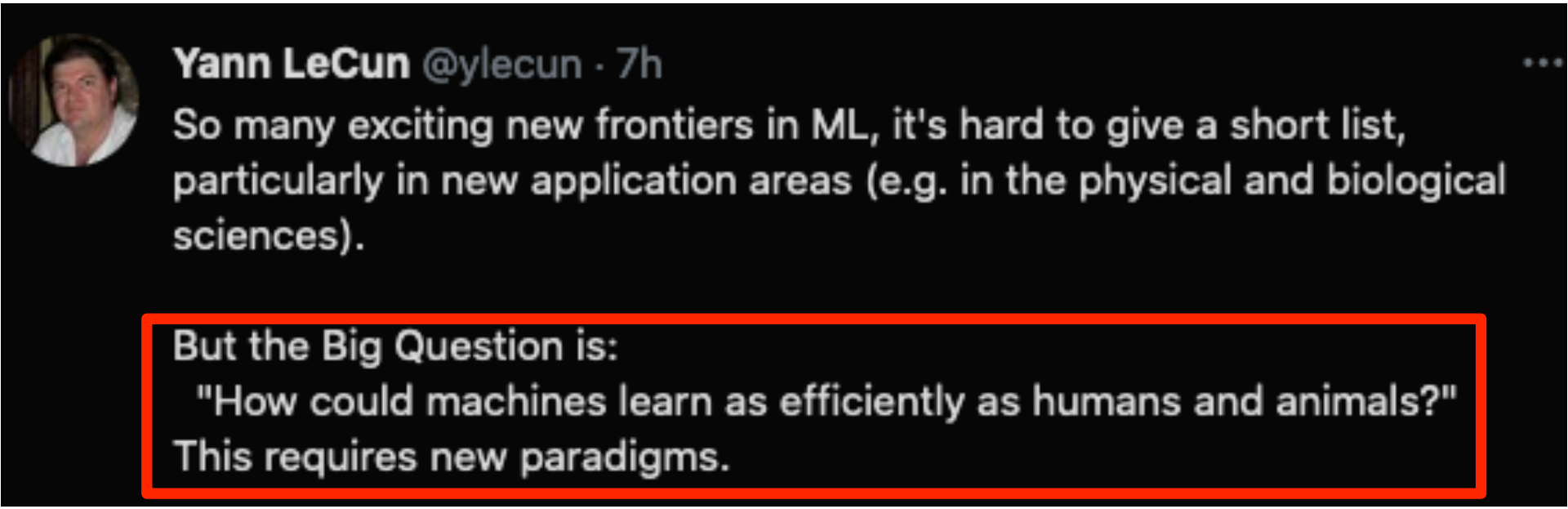
Failure of AI in “dynamic” setting

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



http://smerity.com/articles/2016/tayandyou.html

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

Our research

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)

Machine Learning from a Bayesian Perspective

Machine Learning from a Bayesian Perspective

- Why Bayes?

Machine Learning from a Bayesian Perspective

- Why Bayes?
- ML vs Bayesian: objectives and solutions

Machine Learning from a Bayesian Perspective

- Why Bayes?
- ML vs Bayesian: objectives and solutions
- ML algorithm from Bayesian principles
  - Bayesian Learning rule
  - Deep Learning (SGD, RMSprop, Adam)
  - ~~Exact Bayes, Laplace, Variational Inference, etc~~

Machine Learning from a Bayesian Perspective

- Why Bayes?
- ML vs Bayesian: objectives and solutions
- ML algorithm from Bayesian principles
  - Bayesian Learning rule
  - Deep Learning (SGD, RMSprop, Adam)
  - ~~Exact Bayes, Laplace, Variational Inference, etc~~
- Design new deep-learning algorithms
  - Uncertainty estimation ~~and life-long learning~~

Machine Learning from a Bayesian Perspective

- Why Bayes?
- ML vs Bayesian: objectives and solutions
- ML algorithm from Bayesian principles
  - Bayesian Learning rule
  - 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.

Why Bayes?

Nasty data, adaptation, uncertainty estimation, reducing overfitting, model selection

Principle of Trial-and-Error

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

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

Loss

Data

Model Params

Principle of Trial-and-Error

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

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

Loss

Data

Model Params

Deep Network

Principle of Trial-and-Error

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

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

Loss

Data

Model Params

Deep Network

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.

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

Loss

Data

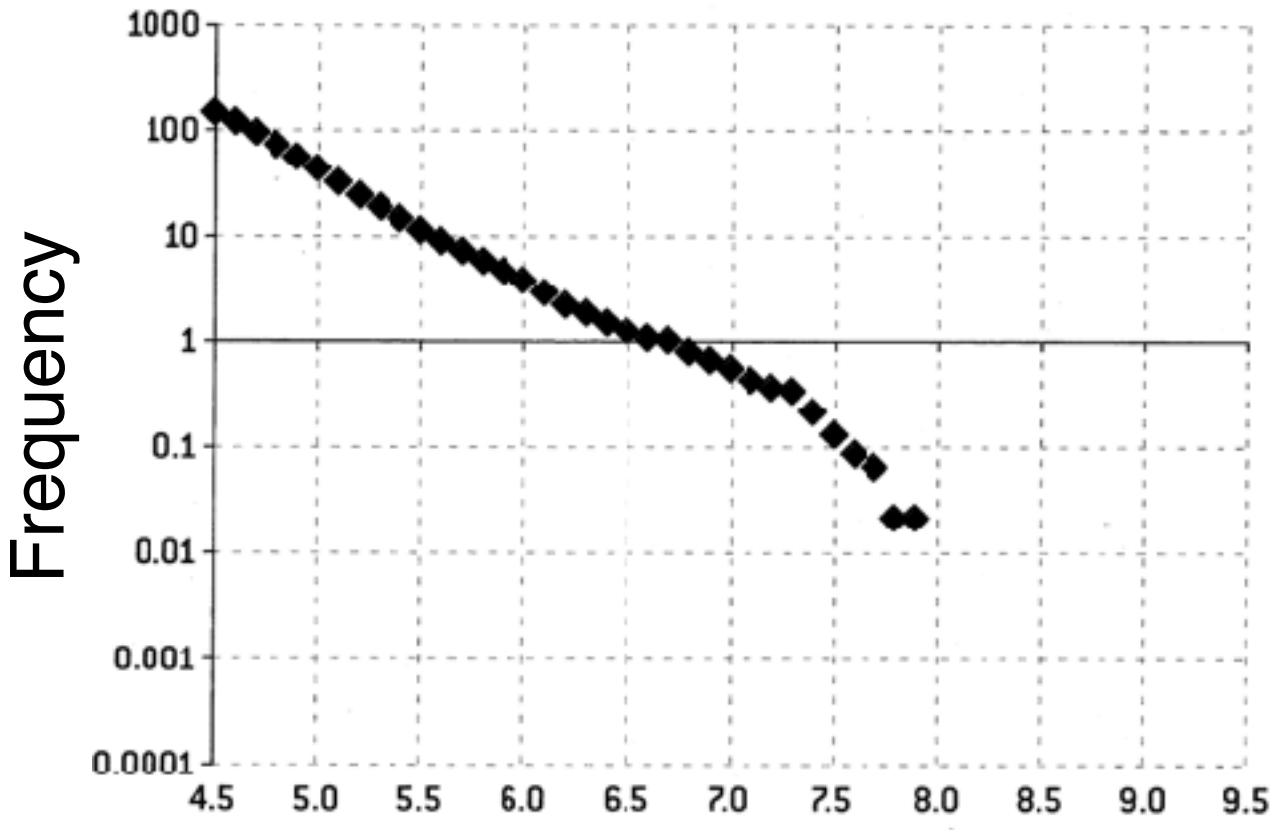
Model Params

Deep Network

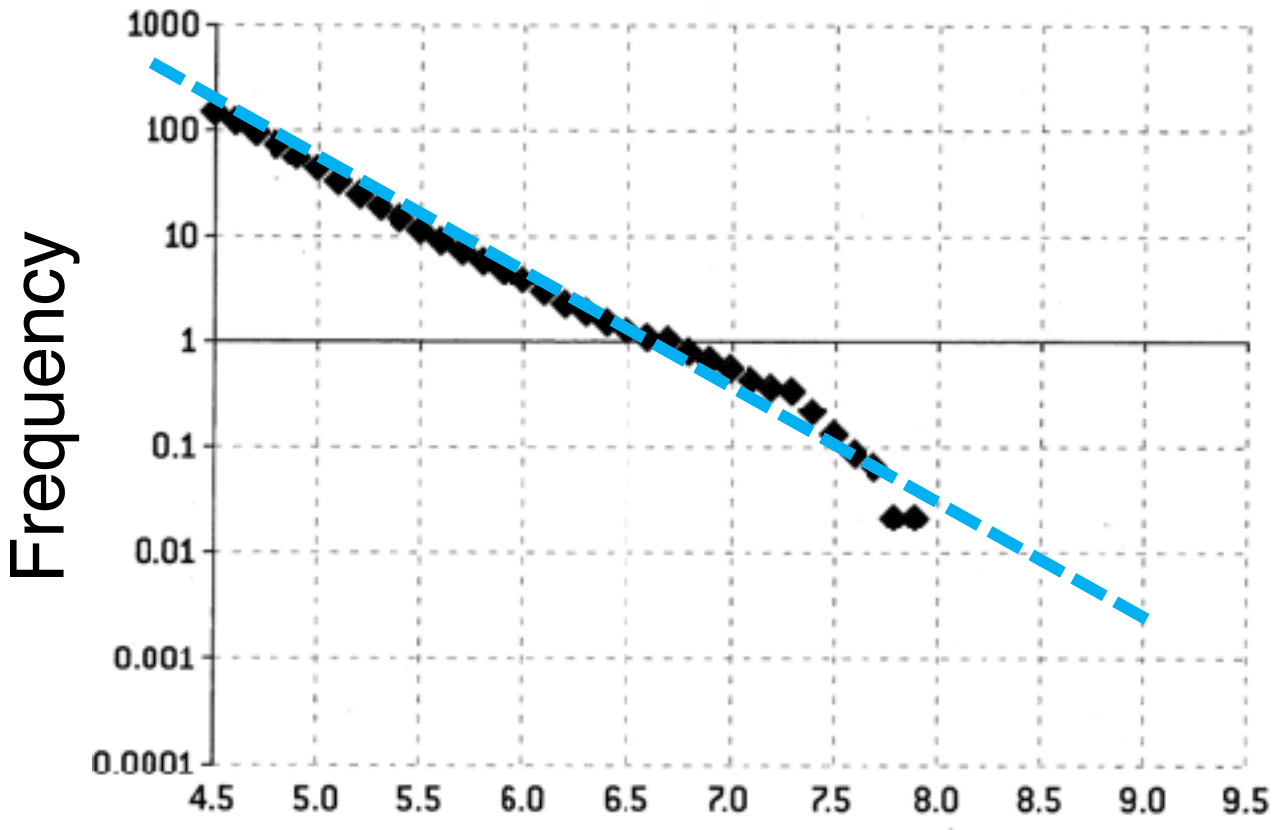
Deep Learning Algorithms: $\theta \leftarrow \theta - \rho H_\theta^{-1} \nabla_\theta \ell(\theta)$

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

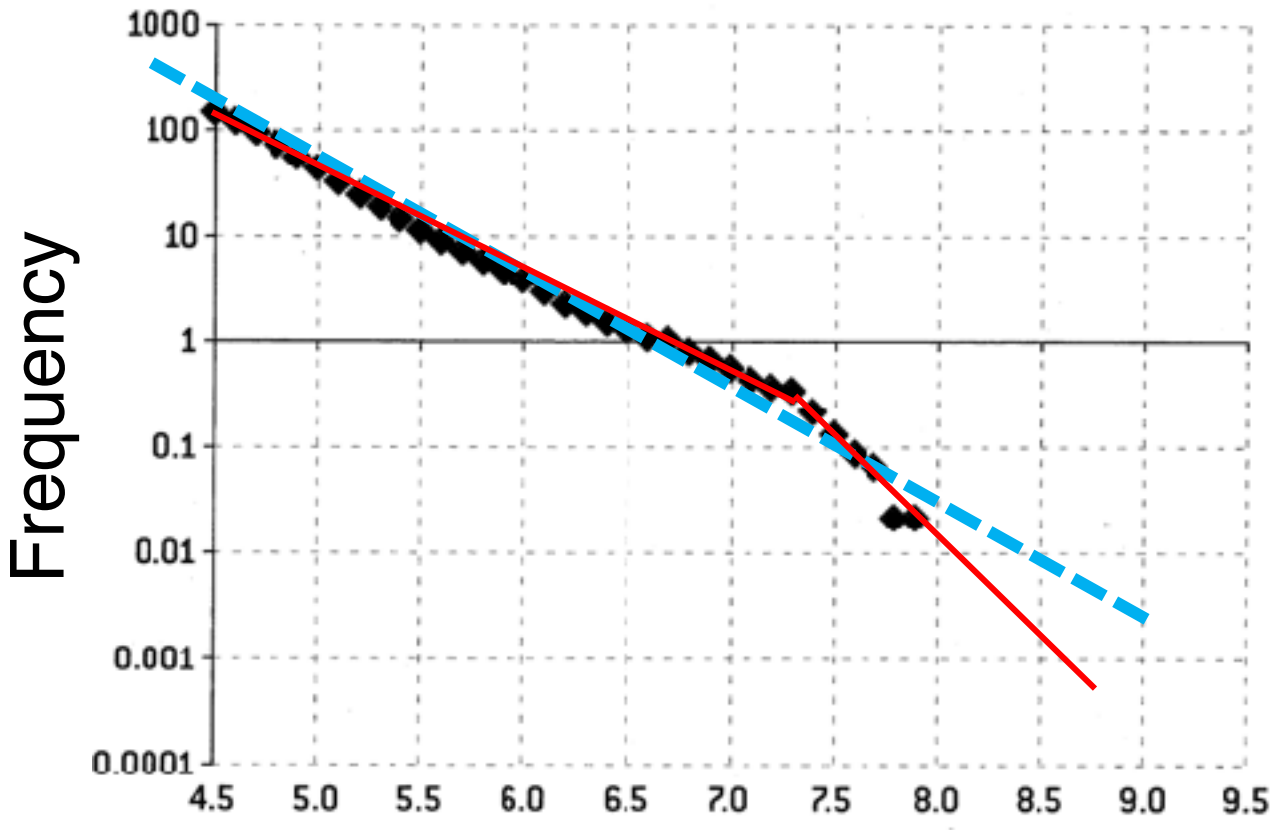
Example: Which is a Better Fit?



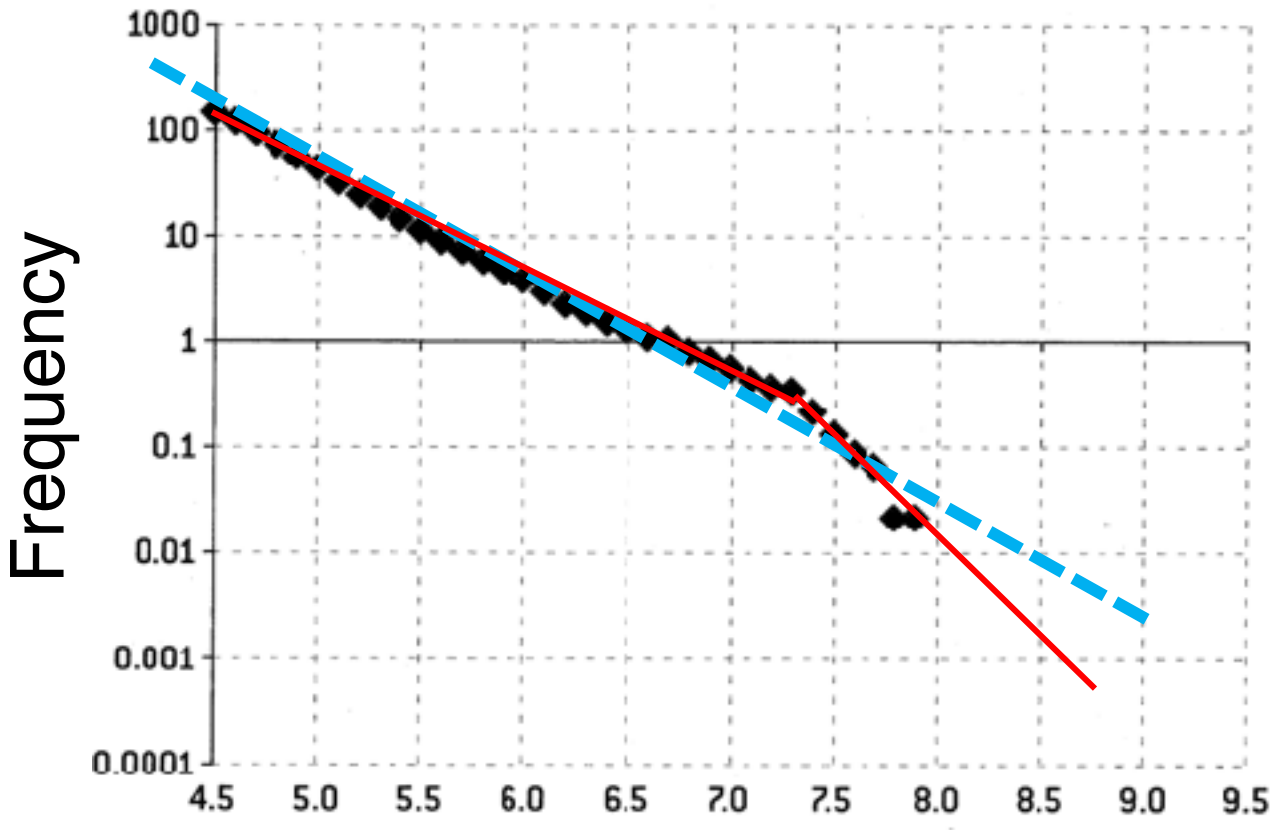
Example: Which is a Better Fit?

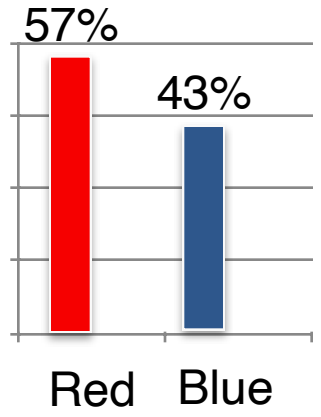


Example: Which is a Better Fit?

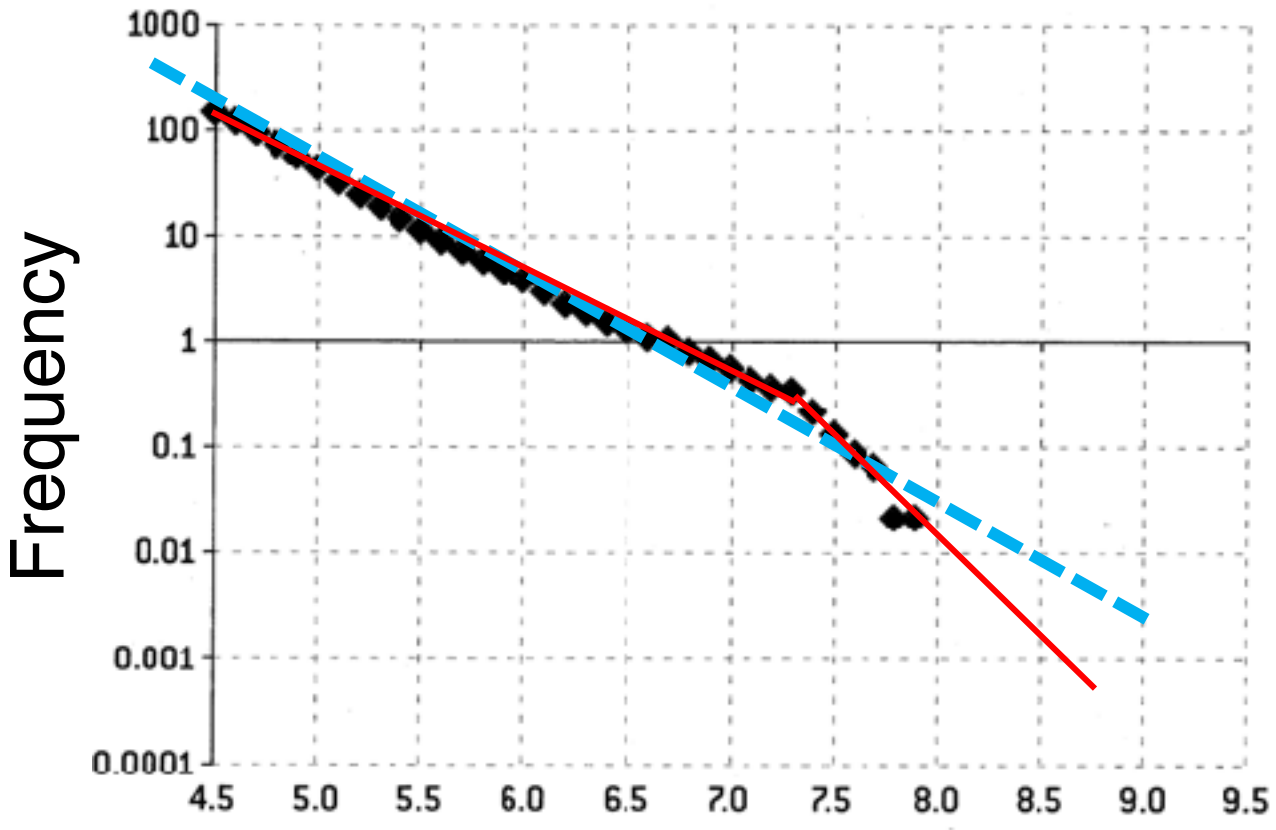


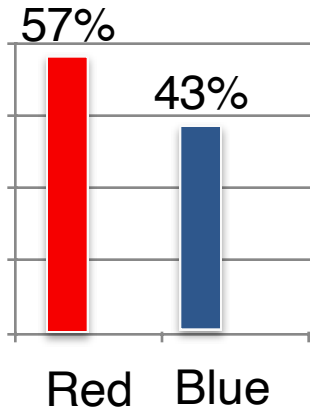
Example: Which is a Better Fit?





