# Bayesian Learning Rule

Mohammad Emtiyaz Khan
RIKEN Center for AI Project, Tokyo
http://emtiyaz.github.io

1. Summary at https://emtiyaz.github.io/papers/MLfromBayes.pdf
2. Slides at https://emtiyaz.github.io/papers/Feb22_2022_AI4Sec.pdf

# AI that learn like humans

Quickly adapt to learn new skills, throughout their lives

Human Learning at the age of 6 months.

Converged at the
age of 12 months

Transfer skills

at the age of 14 months

# Failure of AI in “dynamic” setting

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

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

# 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)

# Bayesian Learning Rule

- Bayesian principles as a general principle
  - To unify/generalize/improve learning-algorithms
  - By computing “posterior approximations”
- Bayesian Learning rule (BLR)
  - Derive many existing algorithms
  - Deep Learning (SGD, RMSprop, Adam)
  - Design new algorithms for uncertainty in DL
- ~~Dual perspective of BLR for life-long learning~~
- Impact: Everything with the same principle

# The Bayesian Learning Rule

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

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

**Abstract**

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

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

# Machine Learning from a Bayesian Perspective

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

November 8, 2021

**Abstract**

I summarize a Bayesian perspective of machine learning. We view Bayes as an optimization problem whose solutions use the information-geometry of the posterior. Using this perspective, we can show that many machine-learning methods have a (more general) Bayesian side to them. I believe this perspective to be essential for bridging the gap between 'artificial' and 'natural' learning systems.

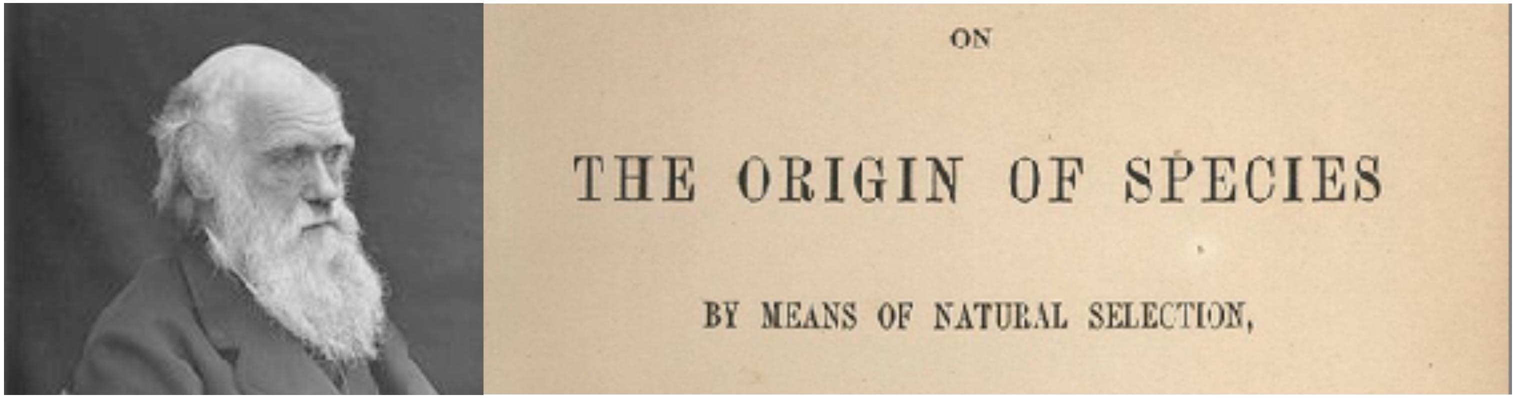
## 1 A note about the note

For now, this note deliberately lacks details. One way to read this is to use the accompanying slides. My hope is to add some equations, figures and illustration in the future. Many technical details discussed here can be found in Khan and Rue [2021]

## 2 Machine learning and Bayes

A main goal of machine-learning is to design AI systems that can learn like us. We humans, and other animals, collect experiences throughout our lives to learn and adapt. Machines currently are extremely bad at this. Majority of successful machine-learning paradigms are the ones that use 'bulk' learning in a 'static' world, where all the information is assumed to be available at once and the world stands still while we learn about it. This is far from the reality of the world we live in, and it is not surprising to see such systems fail. How can we bridge this gap between machines and living-beings? Taking a Bayesian perspective seems to be one way to go, but we argue that this is perhaps the only way forward.