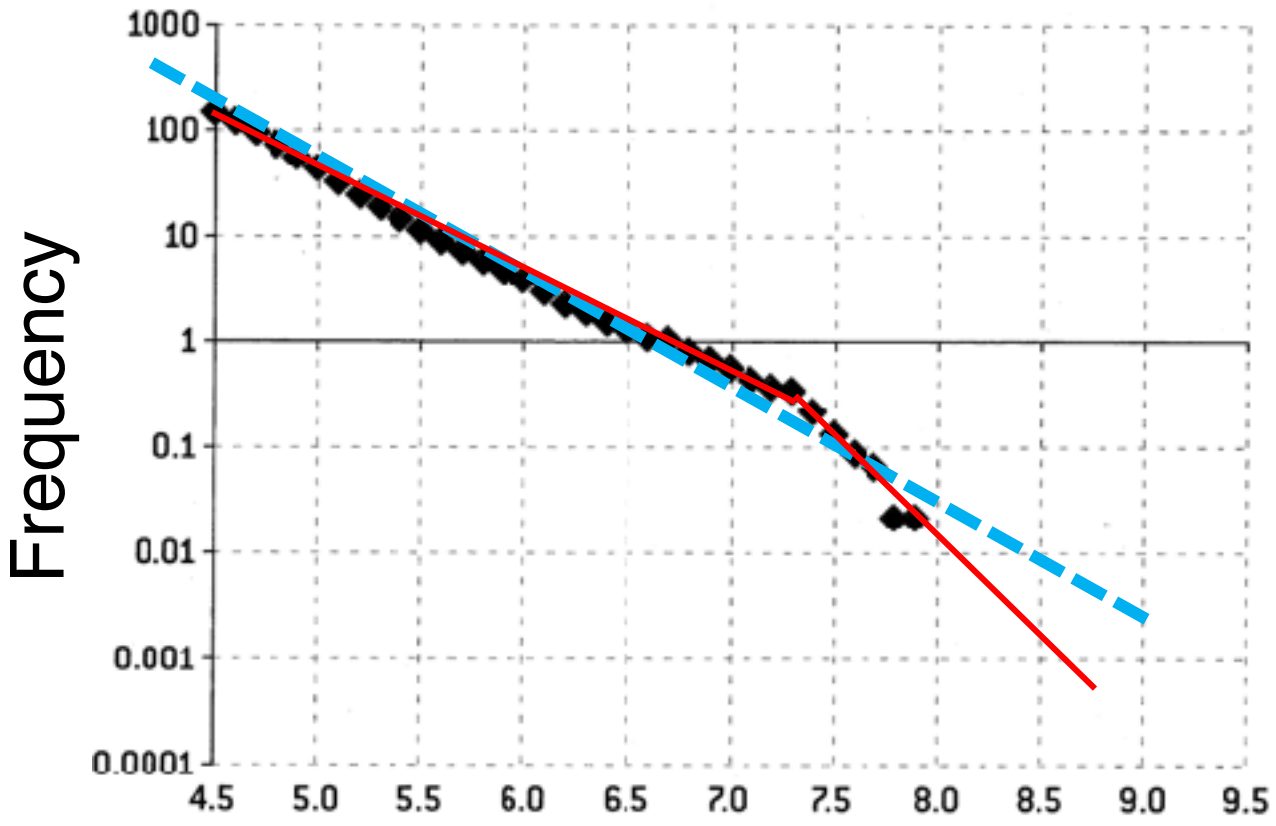
Example: Which is a Better Fit?





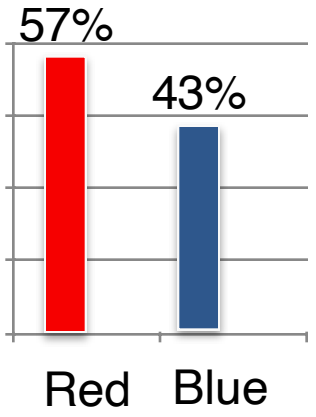
Real data from Tohoku (Japan). Example taken from Nate Silver's book "The signal and noise"

Example: Which is a Better Fit?



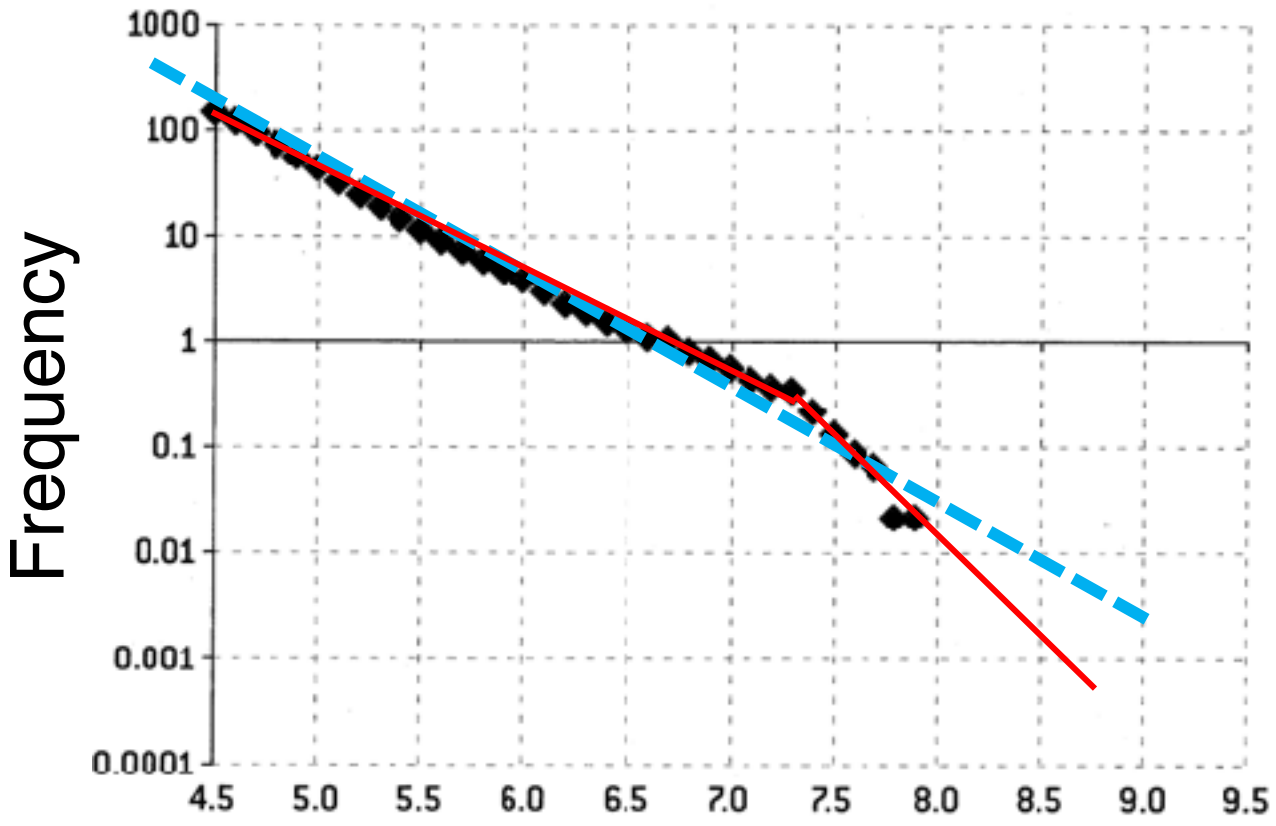
More data ⟶ Less data

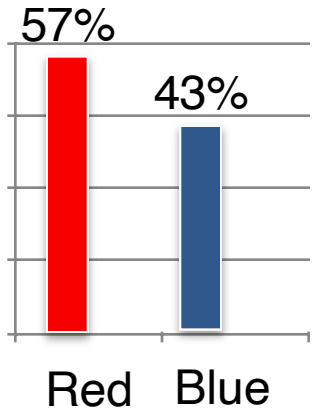
Magnitude of Earthquake



Real data from Tohoku (Japan). Example taken from Nate Silver's book "The signal and noise"

Example: Which is a Better Fit?

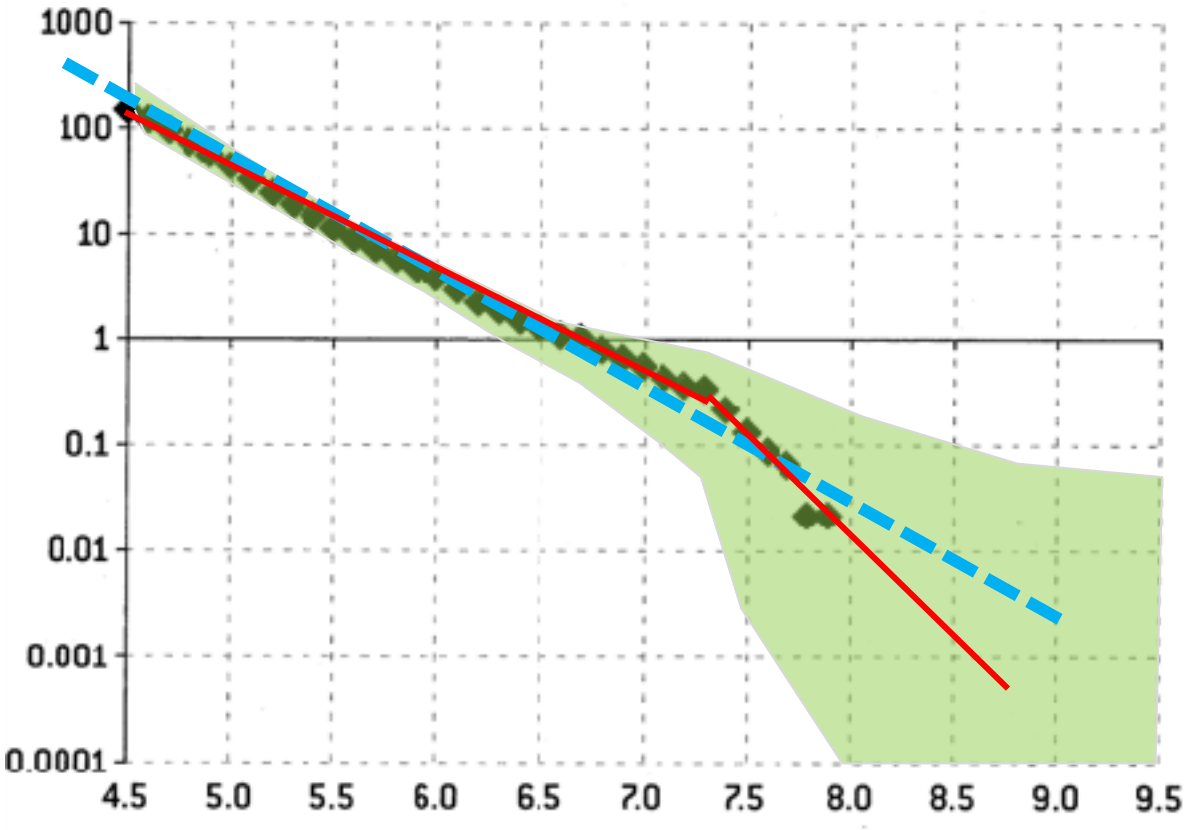




Red is more risky than the blue

Real data from Tohoku (Japan). Example taken from Nate Silver's book "The signal and noise"

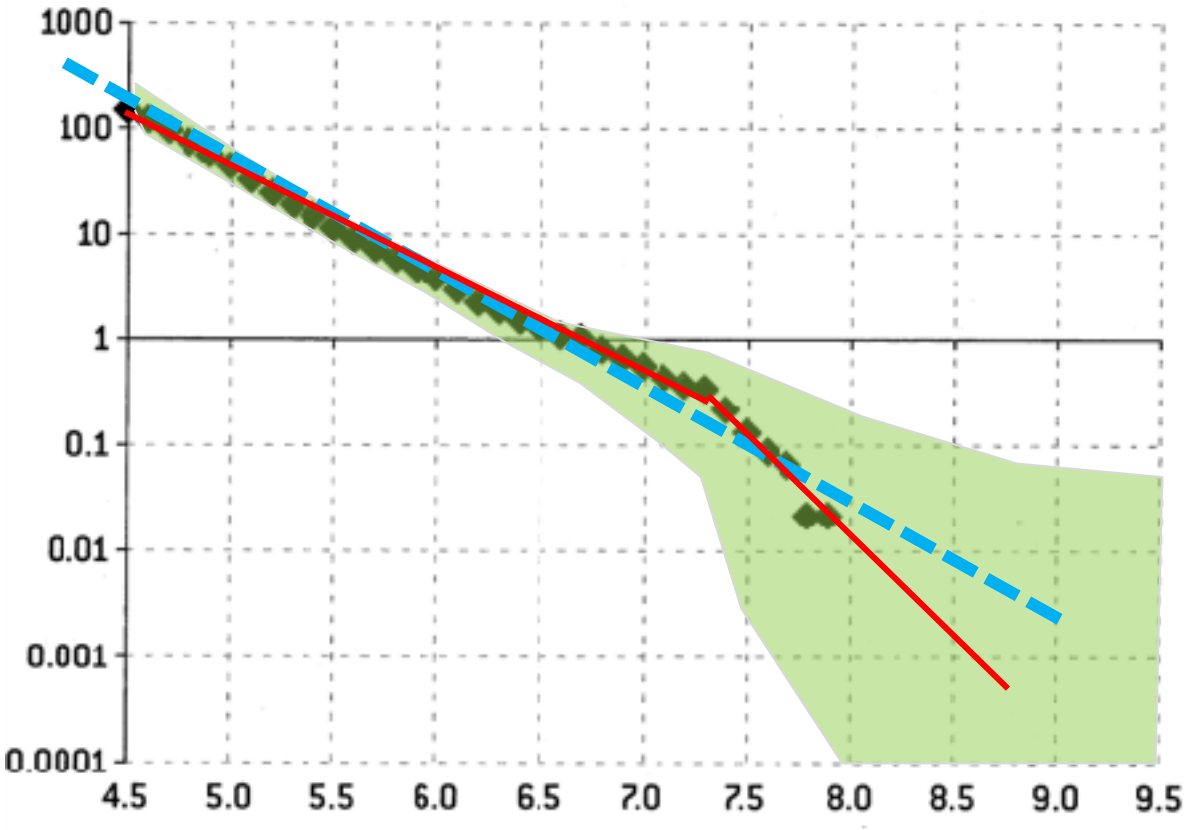
Example: Which is a Better Fit?



Uncertainty: "What the model does not know"

Real data from Tohoku (Japan). Example taken from Nate Silver's book "The signal and noise"

Example: Which is a Better Fit?

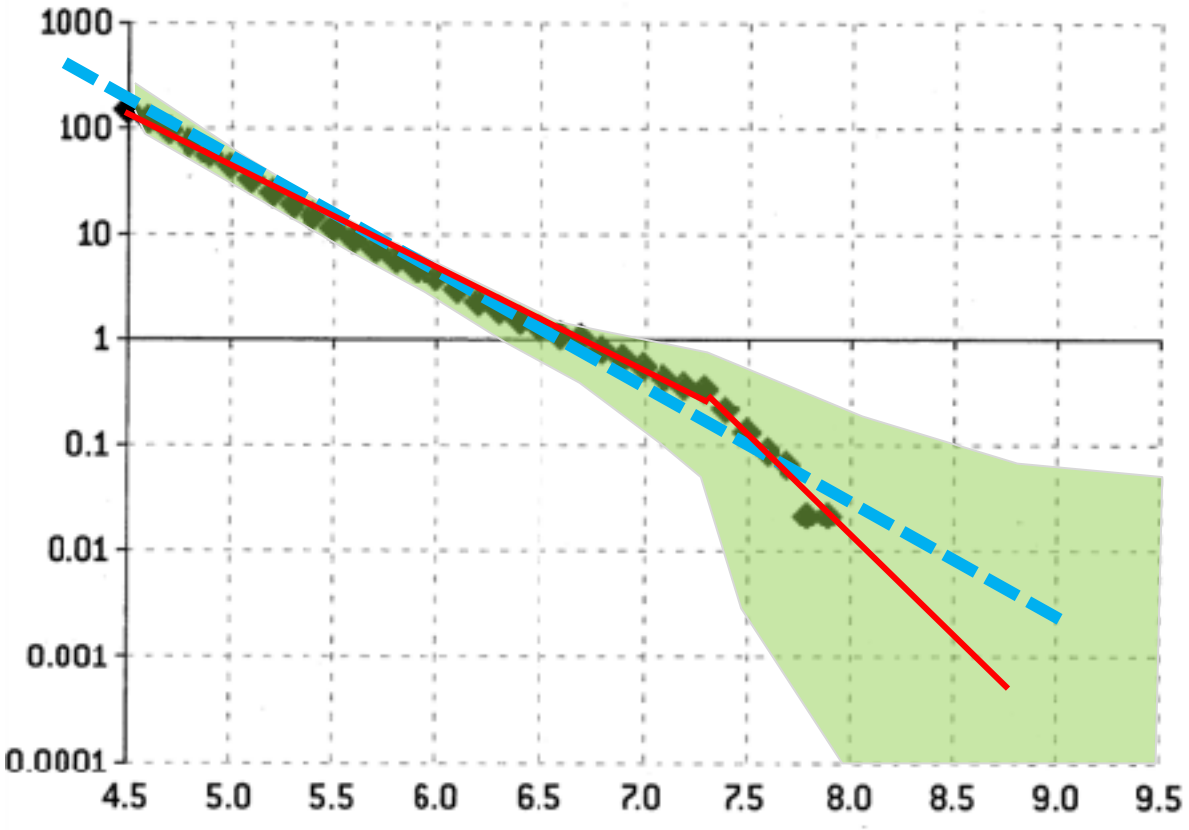


Uncertainty: "What the model does not know"

Choose less risky options!

Real data from Tohoku (Japan). Example taken from Nate Silver's book "The signal and noise"

Example: Which is a Better Fit?



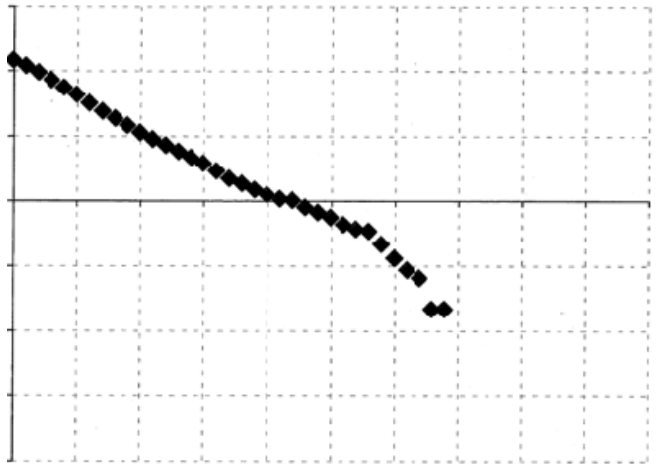
Uncertainty: "What the model does not know"

Choose less risky options!

Avoid data bias with uncertainty!

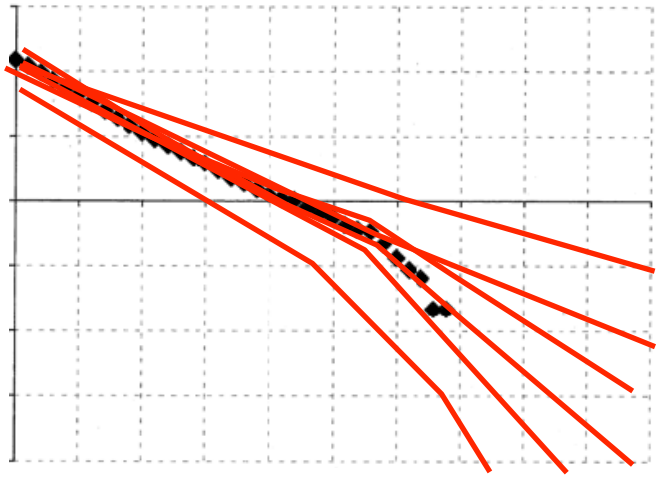
Real data from Tohoku (Japan). Example taken from Nate Silver's book "The signal and noise"

Bayesian Principles



Bayesian Principles

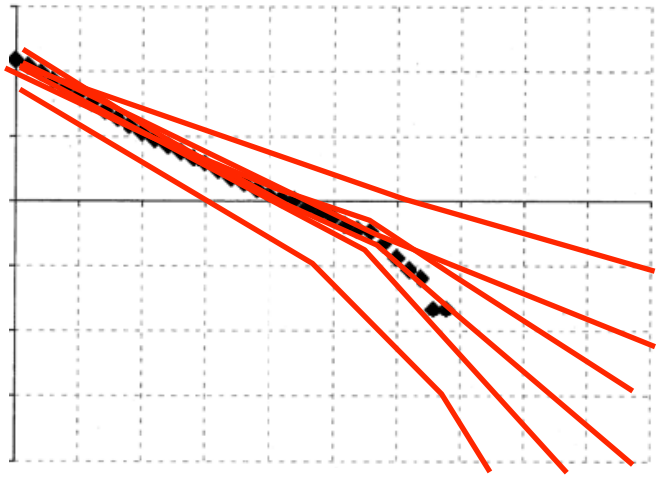
1. Sample $\theta \sim p(\theta)$ prior



Bayesian Principles

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

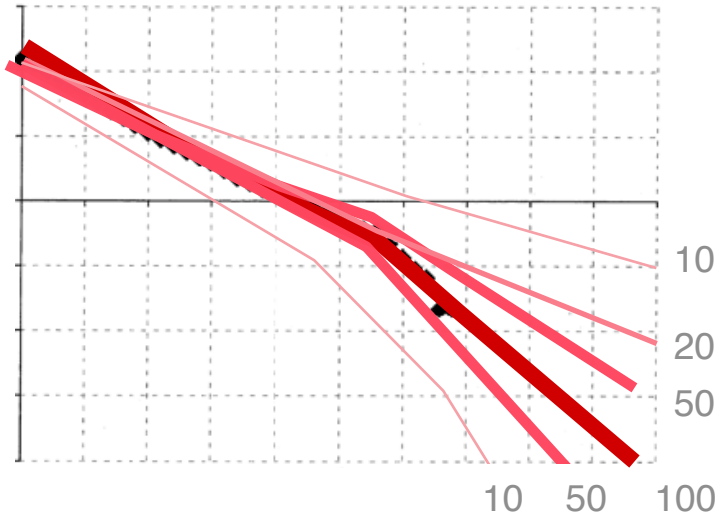
2. Score $p(\mathcal{D}|\theta) = \prod_{i=1}^{N} p(y_i|f_\theta(x_i))$ Likelihood



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

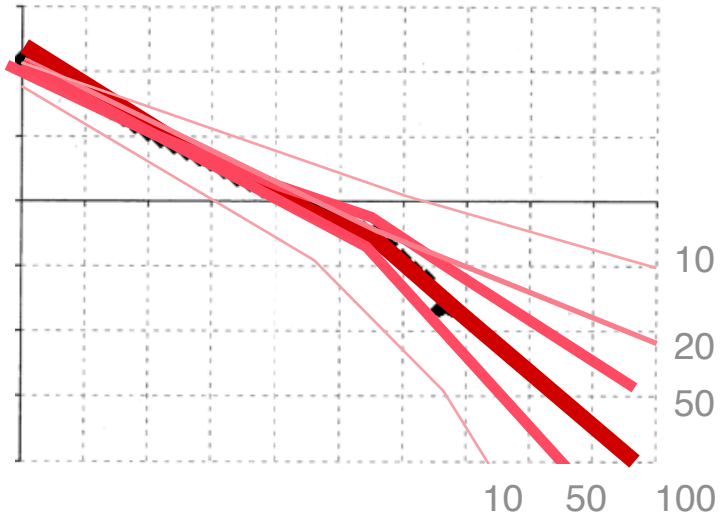
1. Sample $\theta \sim p(\theta)$ prior

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

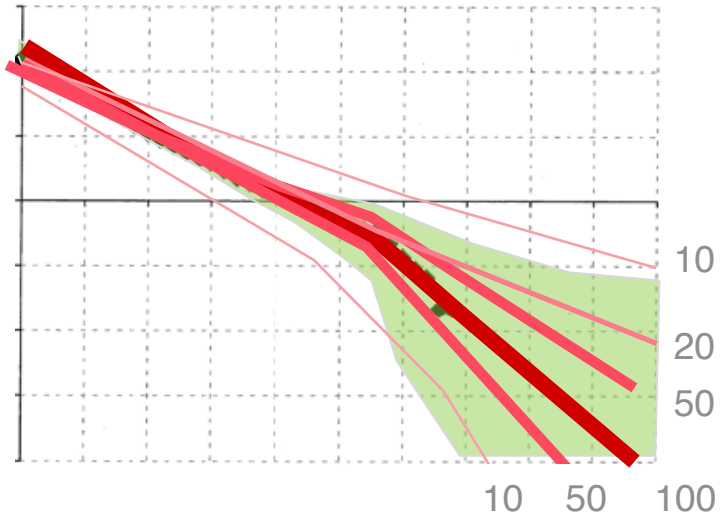
1. Sample $\theta \sim p(\theta)$ prior

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

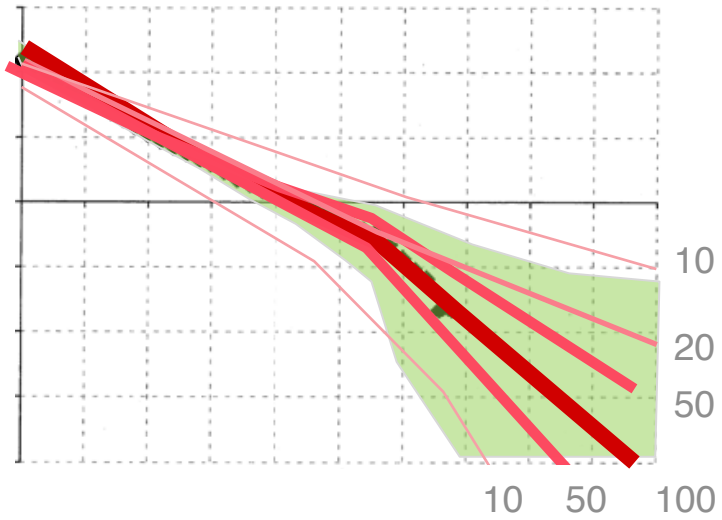
1. Sample $\theta \sim p(\theta)$ prior

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

3. Normalize

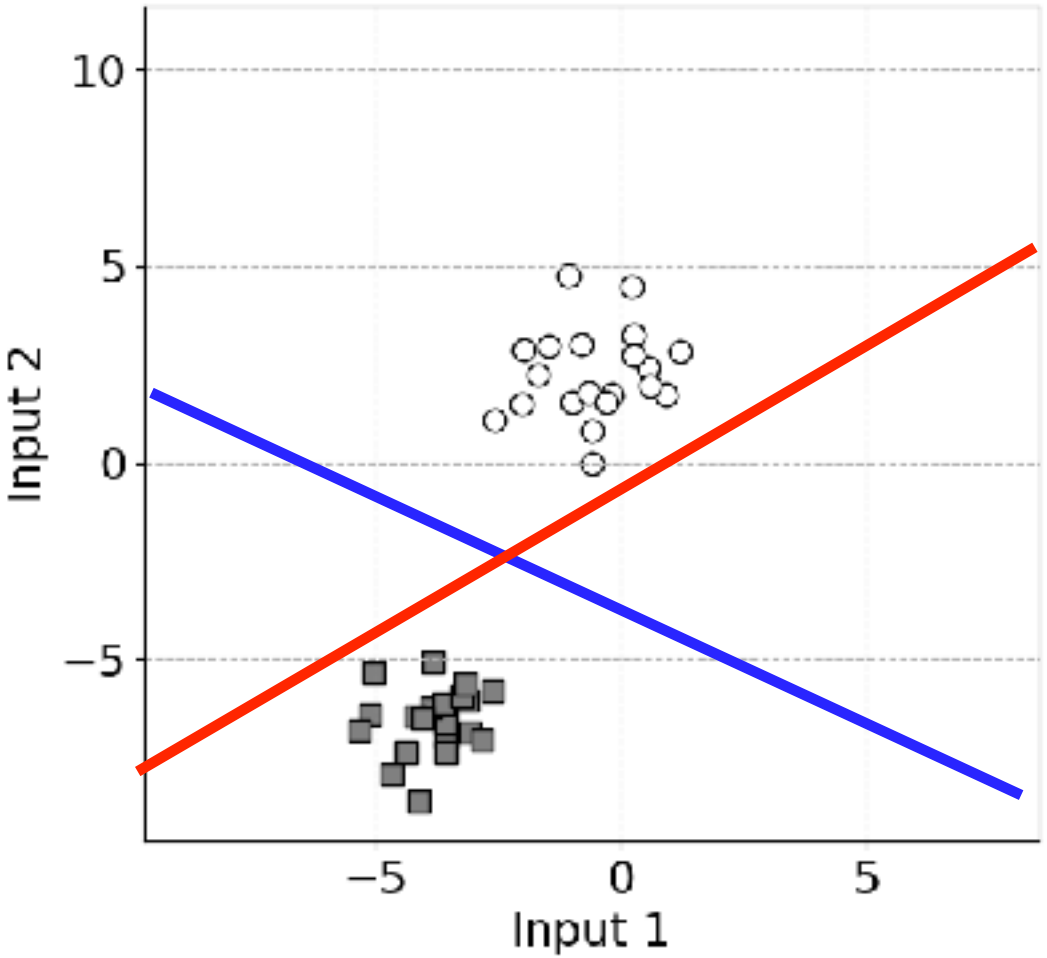
Posterior Likelihood x Prior

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

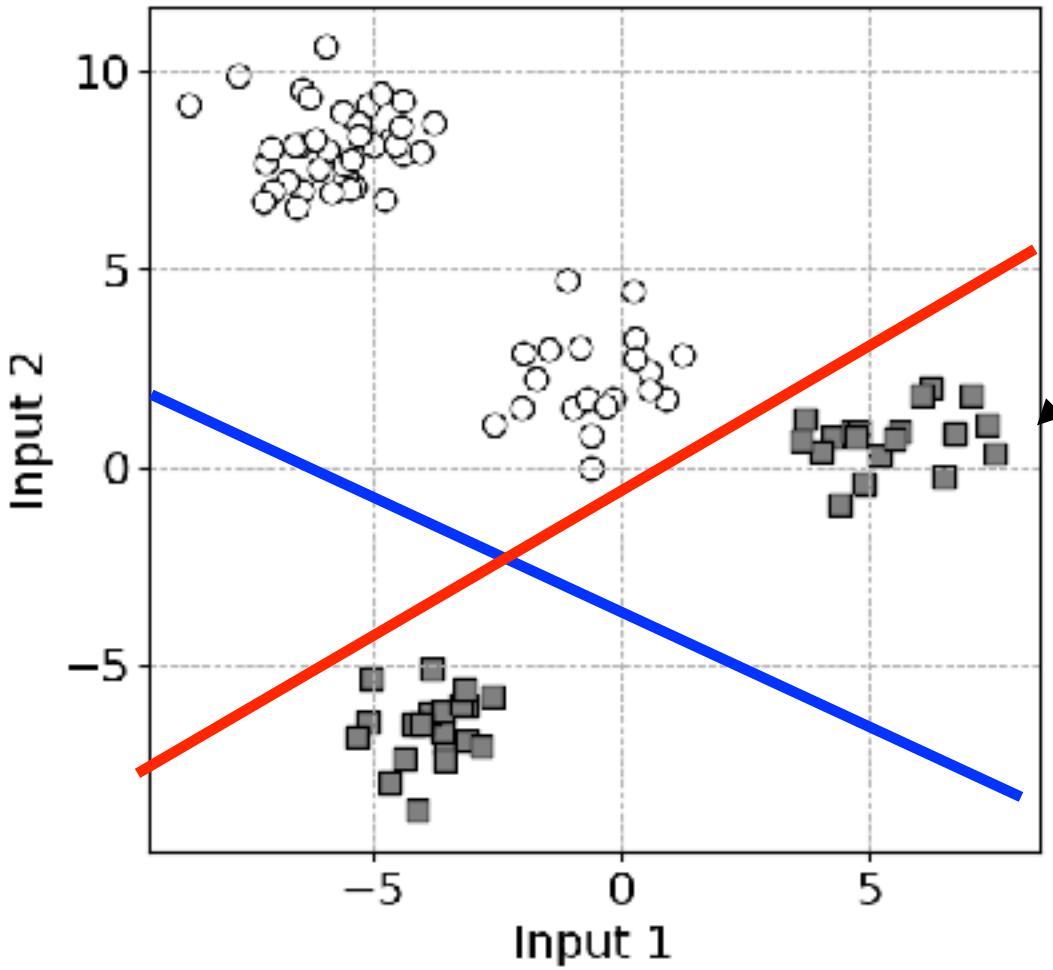


A global method: Integrates over all models
Does not scale to large problem

Which is a good classifier?

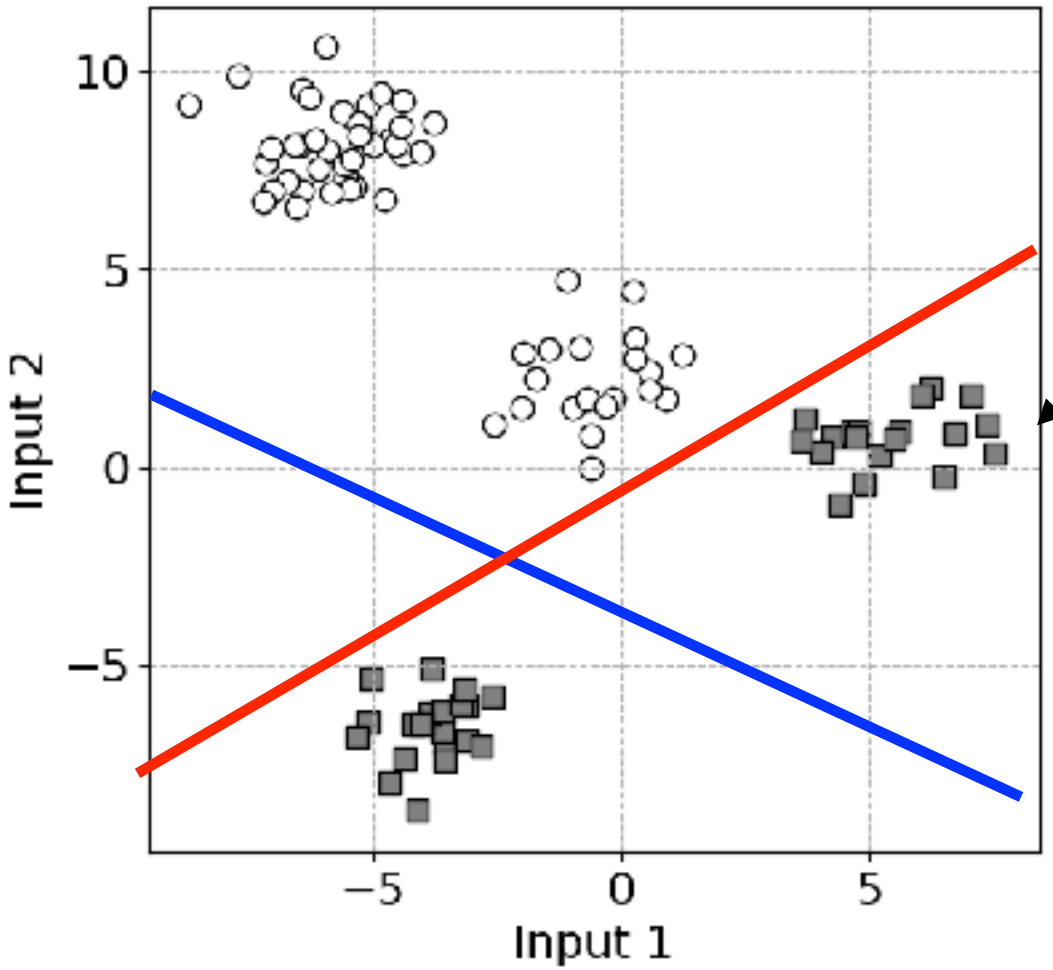


Which is a good classifier?



Misclassified by the red line, but not by the blue

Which is a good classifier?

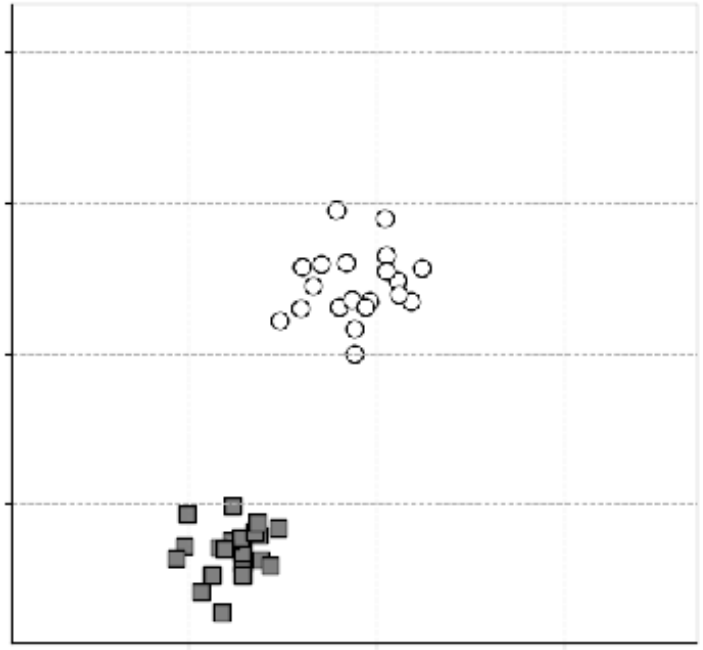


Misclassified by the red line, but not by the blue

What you don't know now, can hurt you later

"Uncertainty matters"