See Section 6 (discussion) in Khan and Rue, 2021




# The Origin of Algorithms

A good algorithm must revise its *past* beliefs by using useful *future* information

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

# Principle of Trial-and-Error

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

$$\min_{\theta} \underbrace{\ell}_{\text{Loss}}(\underbrace{\mathcal{D}}_{\text{Data}}, \underbrace{\theta}_{\text{Model Params}}) = \sum_{i=1}^{N} [y_i - \underbrace{f_\theta}_{\text{Deep Network}}(x_i)]^2 + \gamma \theta^T \theta$$

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.

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

# Bayes Objective

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

Entropy

Generalized-Posterior approx.


1. Zellner, A. "Optimal information processing and Bayes's theorem." *The American Statistician* (1988)
2. Many other: Bissiri, et al. (2016), Shawe-Taylor and Williamson (1997), Cesa-Bianchi and Lugosi (2006)
3. Huszar's blog, Evolution Strategies, Variational Optimisation and Natural ES (2017)
4. Smith et al., On the Origin of Implicit Regularization in Stochastic Gradient Descent, ICLR, 2021

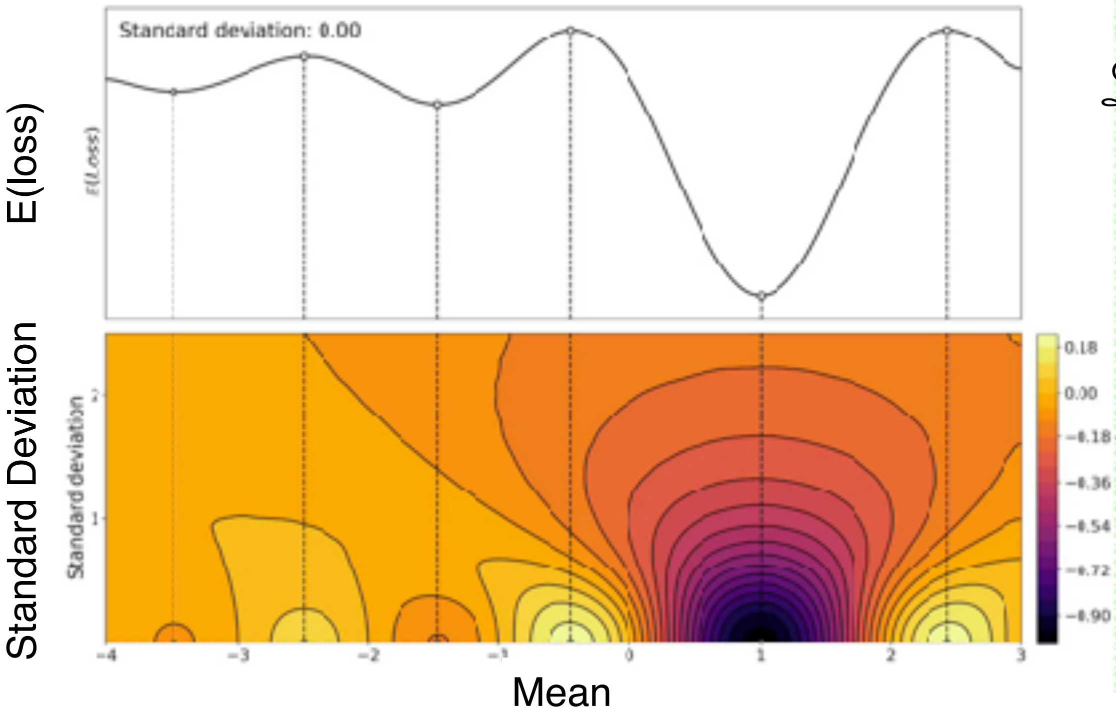
See Section 1.2, Eq 2 in Khan and Rue, 2021

# 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 approx.




$$\mathscr{L}(\mu, \sigma) = \mathbb{E}_{\mathcal{N}(\theta|\mu,\sigma^2)}[\ell(\theta)]$$

Instead of the original loss, optimize a different (smoothed) one.