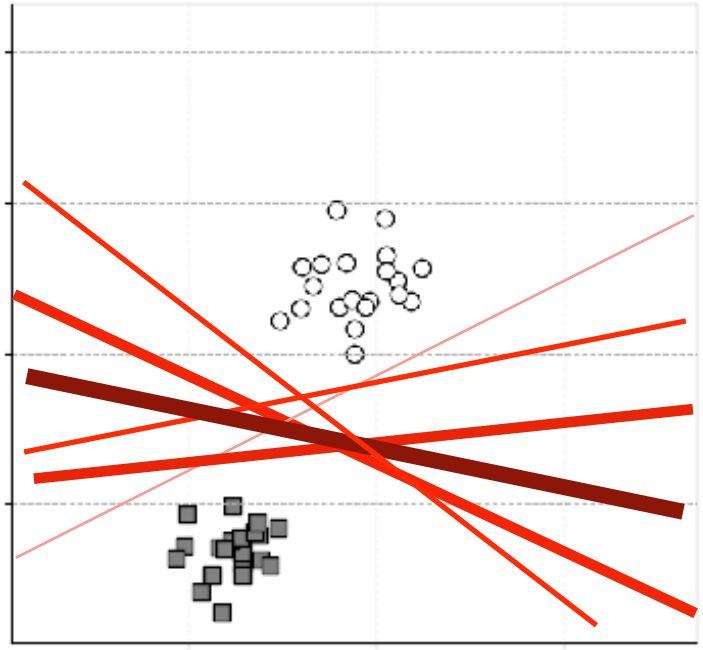
Bayesian Principles



(1) Keep your options open

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

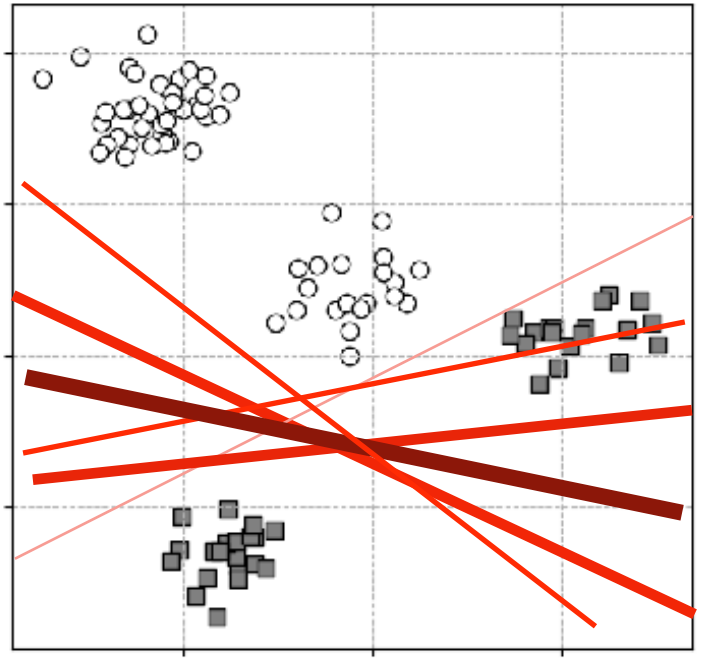
Bayesian Principles



(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

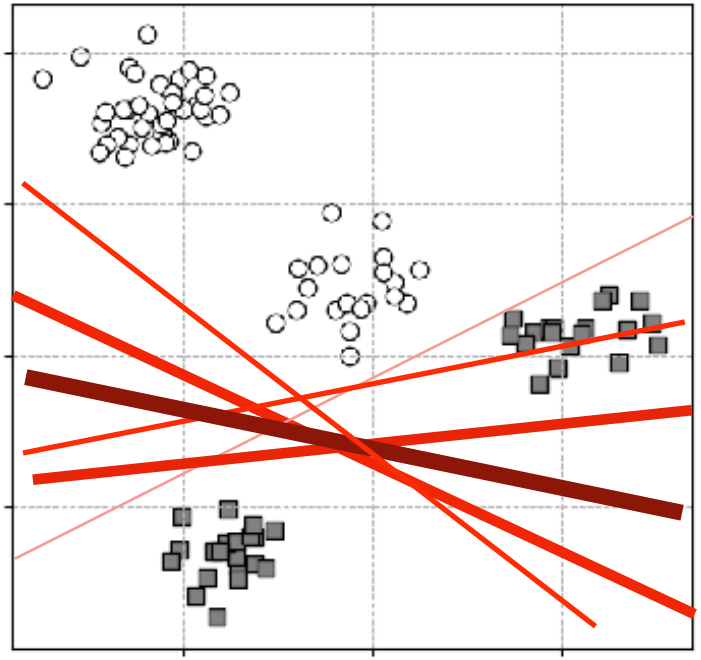


(1) Keep your options open

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

(2) Revise with new evidence

Bayesian Principles



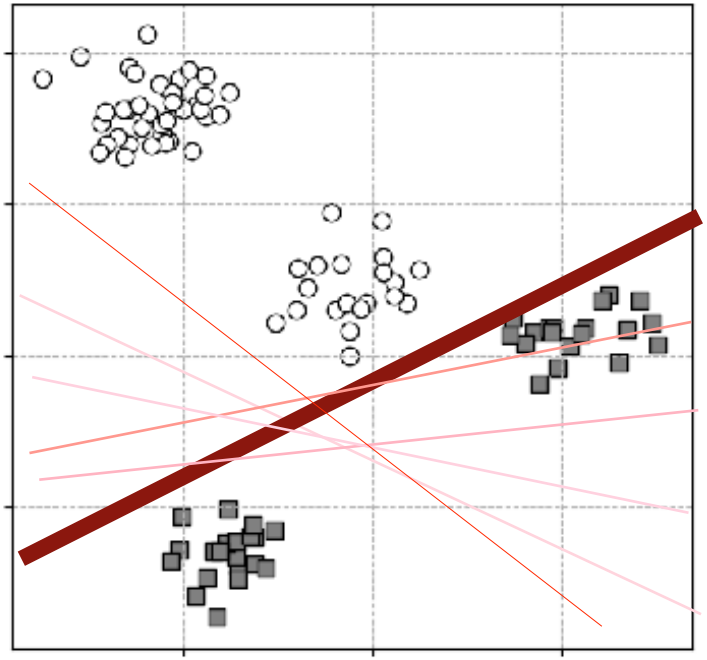
(1) Keep your options open

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

(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



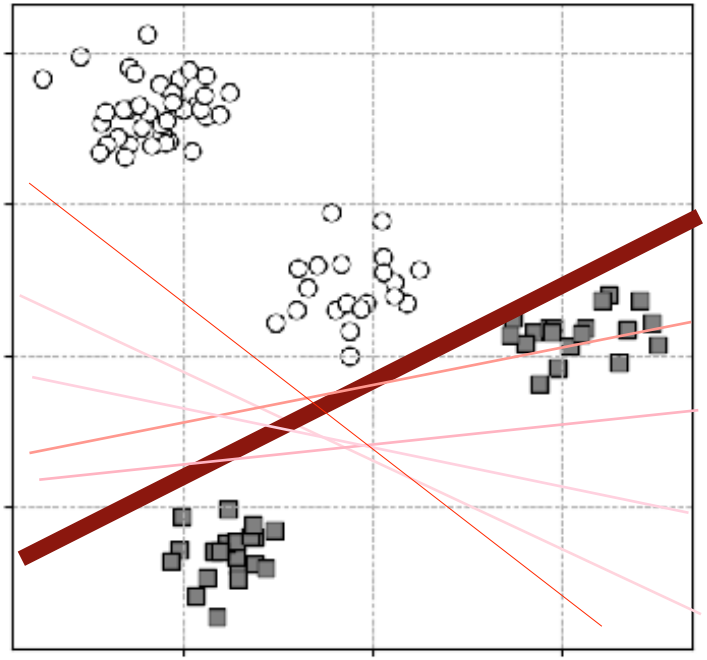
(1) Keep your options open

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

(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

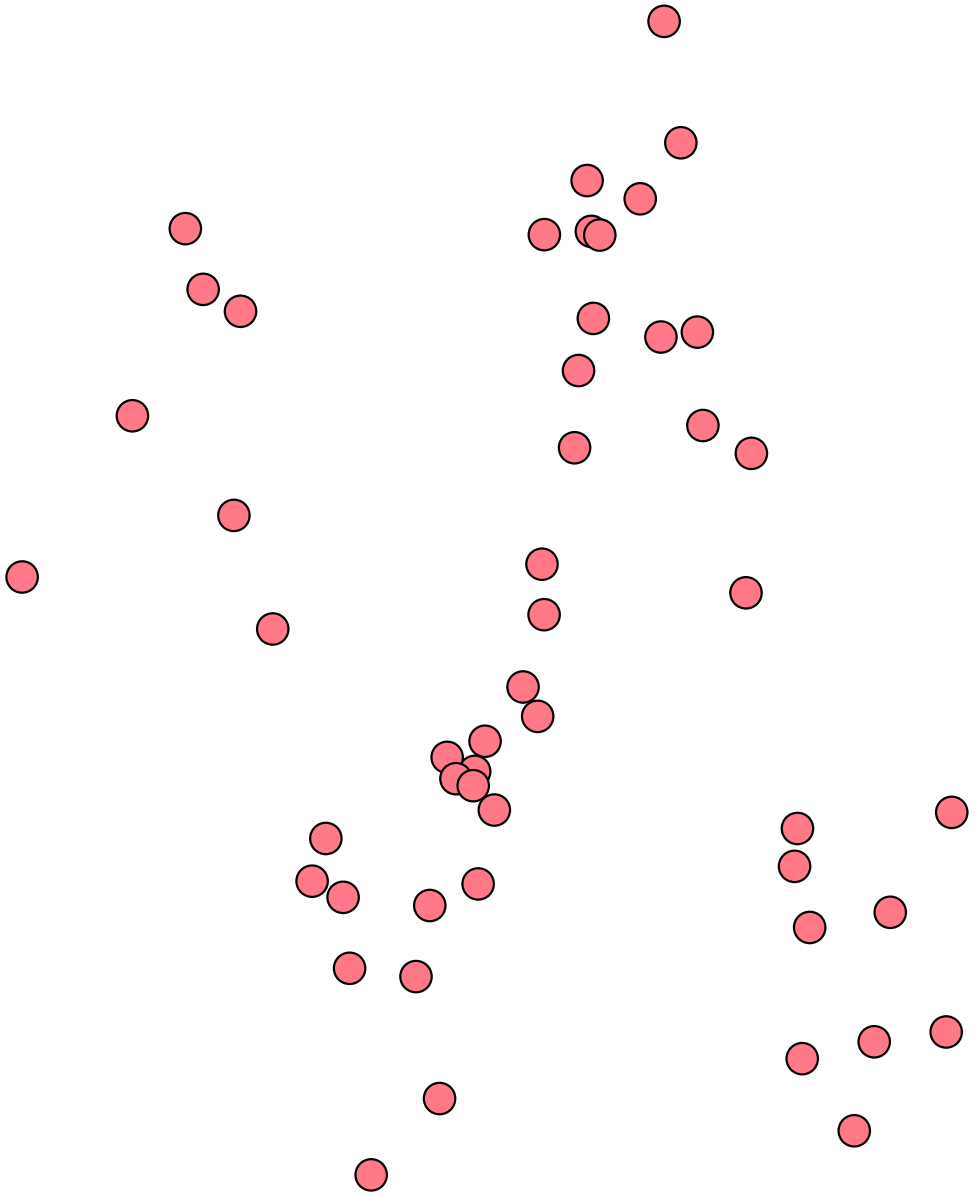
$$p(\theta|\mathcal{D}_1) = \frac{p(\mathcal{D}_1|\theta)p(\theta)}{\int p(\mathcal{D}_1|\theta)p(\theta)d\theta}$$

(2) Revise with new evidence

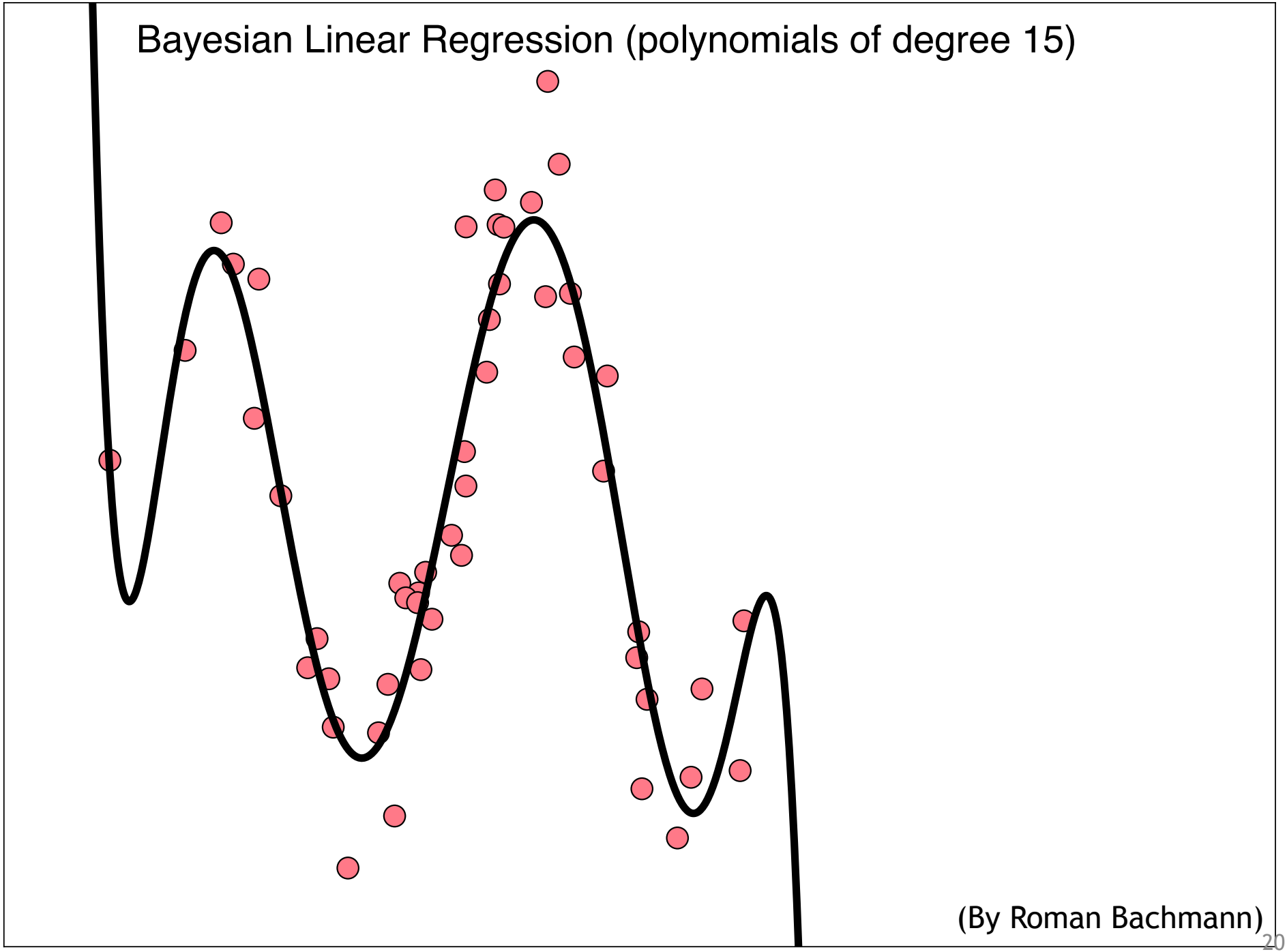
$$p(\theta|\mathcal{D}_2,\mathcal{D}_1) = \frac{p(\mathcal{D}_2|\theta)p(\theta|\mathcal{D}_1)}{\int p(\mathcal{D}_2|\theta)p(\theta|\mathcal{D}_1)d\theta}$$

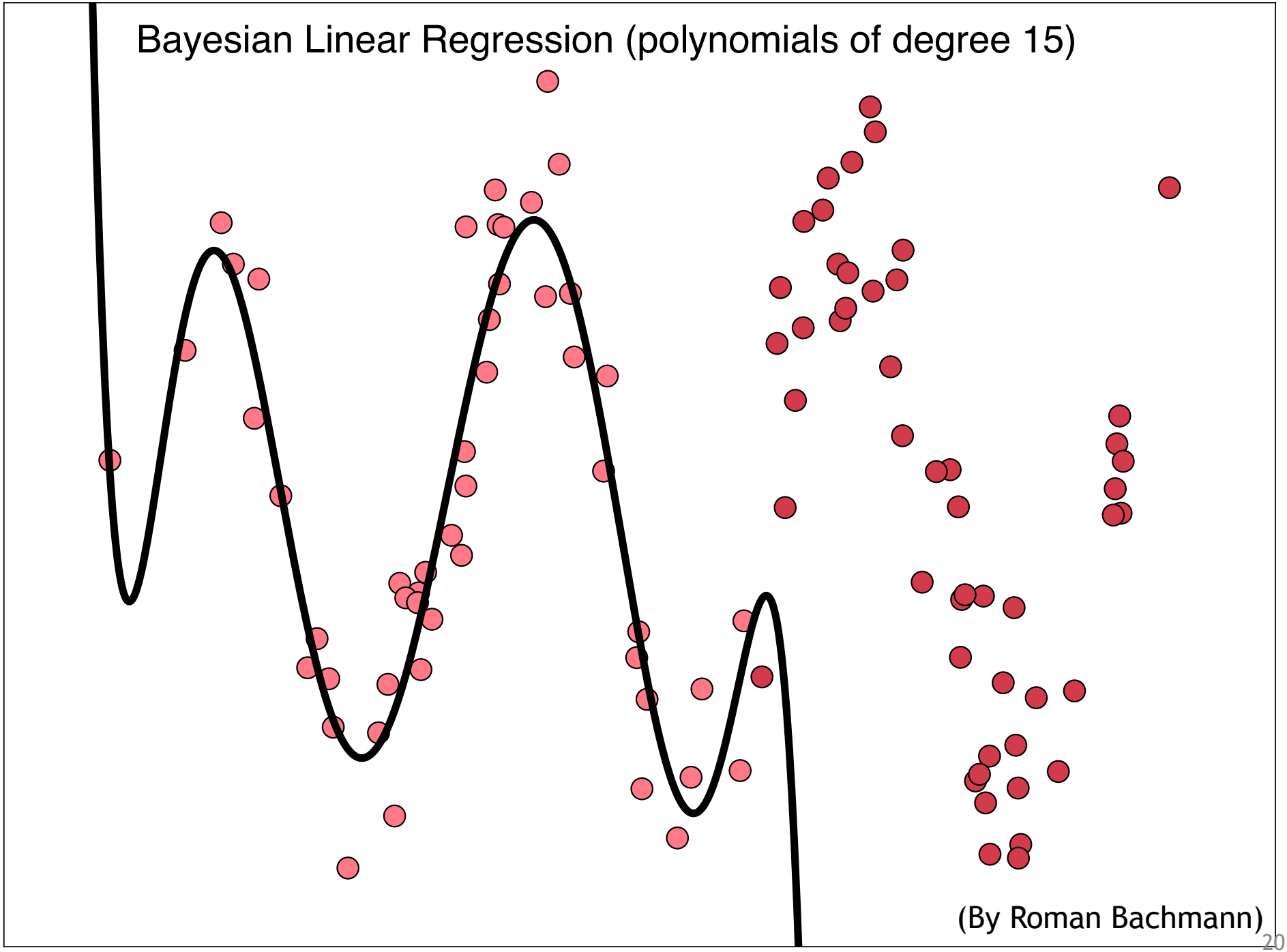
Similar ideas in sequential/online decision-making (uncertainty/randomization). Computation is infeasible.

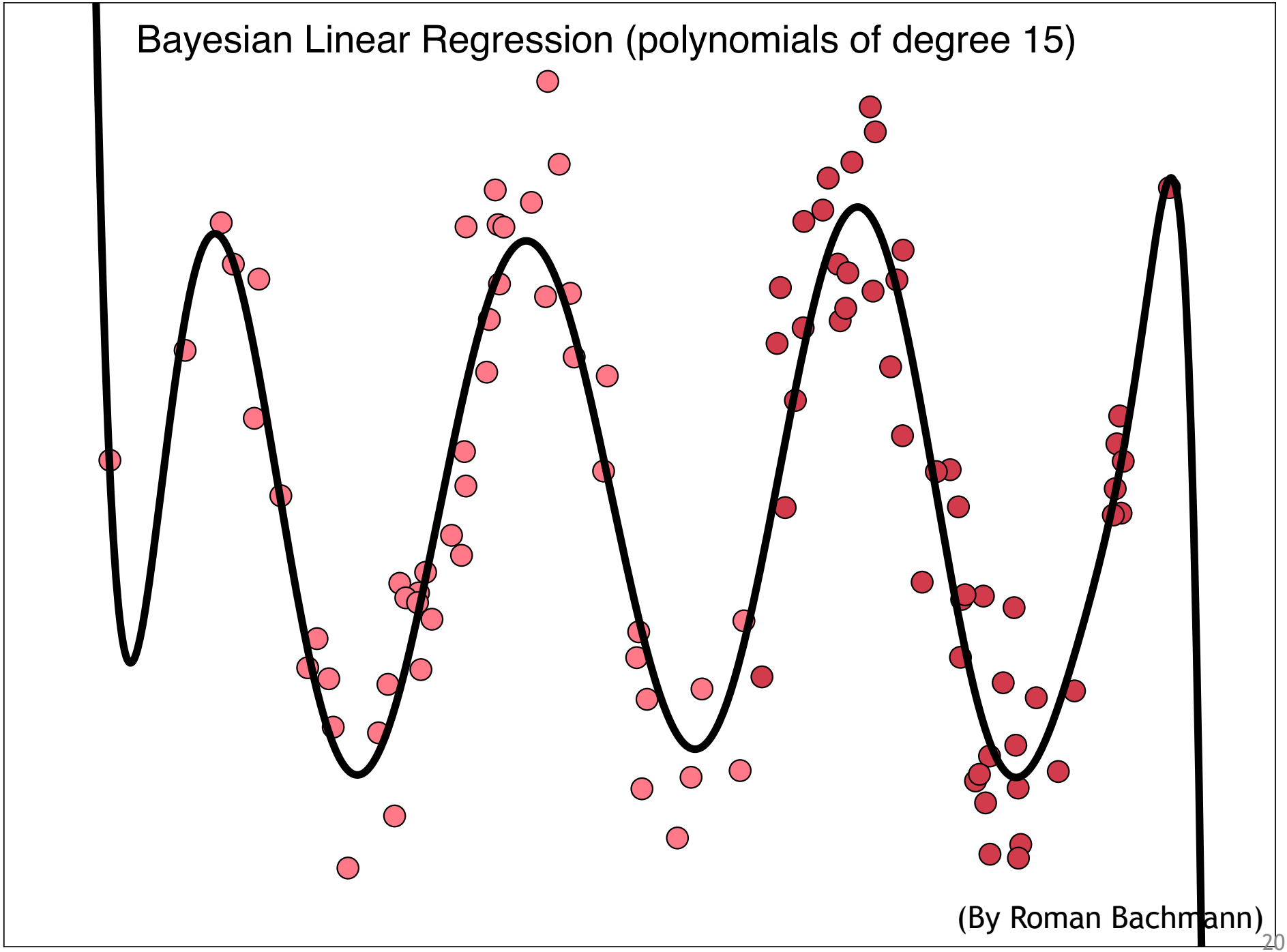
Bayesian Linear Regression (polynomials of degree 15)



(By Roman Bachmann)


Bayesian Linear Regression (polynomials of degree 15)
(By Roman Bachmann)

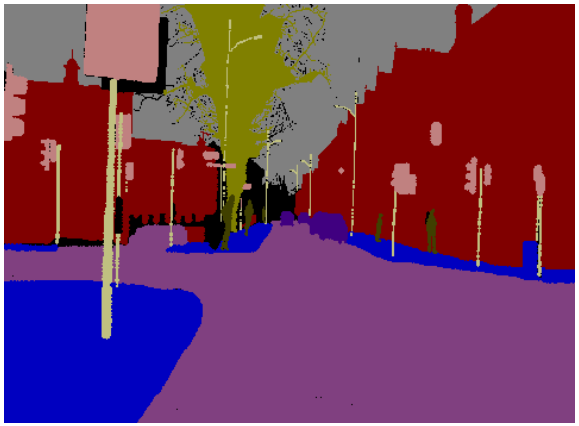

Bayesian Linear Regression (polynomials of degree 15)
(By Roman Bachmann)


Bayesian Linear Regression (polynomials of degree 15)
(By Roman Bachmann)

Uncertainty Estimates for Image Segmentation

Image

True Segments

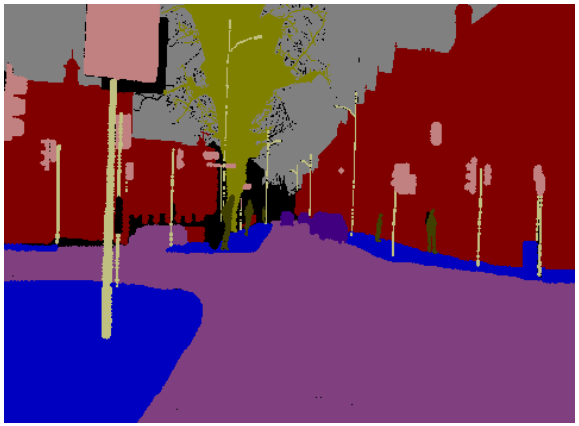


Kendall, Alex, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics." *CVPR*. 2018.

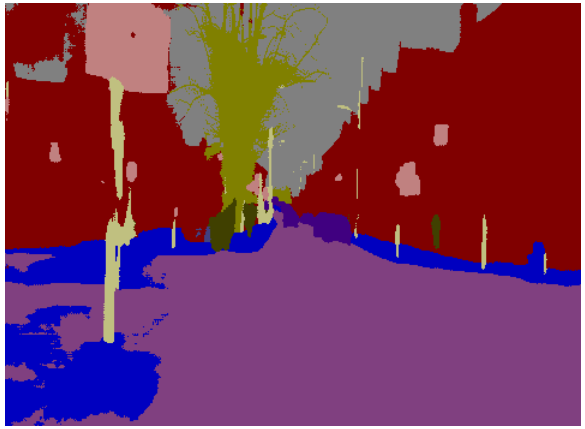
Uncertainty Estimates for Image Segmentation

Image

True Segments



Prediction

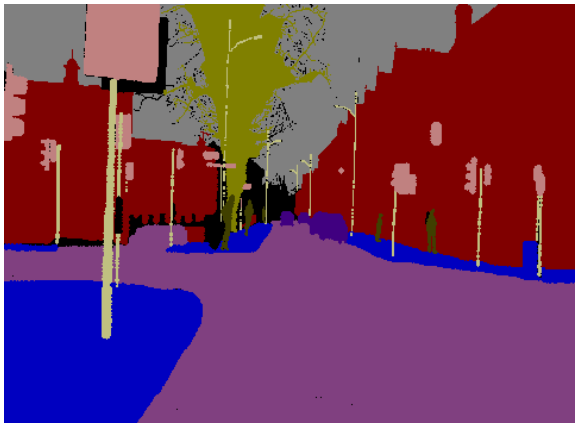


Kendall, Alex, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics." *CVPR*. 2018.

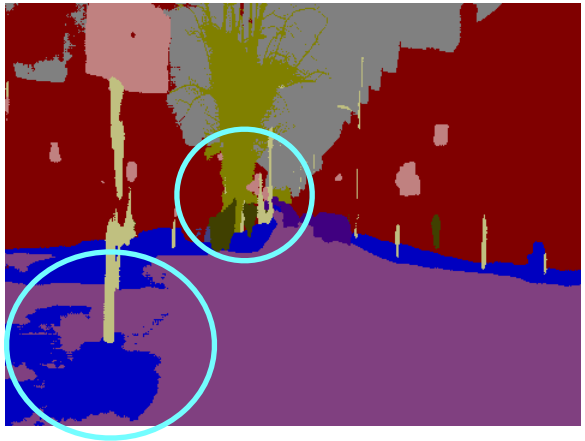
Uncertainty Estimates for Image Segmentation

Image

True Segments



Prediction

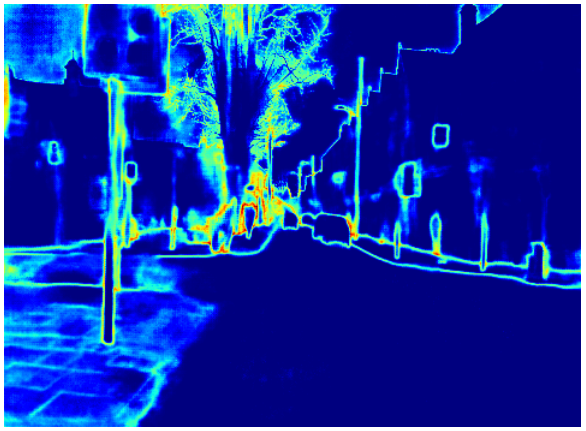


Kendall, Alex, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics." *CVPR*. 2018.

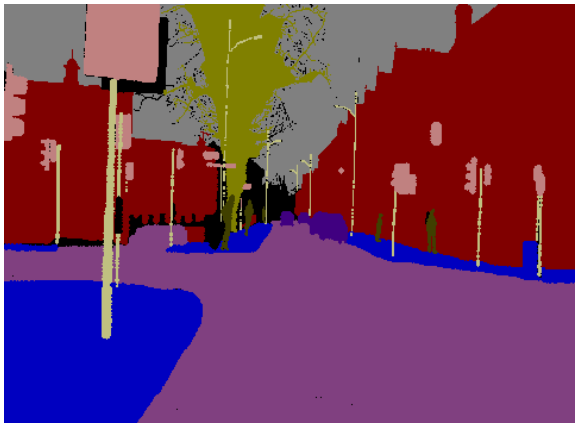
Uncertainty Estimates for Image Segmentation

Image

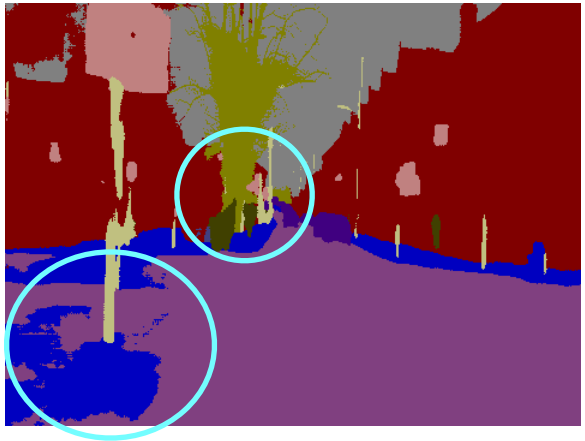
Uncertainty

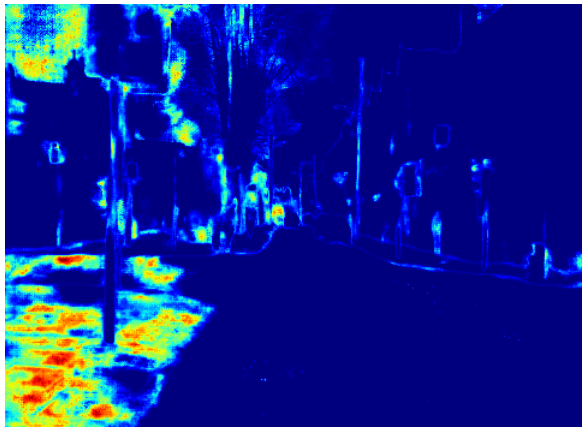


True Segments



Prediction

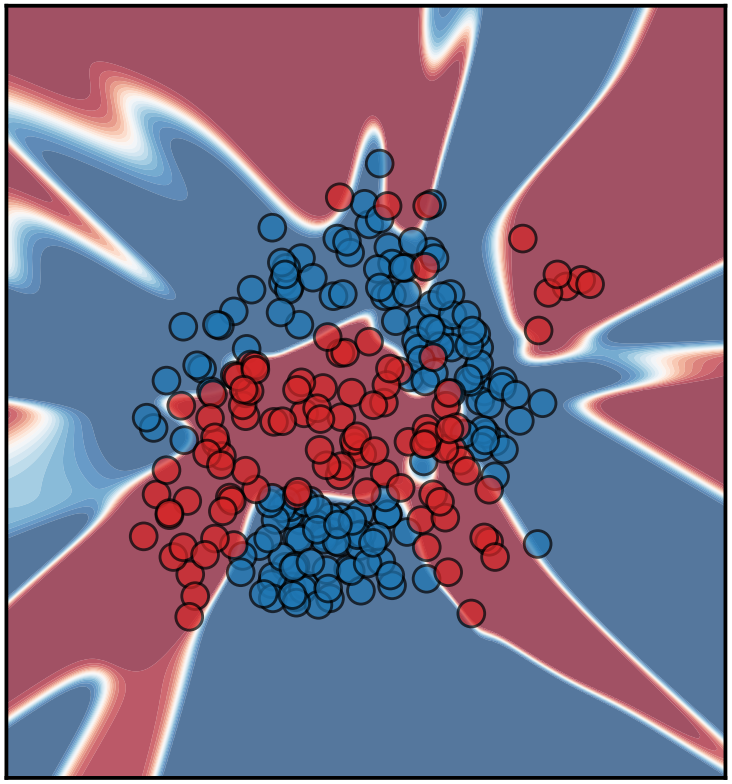




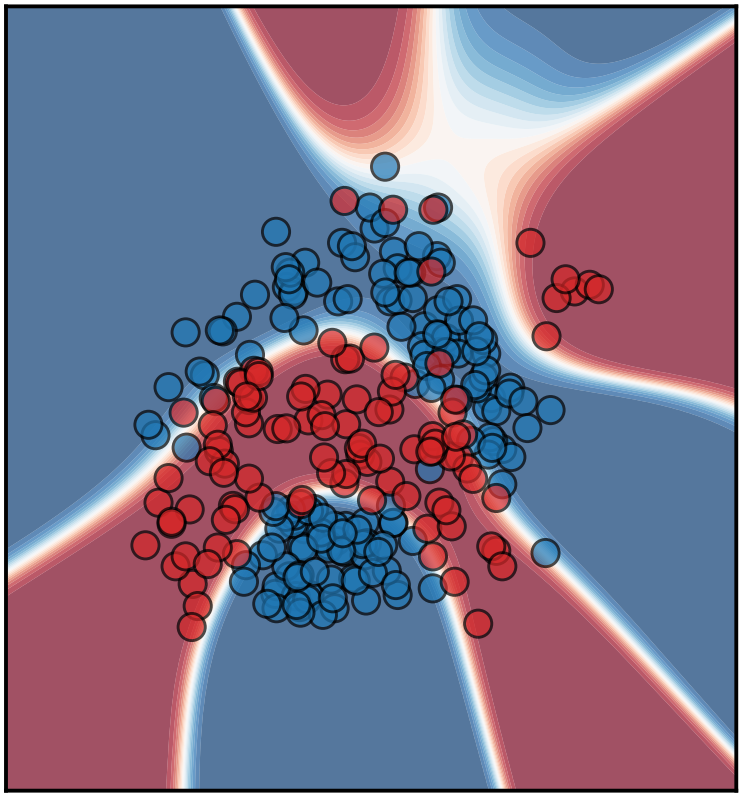
Kendall, Alex, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics." *CVPR*. 2018.

Reduce Overfitting

Standard DL



Bayesian DL

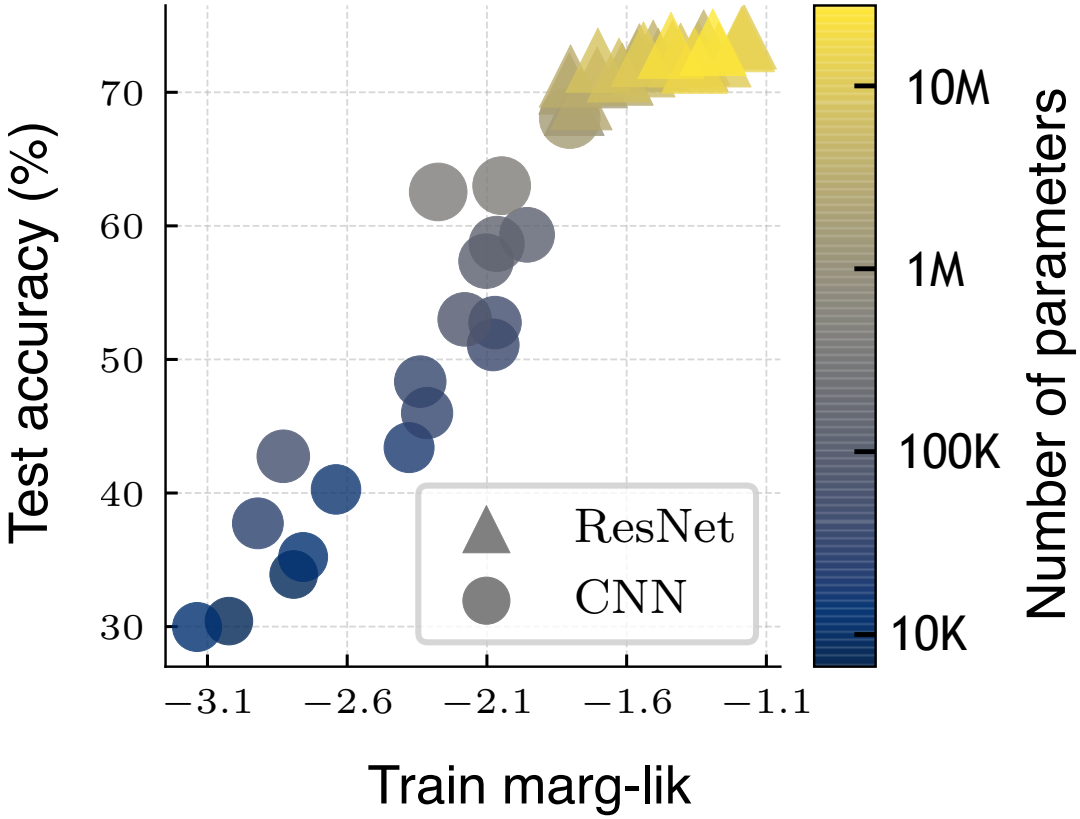


Left figure is cross-validation. Right figure is “Marginal Likelihood”.

Immer et al., Scalable Marginal Likelihood Estimation for Model Selection in Deep Learning, *ICML*, 2021.

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.

Immer et al., Scalable Marginal Likelihood Estimation for Model Selection in Deep Learning, *ICML*, 2021.

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?	✓	✗

ML vs Bayes

Objectives and solutions

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.

Khan and Rue, The Bayesian Learning Rule, arXiv, https://arxiv.org/abs/2107.04562, 2021

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.

Khan and Rue, The Bayesian Learning Rule, arXiv, https://arxiv.org/abs/2107.04562, 2021

See Appendix A in Khan and Rue, 2021

Bayes Rule as Optimization

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

See Appendix A in Khan and Rue, 2021

Bayes Rule as Optimization

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

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

See Appendix A in Khan and Rue, 2021

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

See Appendix A in Khan and Rue, 2021

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

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

See Appendix A in Khan and Rue, 2021

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

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

See Appendix A in Khan and Rue, 2021

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

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

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

See Appendix A in Khan and Rue, 2021

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

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

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

See Appendix A in Khan and Rue, 2021

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

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

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

See Appendix A in Khan and Rue, 2021

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

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

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

Holds for any loss function (generalized-posterior)

Zellner (1988), Bissiri, et al. (2016), Shawe-Taylor and Williamson (1997), Cesa-Bianchi and Lugosi (2006)

See Section 1.2, Eq 2 in Khan and Rue, 2021

ML vs Bayes Objective

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

Entropy

Generalized-Posterior approximation

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

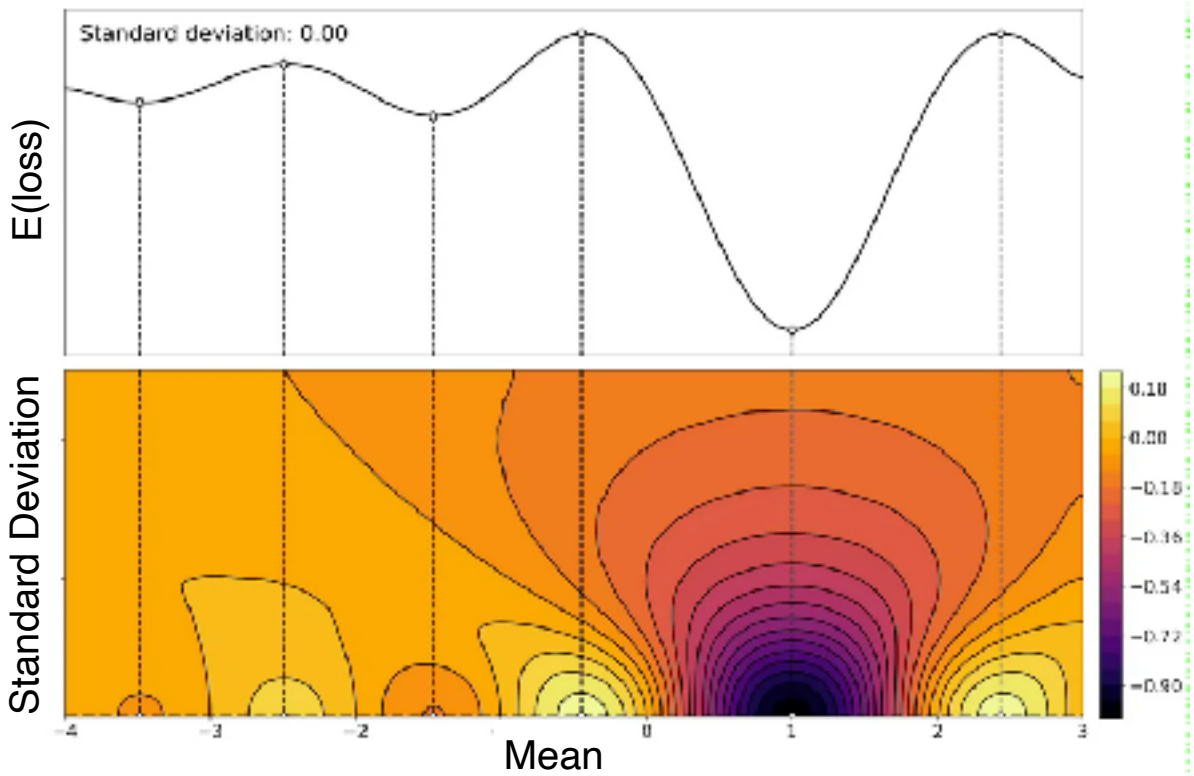
See Section 1.2, Eq 2 in Khan and Rue, 2021

Bayes Objective

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

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).

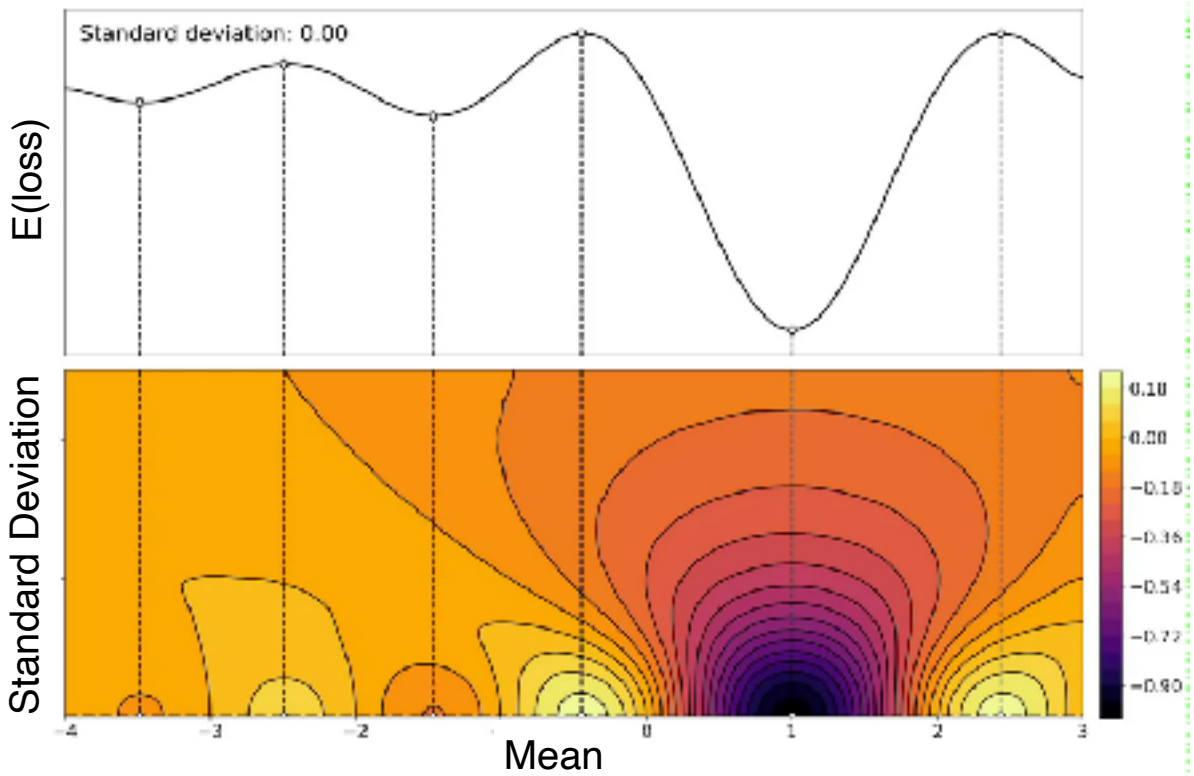
See Section 1.2, Eq 2 in Khan and Rue, 2021

Bayes Objective

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

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).

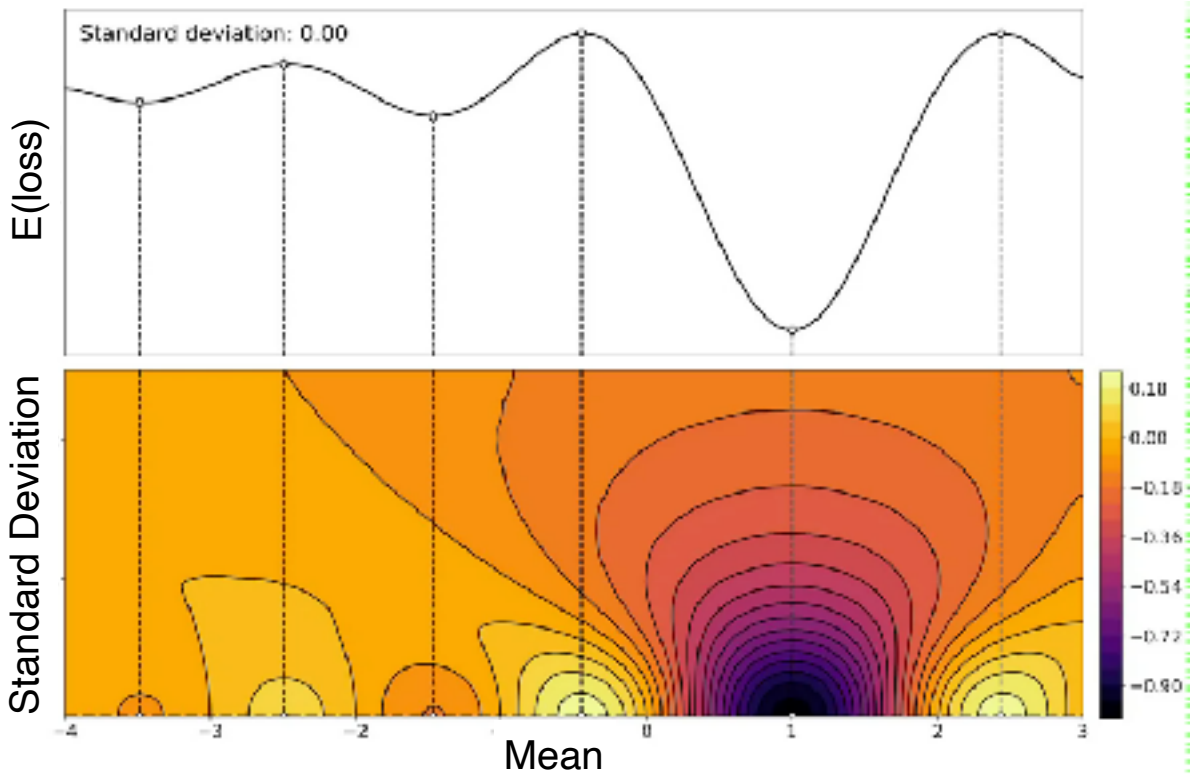
See Section 1.2, Eq 2 in Khan and Rue, 2021

Bayes Objective

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

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).