A popular idea of “implicit regularization” in DL [4] now, but also A common idea in many other fields


1. Zellner, A. "Optimal information processing and Bayes's theorem." *The American Statistician* (1988)
2. Many other: Bissiri, et al. (2016), Shawe-Taylor and Williamson (1997), Cesa-Bianchi and Lugosi (2006)
3. Huszar’s blog, Evolution Strategies, Variational Optimisation and Natural ES (2017)
4. Smith et al., On the Origin of Implicit Regularization in Stochastic Gradient Descent, ICLR, 2021

# Bayesian Learning Rule

Unify, generalize, and improve
machine-learning algorithms

See Section 1.2 in Khan and Rue, 2021

# A 2-step Bayesian Scheme

Step 1: Choose an approximation (mix-exp-family)

Natural parameters / Sufficient statistics / Expectation parameters

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

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

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

| | |
|---|---|
| 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 Section 1.2 and 2 in Khan and Rue, 2021

# A 2-step Bayesian Scheme

Step 2: $\min_{\theta} \ \ell(\theta)$ vs $\min_{q \in \mathcal{Q}} \ \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$

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).

# 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 |

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

See Eq 25 in Khan and Rue, 2021 (Bonnet's theorem)

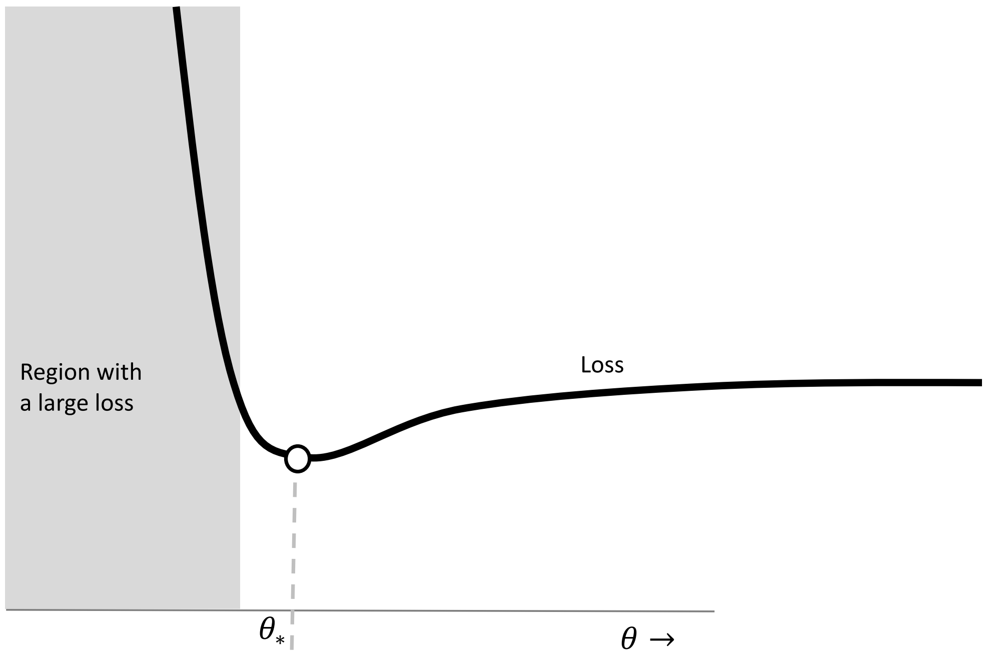
# Bayes vs Non-Bayes

GD: $\theta \leftarrow \theta - \rho \nabla_\theta \ell(\theta) \implies \nabla_\theta \ell(\theta_*) = 0$