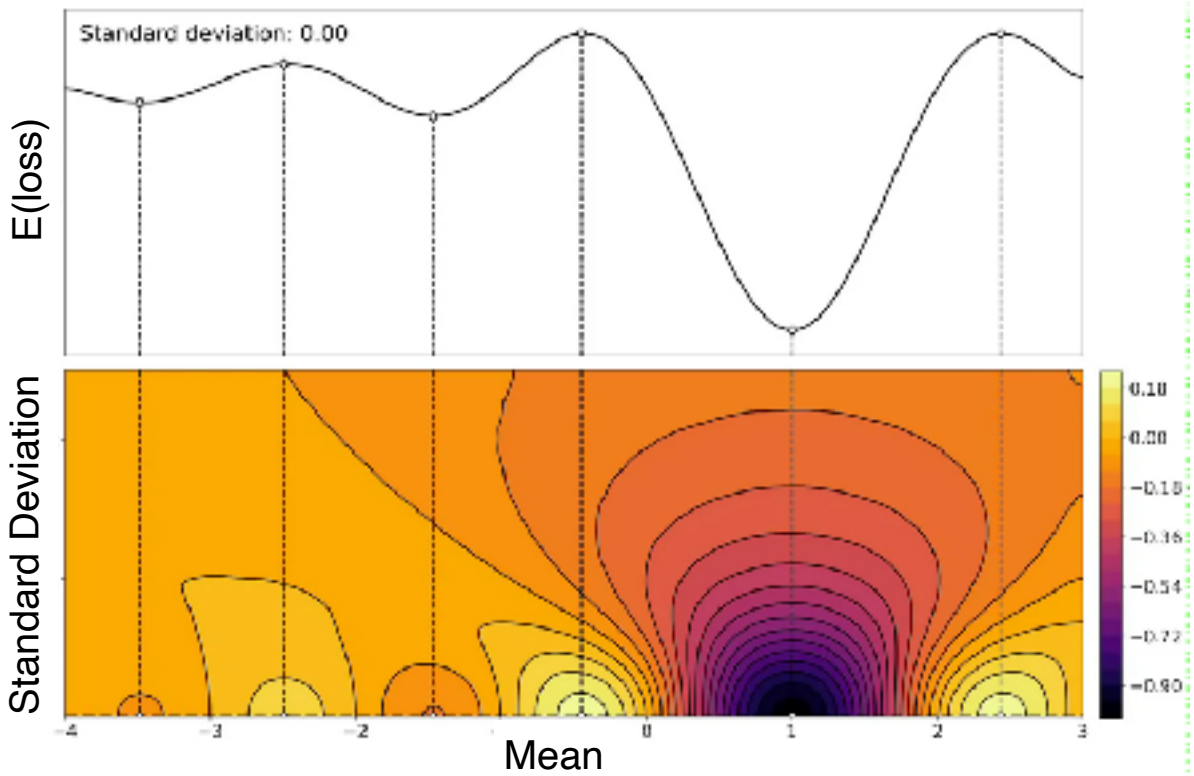
See Section 1.2, Eq 2 in Khan and Rue, 2021

Bayes Objective

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

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).

See Section 1.2 in Khan and Rue, 2021

Exponential Family Approximations

Natural parameters

Sufficient Statistics

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

See Section 1.2 in Khan and Rue, 2021

Exponential Family Approximations

Natural parameters

Sufficient Statistics

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

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

See Section 1.2 in Khan and Rue, 2021

Exponential Family Approximations

Natural parameters

Sufficient Statistics

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

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

$$\propto \exp\left[(Sm)^\top \theta + \mathrm{Tr}\left(-\frac{S}{2}\theta\theta^\top\right)\right]$$

See Section 1.2 in Khan and Rue, 2021

Exponential Family Approximations

Natural parameters

Sufficient Statistics

Expectation parameters

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

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

$$\propto \exp\left[(Sm)^\top \theta + \mathrm{Tr}\left(-\frac{S}{2}\theta\theta^\top\right)\right]$$

Exponential Family Approximations

Natural parameters

Sufficient Statistics

Expectation parameters

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

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

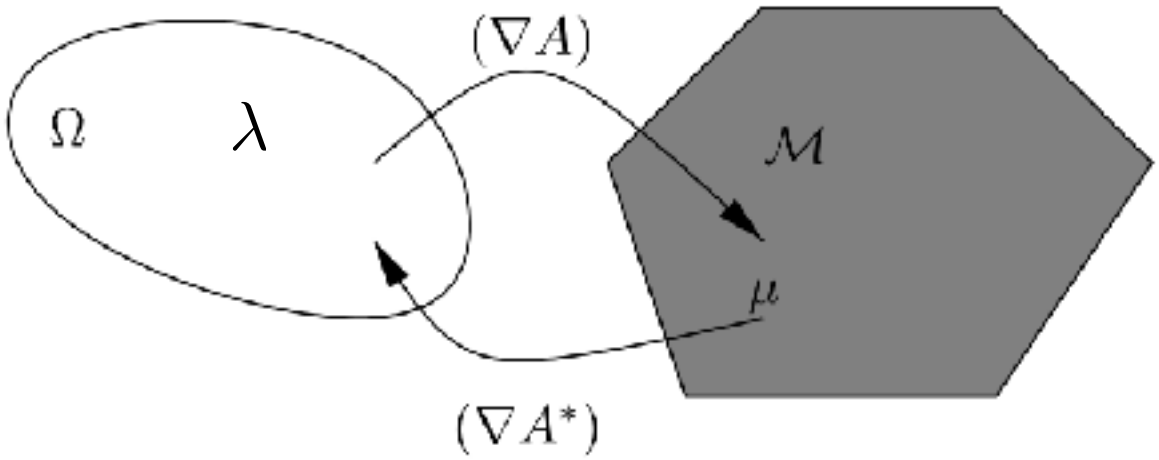
$$\propto \exp\left[(Sm)^\top \theta + \mathrm{Tr}\left(-\frac{S}{2}\theta\theta^\top\right)\right]$$

Gaussian distribution	$q(\theta) := \mathcal{N}(\theta\|m, S^{-1})$
Natural parameters	$\lambda := \{Sm, -S/2\}$
Expectation parameters	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

See Eq. 4 and Section 3 in Khan and Rue, 2021

Duality of Exponential-Family

Natural parameters



Natural gradient wrt λ = Gradient wrt μ [1]

$$F(\lambda)^{-1}\nabla_\lambda = \nabla_\mu$$

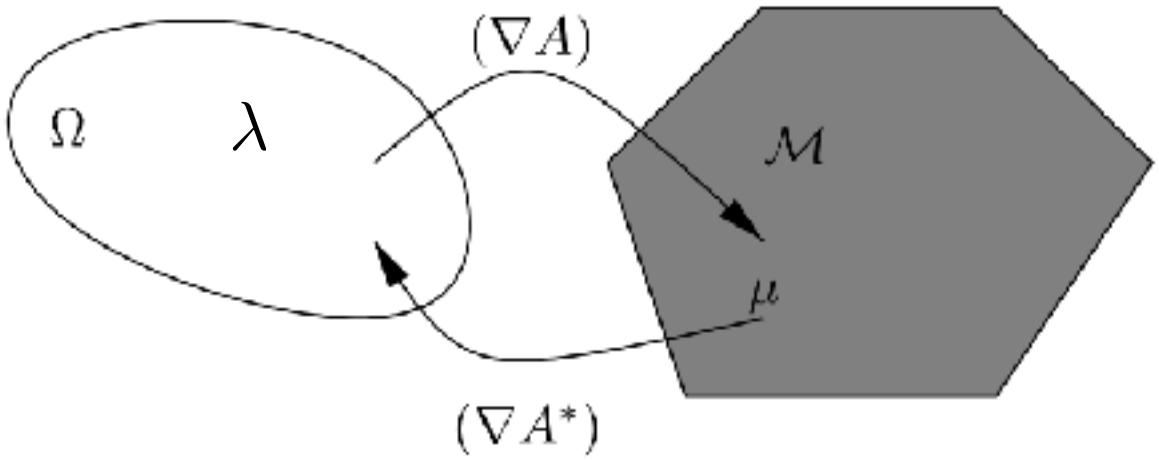
FIM

See Eq. 4 and Section 3 in Khan and Rue, 2021

Duality of Exponential-Family

Natural parameters

Expectation parameters



Natural gradient wrt λ = Gradient wrt μ [1]

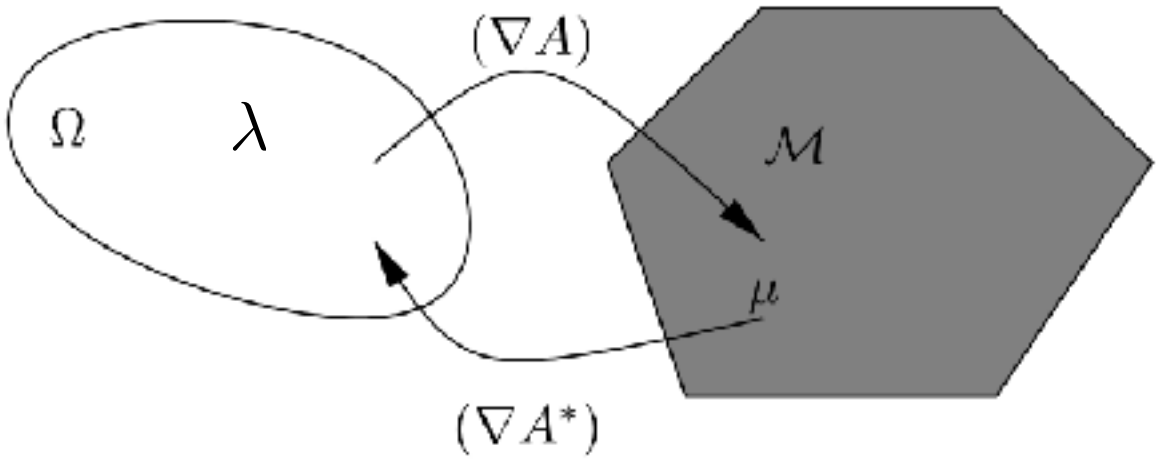
$$F(\lambda)^{-1}\nabla_\lambda = \nabla_\mu$$

FIM

Duality of Exponential-Family

Natural parameters

Expectation parameters



Natural gradient wrt λ = Gradient wrt μ [1]

$$F(\lambda)^{-1}\nabla_\lambda = \nabla_\mu$$

FIM

1. Khan and Nielsen, Fast yet simple natural-gradient descent for variational inference… *ISITA* (2018).

See Eq. 5 and the discussion afterward, in Khan and Rue, 2021

Solutions of Bayes Objective

A fundamental equation

Optimal approx — Natural gradient

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

Entropy

See Eq. 5 and the discussion afterward, in Khan and Rue, 2021

Solutions of Bayes Objective

A fundamental equation

Optimal approx — Natural gradient

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

Entropy

For minimal Exp-Family

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

See Eq. 5 and the discussion afterward, in Khan and Rue, 2021

Solutions of Bayes Objective

A fundamental equation

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

Optimal approx (Entropy) — Natural gradient

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

See Eq. 5 and the discussion afterward, in Khan and Rue, 2021

Solutions of Bayes Objective

Optimal approx Natural gradient

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

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”

Based on Sec 1.3.3 in Khan and Rue, 2021

A simple example

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)

Related Frormulations

- 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.

ML algorithms from Bayesian Principles

Unify, generalize, and improve learning-algorithms

See Section 1.2 and 2 in Khan and Rue, 2021

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

Khan and Lin. "Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models." AIstats (2017).

See Section 1.2 and 2 in Khan and Rue, 2021

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

Natural Gradient

Natural and Expectation parameters of an exponential family distribution q

(natural-gradient descent & mirror descent)

Khan and Lin. "Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models." AIstats (2017).

See Section 1.2 and 2 in Khan and Rue, 2021

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

Natural Gradient

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

By changing *Q*, we can recover DL algorithms (and more)

Khan and Lin. "Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models." AIstats (2017).

See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

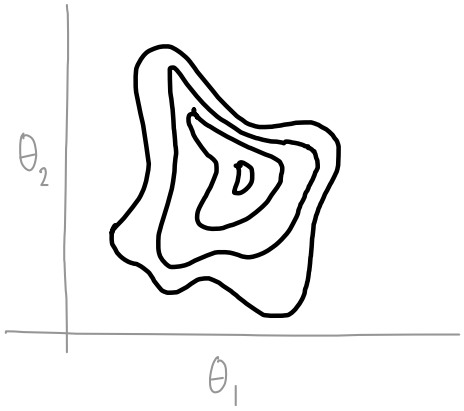
Complex  Simple

See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple

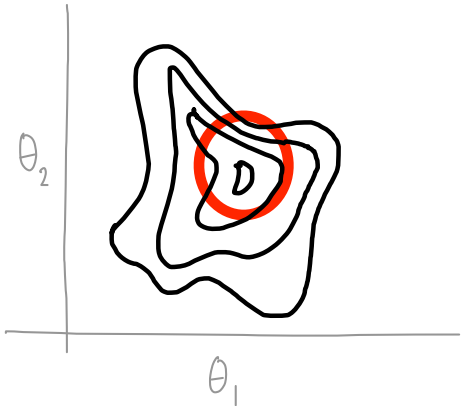


See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple

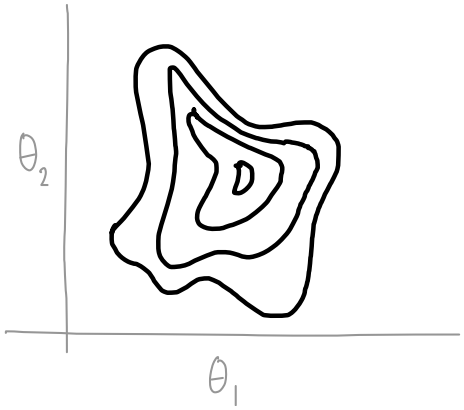


See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple

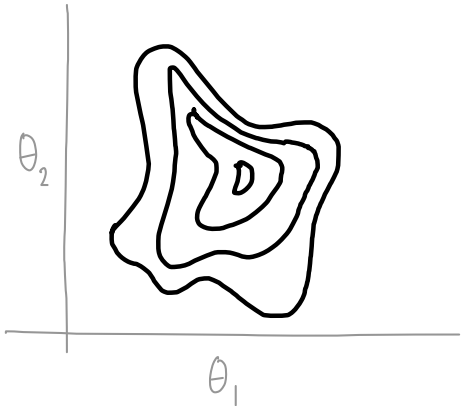


See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple



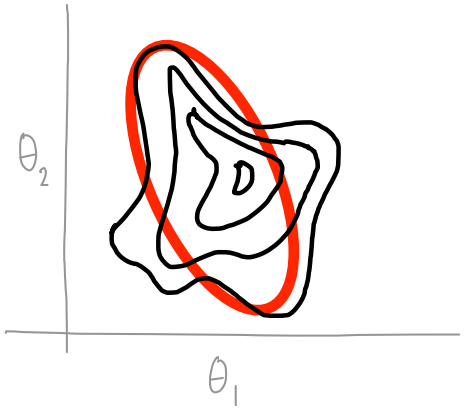
Gradient Descent

See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple



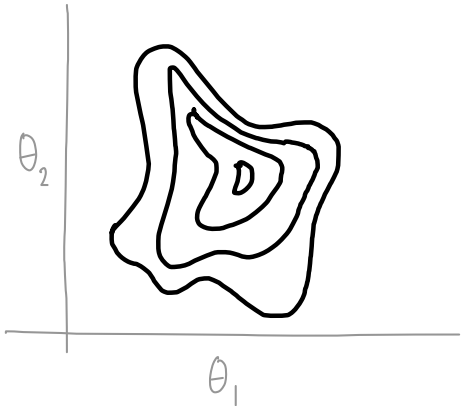
Gradient Descent

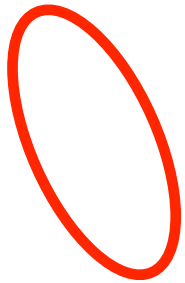
See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple





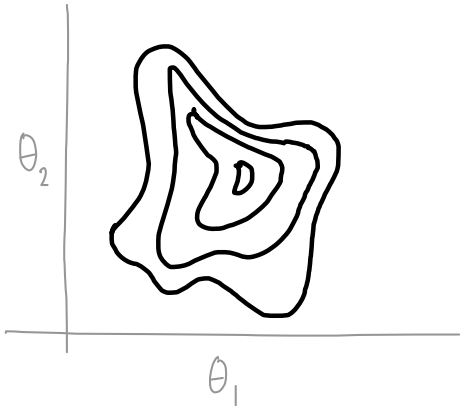
Gradient Descent

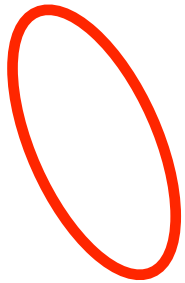
See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple





Newton

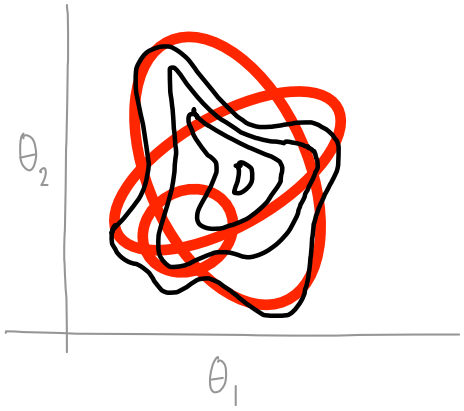
Gradient Descent

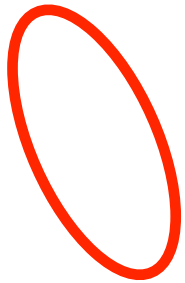
See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple





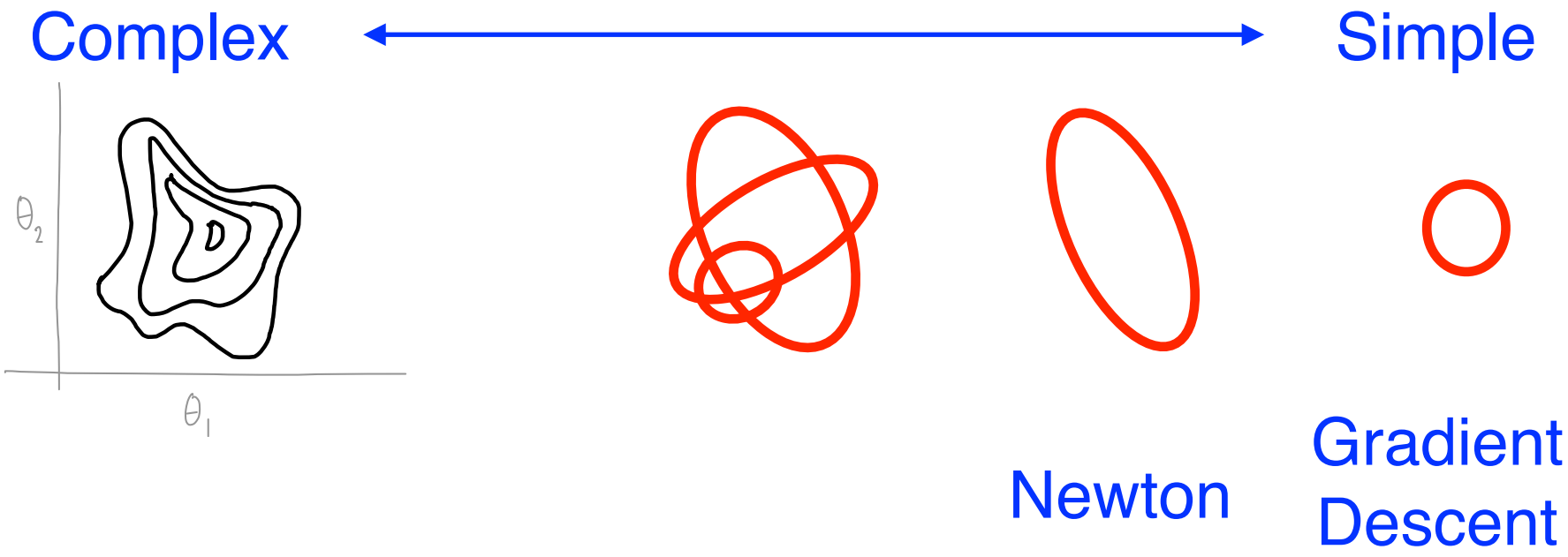
Newton

Gradient Descent

See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

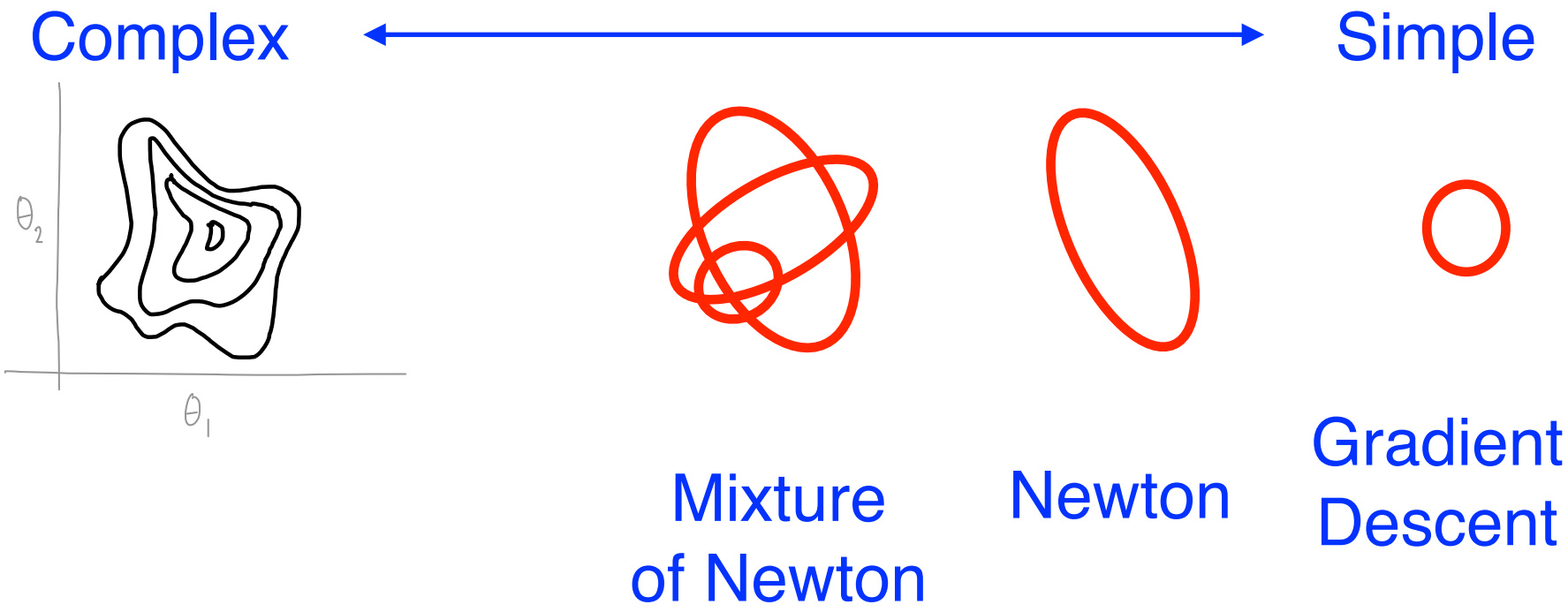
Posterior Approximation $\longleftrightarrow$ Learning-Algorithm



See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

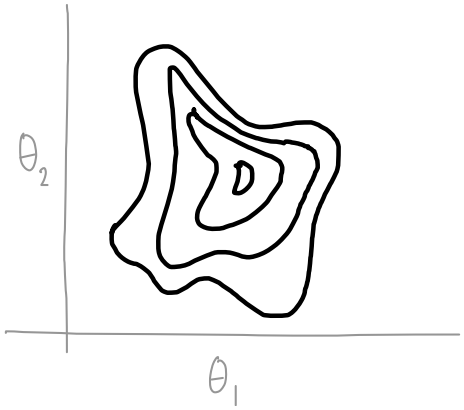


See Section 1.3 and 3.2 in Khan and Rue, 2021

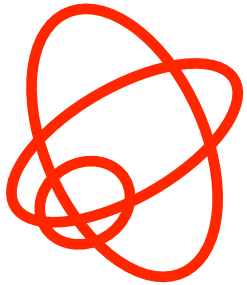
Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

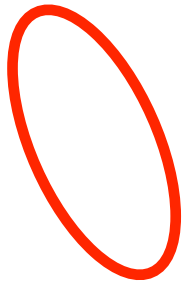
Complex $\longleftarrow$ Simple



Bayes' rule



Mixture of Newton



Newton

Gradient Descent

See Section 1.3.1 in Khan and Rue, 2021

Gradient Descent from Bayes

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

See Section 1.3.1 in Khan and Rue, 2021

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

See Section 1.3.1 in Khan and Rue, 2021

Gradient Descent from Bayes

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

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

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

See Section 1.3.1 in Khan and Rue, 2021

Gradient Descent from Bayes

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

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

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

See Section 1.3.1 in Khan and Rue, 2021

Gradient Descent from Bayes

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

"Global" to "local" (the delta method)
$\mathbb{E}_q[\ell(\theta)] \approx \ell(m)$

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

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

See Section 1.3.1 in Khan and Rue, 2021

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

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

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

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

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

We can compute uncertainty using a variant of Adam.

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).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

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

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

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

Derived by choosing a multivariate Gaussian

Gaussian distribution	$q(\theta) := \mathcal{N}(\theta \vert m, S^{-1})$
Natural parameters	$\lambda := \{Sm, -S/2\}$
Expectation parameters	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

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

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

Derived by choosing a multivariate Gaussian

Gaussian distribution	$q(\theta) := \mathcal{N}(\theta \vert m, S^{-1})$
Natural parameters	$\lambda := \{Sm, -S/2\}$
Expectation parameters	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

Newton's Method from Bayes

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

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

Derived by choosing a multivariate Gaussian

Gaussian distribution	$q(\theta) := \mathcal{N}(\theta \vert m, S^{-1})$
Natural parameters	$\lambda := \{Sm, -S/2\}$
Expectation parameters	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

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

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

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

Derived by choosing a multivariate Gaussian

Gaussian distribution	$q(\theta) := \mathcal{N}(\theta \vert m, S^{-1})$
Natural parameters	$\lambda := \{Sm, -S/2\}$
Expectation parameters	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

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

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

Derived by choosing a multivariate Gaussian

Gaussian distribution	$q(\theta) := \mathcal{N}(\theta \mid m, S^{-1})$
Natural parameters	$\lambda := \{Sm, -S/2\}$
Expectation parameters	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

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

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

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

Derived by choosing a multivariate Gaussian

Gaussian distribution	$q(\theta) := \mathcal{N}(\theta \mid m, S^{-1})$
Natural parameters	$\lambda := \{Sm, -S/2\}$
Expectation parameters	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

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

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

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

Derived by choosing a multivariate Gaussian

Gaussian distribution	$q(\theta) := \mathcal{N}(\theta \mid m, S^{-1})$
Natural parameters	$\lambda := \{Sm, -S/2\}$
Expectation parameters	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

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

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

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

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

Derived by choosing a multivariate Gaussian

Gaussian distribution	$q(\theta) := \mathcal{N}(\theta \mid m, S^{-1})$
Natural parameters	$\lambda := \{Sm, -S/2\}$
Expectation parameters	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

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

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

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

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

Express in terms of gradient and Hessian of loss:

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

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

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

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

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

Delta Method

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

Express in terms of gradient and Hessian of loss:

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

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

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

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

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

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

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

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

Express in terms of gradient and Hessian of loss:

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

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

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

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

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

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

Delta Method

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

Express in terms of gradient and Hessian of loss:

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

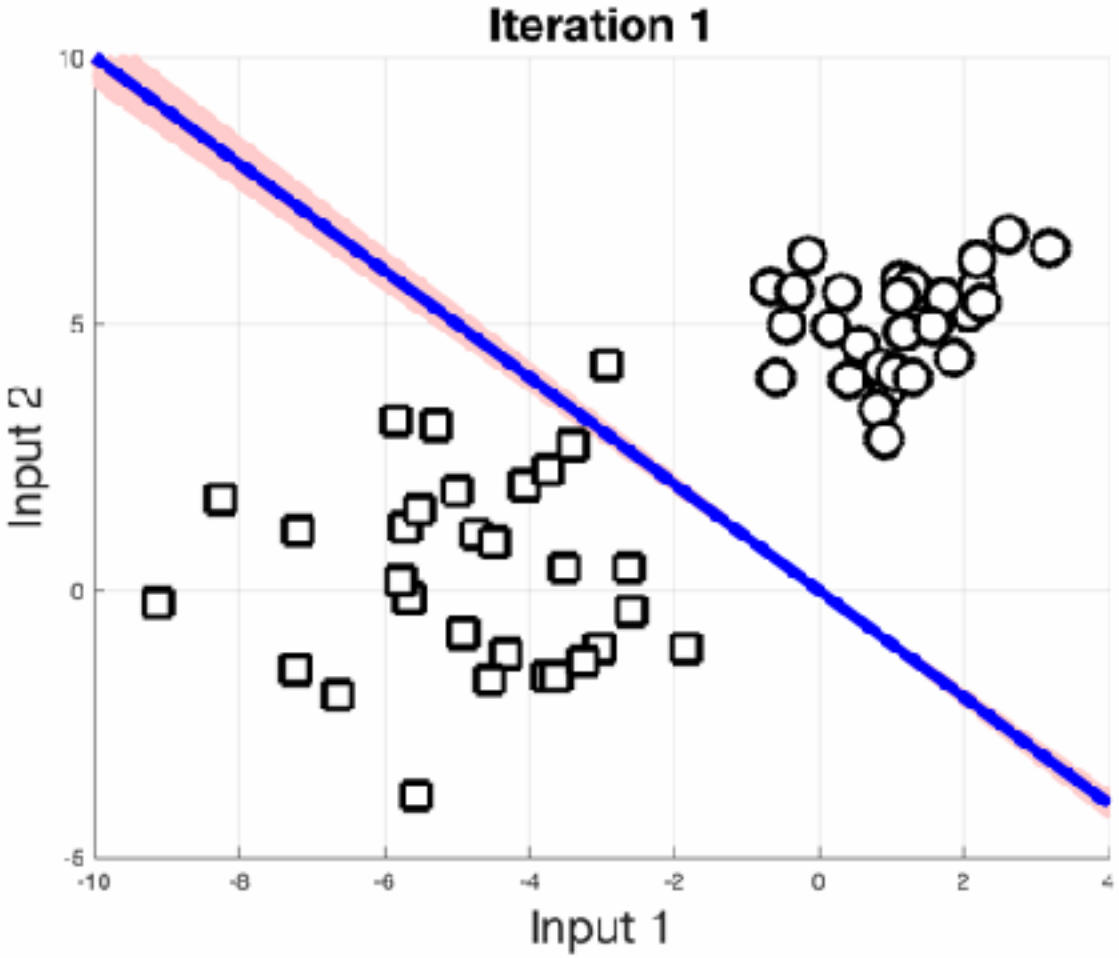
$$\nabla_{\mathbb{E}_q(\theta\theta^\top)} \mathbb{E}_q[\ell(\theta)] = \mathbb{E}_q[H_\theta]$$

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

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

Uncertainty Estimation

Uncertainty in Logistic Regression



Variational Online Newton method

Frequentist (Adam)

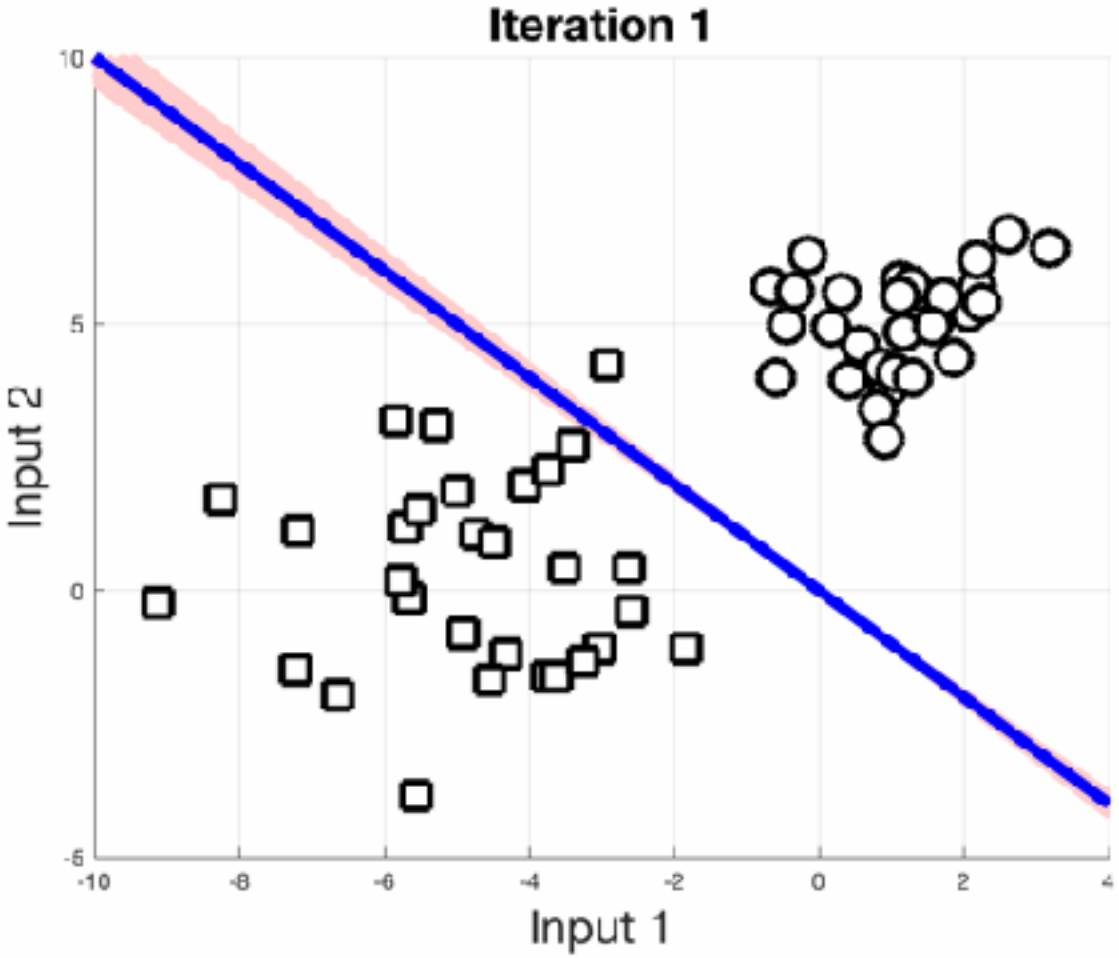
Bayes (VON, mean)

Bayes (samples)

Logistic Regression
Minibatch = 5,
Learning rates = (0.01, 0.01)

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

Uncertainty in Logistic Regression



Variational Online Newton method

Frequentist (Adam)

Bayes (VON, mean)

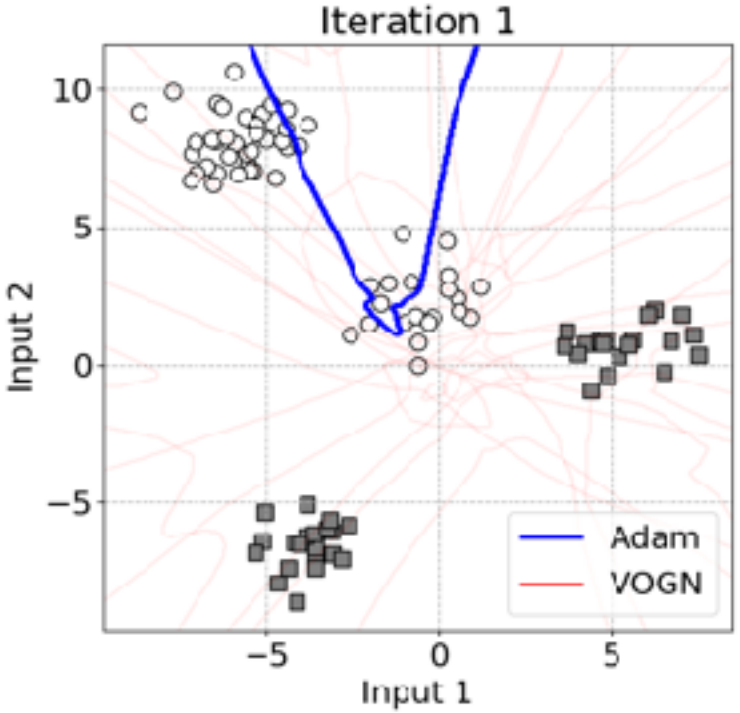
Bayes (samples)

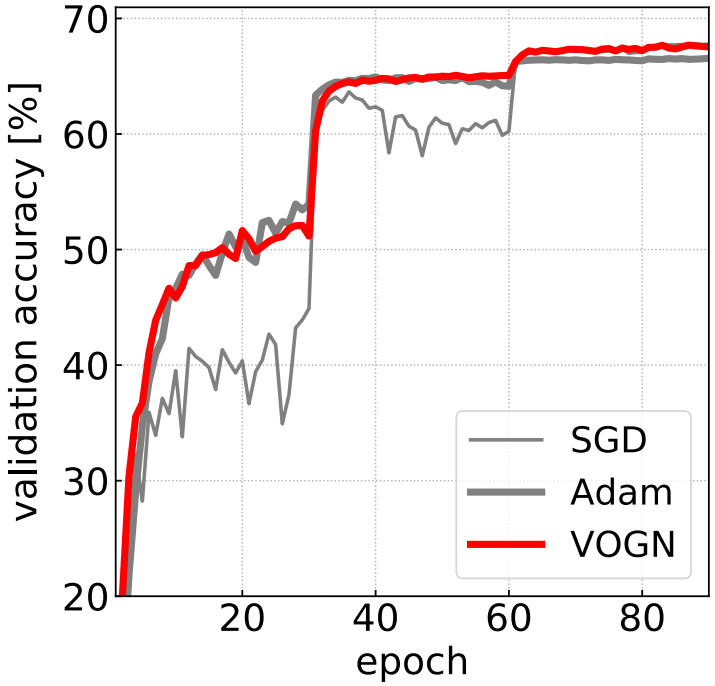
Logistic Regression
Minibatch = 5,
Learning rates = (0.01, 0.01)

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

Uncertainty in Deep Nets

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



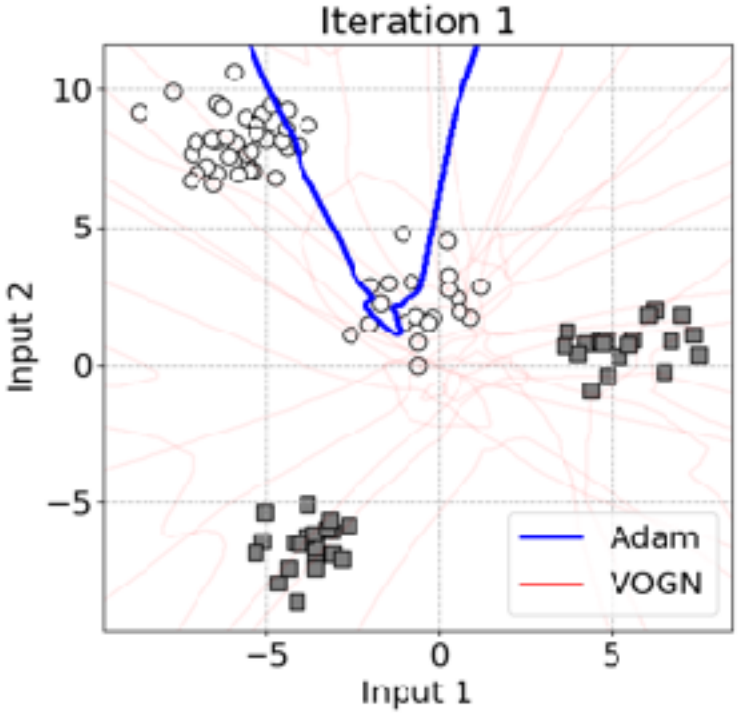


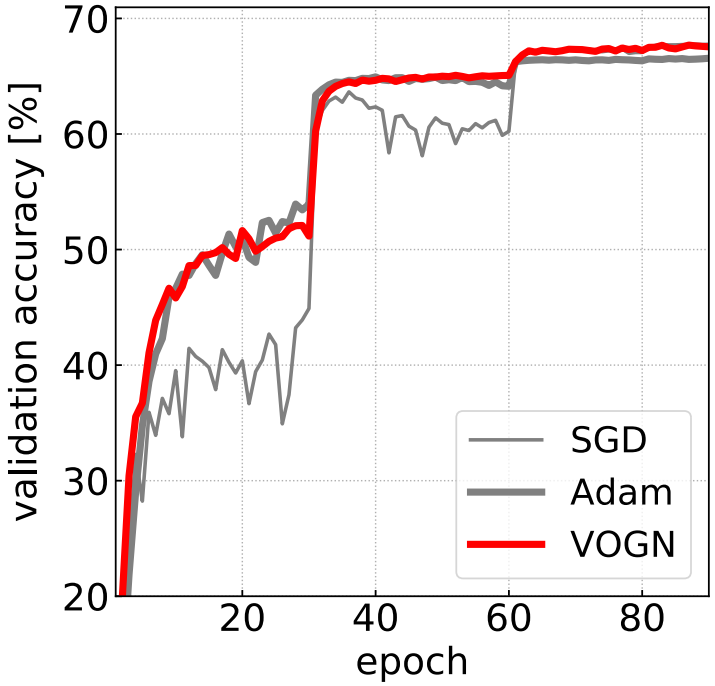
Code available at https://github.com/team-approx-bayes/dl-with-bayes

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 in 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).

See Section 4.2 in Khan and Rue, 2021

RMSprop/Adam from Bayes

RMSprop

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

BLR for Gaussian approx

$$S \leftarrow (1-\rho)S + \rho(H_\theta)$$
$$m \leftarrow m - \alpha S^{-1}\nabla_\theta \ell(\theta)$$

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 4.2 in Khan and Rue, 2021

RMSprop/Adam from Bayes

RMSprop

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

BLR for Gaussian approx

$$S \leftarrow (1-\rho)S + \rho(H_\theta)$$
$$m \leftarrow m - \alpha S^{-1}\nabla_\theta \ell(\theta)$$

To get RMSprop, make the following choices

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

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 4.2 in Khan and Rue, 2021

RMSprop/Adam from Bayes

RMSprop

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

BLR for Gaussian approx

$$S \leftarrow (1-\rho)S + \rho(H_\theta)$$
$$m \leftarrow m - \alpha S^{-1}\nabla_\theta \ell(\theta)$$

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])

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

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

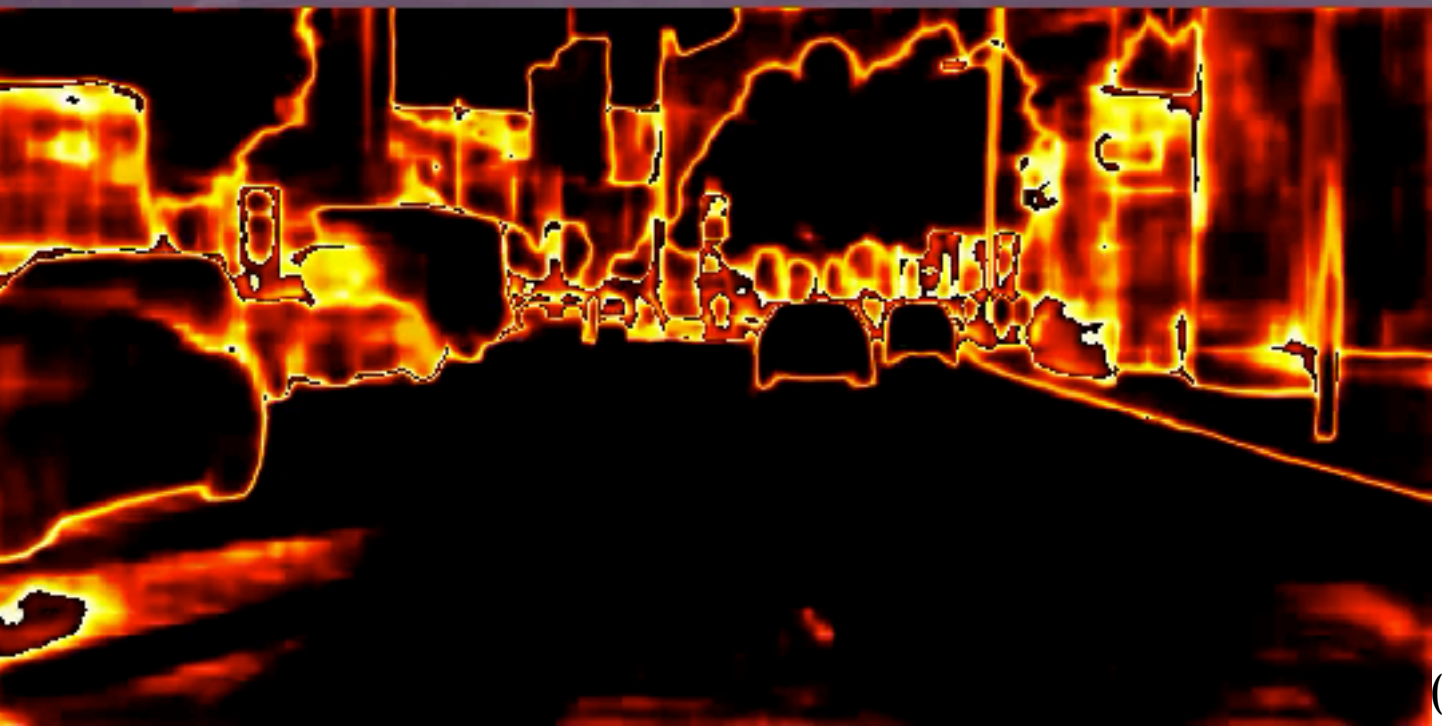
$$g \leftarrow \hat{\nabla}\ell(\theta), \text{ where } \theta \sim \mathcal{N}(m, \sigma^2)$$
$$s \leftarrow (1-\rho)s + \rho(\Sigma_i g_i^2)$$
$$m \leftarrow m - \alpha(s+\gamma)^{-1}\nabla_\theta \ell(\theta)$$
$$\sigma^2 \leftarrow (s+\gamma)^{-1}$$

```
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).

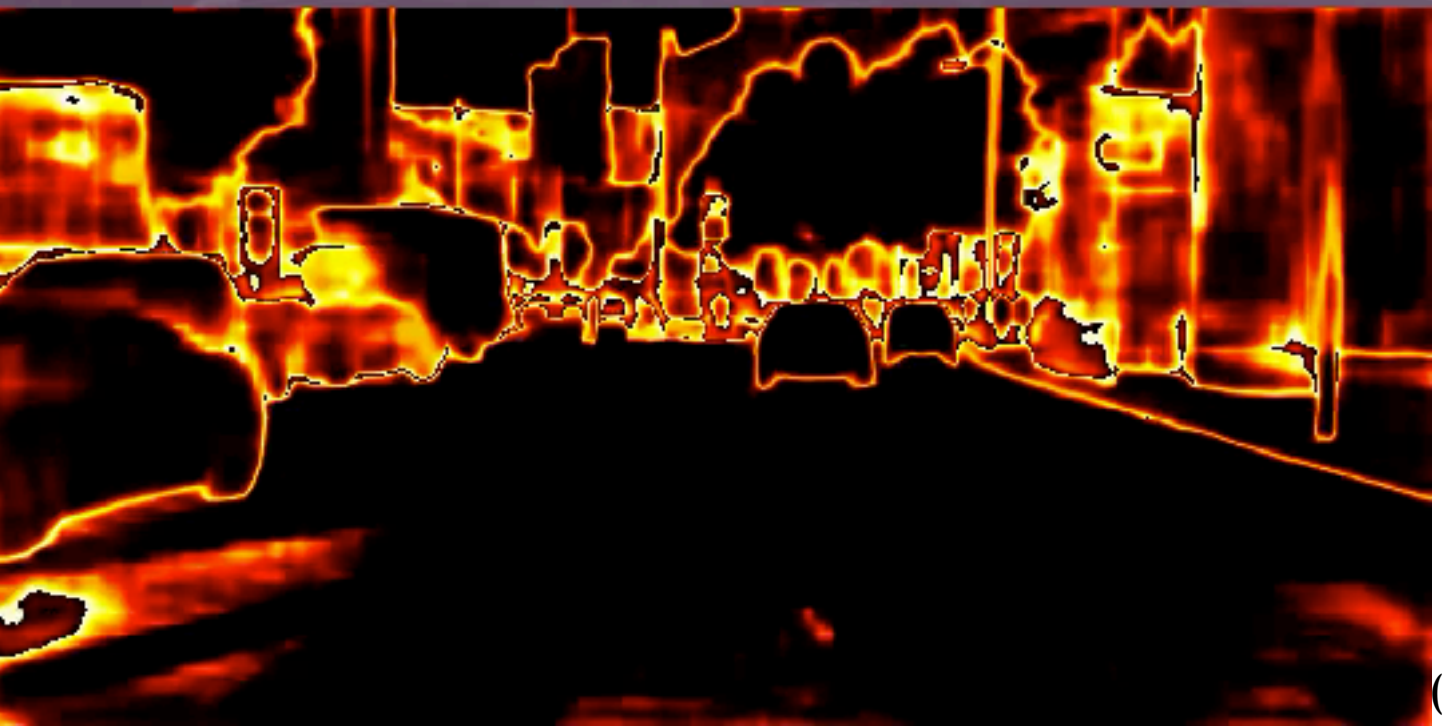
Image Segmentation



Uncertainty (entropy of class probs)

(By Roman Bachmann)

Image Segmentation



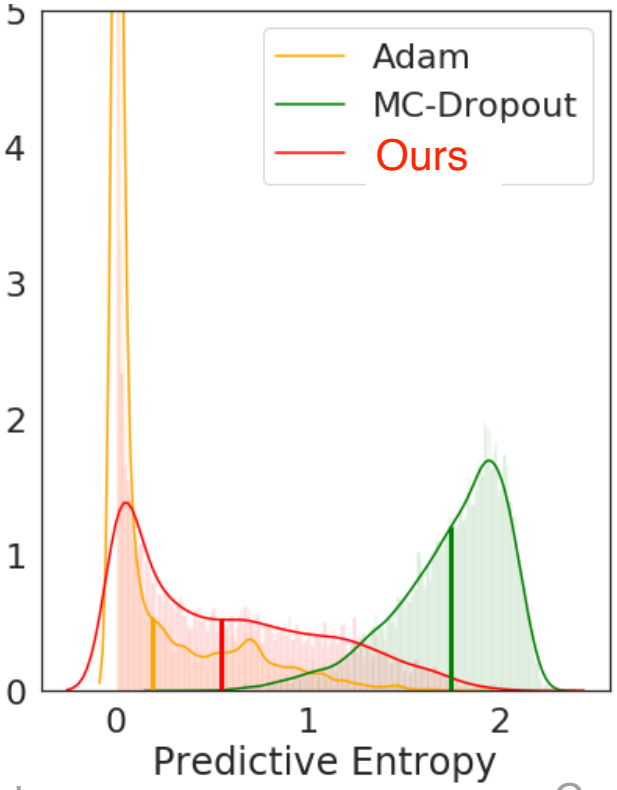
Uncertainty (entropy of class probs)

(By Roman Bachmann)

Out-of-Distributions Test

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

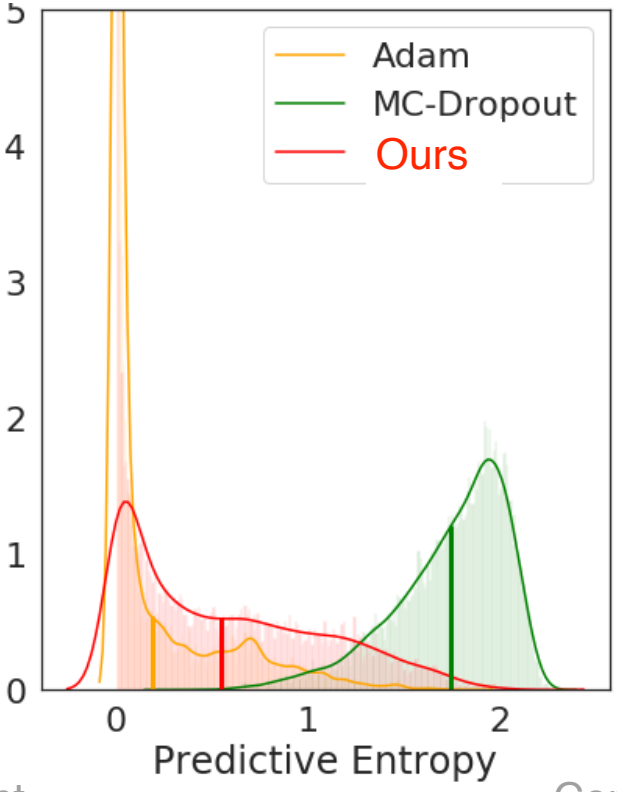
In-Distribution (CIFAR-10)



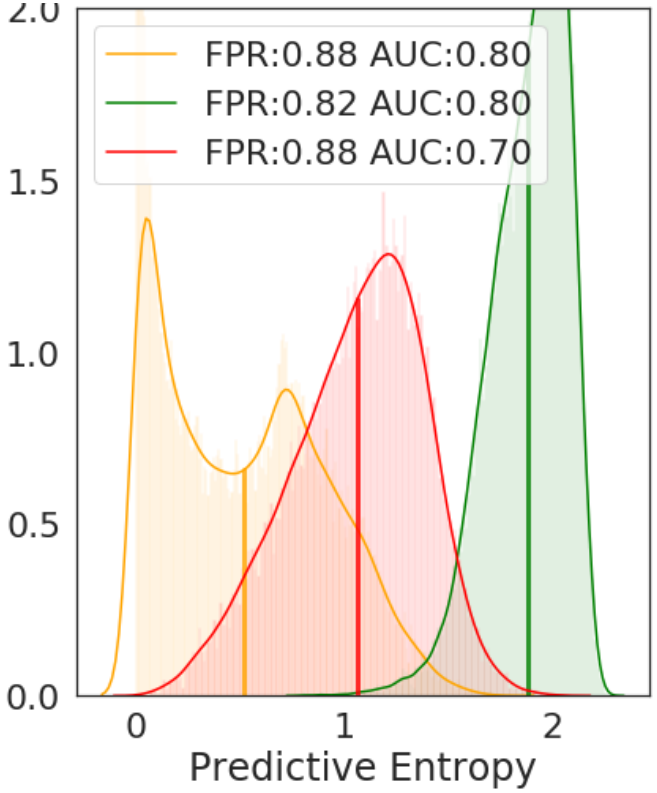
Out-of-Distributions Test

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)



More Confident

Less Confident

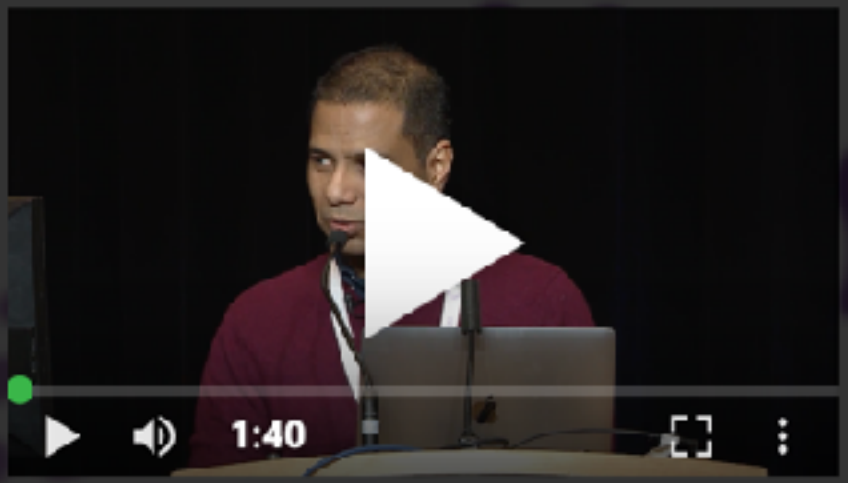
Tuning VOGN

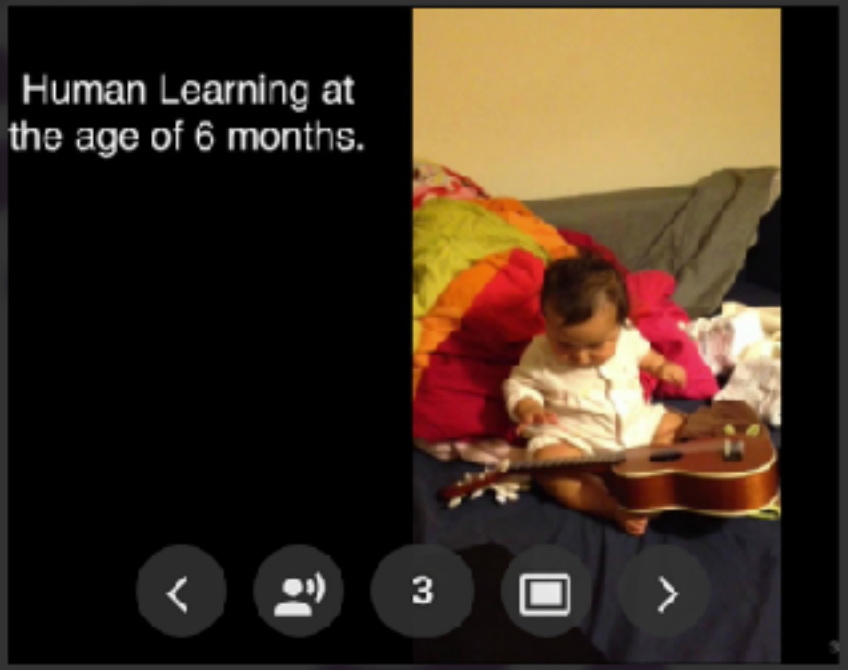
The trick is to mimic Adam's trajectory as closely as possible

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 $\mathbf{s}_t$. This increases the lower bound of the eigenvalues of the diagonal covariance $\mathbf{\Sigma}_t$ 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.

Sec 3, last paragraph in Osawa et al. "Practical Deep Learning with Bayesian Principles." NeurIPS (2019).

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

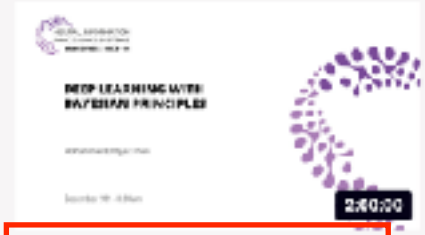
by Yoshua Bengio

17,953 views · Dec 11, 2019

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

by Aaron Hertzmann, Adam Roberts, ...

9,656 views · Dec 14, 2019



Deep Learning with Bayesian Principles

by Mohammad Emtiyaz Khan

8,084 views · Dec 9, 2019

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

by Vivienne Sze

7,163 views · Dec 9, 2019

Approximate Bayesian Inference Methods

Variational Inference, Laplace Approximation, Black-box VI, Bayesian Deep Learning methods

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.

Laplace Approximation

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

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

H_m: Hessian at m

Laplace Approximation

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

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

H_m ← 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

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

H_m ← Hessian at m

Bayesian principles we discussed are general principles to derive learning algorithms

Calling them variational inference limits their scope!

(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]

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

  - 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).

Black-Box VI & Bayesian Learning rule

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

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

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

1. Ranganath, Rajesh, Sean Gerrish, and David Blei. "Black box variational inference." *Artificial Intelligence and Statistics*. 2014.

Bayesian Learning Rule and Related Works

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

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

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

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

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

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 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).

Old and New work on Natural-Gradient VI

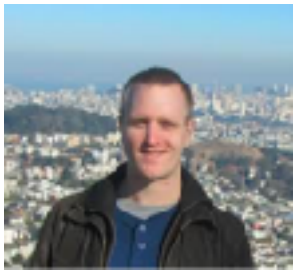
References for Step C: Natural-Gradient VI

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." AIstats (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).

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." AIstats (2017).
  2. Khan and Nielsen. "Fast yet simple natural-gradient descent for variational inference in complex models." (*2018) ISITA*.
- Mixture of Exponential family
  3. Lin et al. "Fast and Simple Natural-Gradient Variational Inference with Mixture of Exponential-family Approximations," ICML (2019).
- 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)
- Gaussian approx <=> Newton-variants

Wu Lin (UBC)

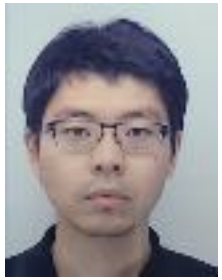


Mark Schmidt (UBC)

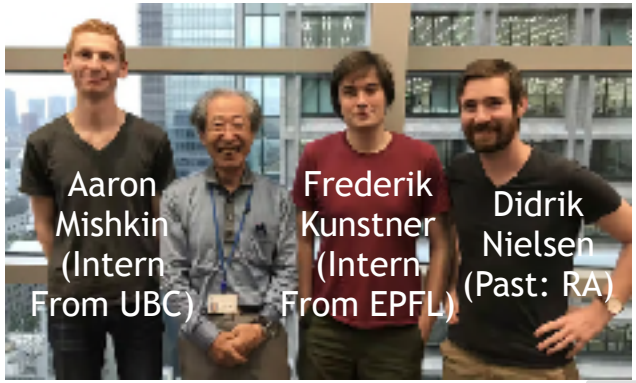
Frank Nielsen (Sony)

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).

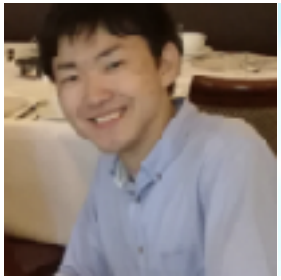


Voot Tangkaratt
(Postdoc, RIKEN-AIP)



Yarin Gal
(UOxford)

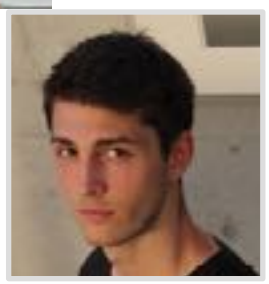
Akash Srivastava
(UEdinburgh)



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)

1. Hensman et al. “Gaussian Process for Big Data”, UAI (2013)

How to design AI that learn like us?

How to design AI that learn like us?

- Three questions

How to design AI that learn like us?

- Three questions
  - Q1: What do we know? (model)

How to design AI that learn like us?

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

How to design AI that learn like us?

- 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?

- 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?

- 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?

- 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?

- 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.

Duality

https://tenor.com/view/clockwork-gears-brain-gif-16784329

Duality

https://tenor.com/view/clockwork-gears-brain-gif-16784329

The webpage is available at https://bayesduality.github.io/, and Twitter account @BayesDuality

The Bayes-Duality Project

Toward AI that learns adaptively, robustly, and continuously, like humans

About People Research Publications

Approx-Bayes team

Emtiyaz Khan

Research director
(Japan side)

Stat-Theory team

Julyan Arbel

Research director
(France side)

Math-Science team

Kenichi Bannai

Co-PI (Japan side)

HPC team

Rio Yokota

Co-PI
(Japan side)

Approximate Bayesian Inference Team

Emtiyaz Khan
Team Leader

Pierre Alquier
Research Scientist

Gian Maria Marconi
Postdoc

Thomas Möllenhoff
Postdoc

Lu Xu
Postdoc

Wu Lin
PhD Student
University of British Columbia

Ted Tinker
PhD Student
Okinawa Institute of Science and Technology

Peter Nickl
Research Assistant

Happy Buzaaba
Part-time Student
University of Tsukuba

Siddharth Swaroop
Remote Collaborator
University of

Dharmesh Tailor
Remote Collaborator
University of Amsterdam

Erik Daxberger
Remote Collaborator
University of Cambridge

Alexandre Piché
Remote Collaborator
MILA

Paul Chang
Remote Collaborator
Aalto University

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

Many open positions!