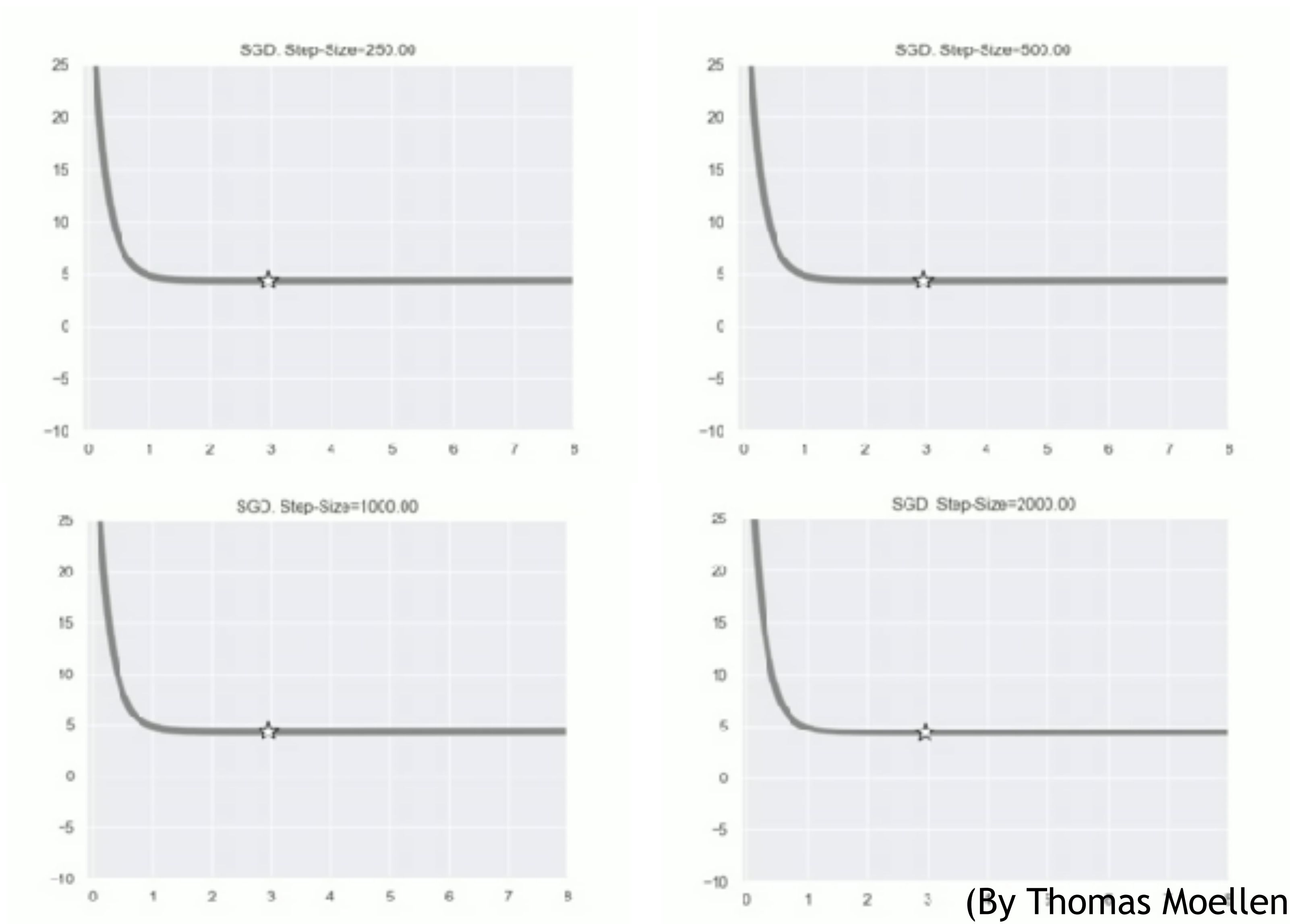
BLR: $m \leftarrow m - \rho \nabla_{m} \mathbb{E}_q[\ell(\theta)]$

$\implies \nabla_m \mathbb{E}_{q_*}[\ell(\theta)] = 0 \implies \mathbb{E}_{q_*}[\nabla_\theta \ell(\theta)] = 0$

Bayesian solution injects "noise" which has a similar regularization effect to noise in Stochastic GD. It prefers "flatter" directions.

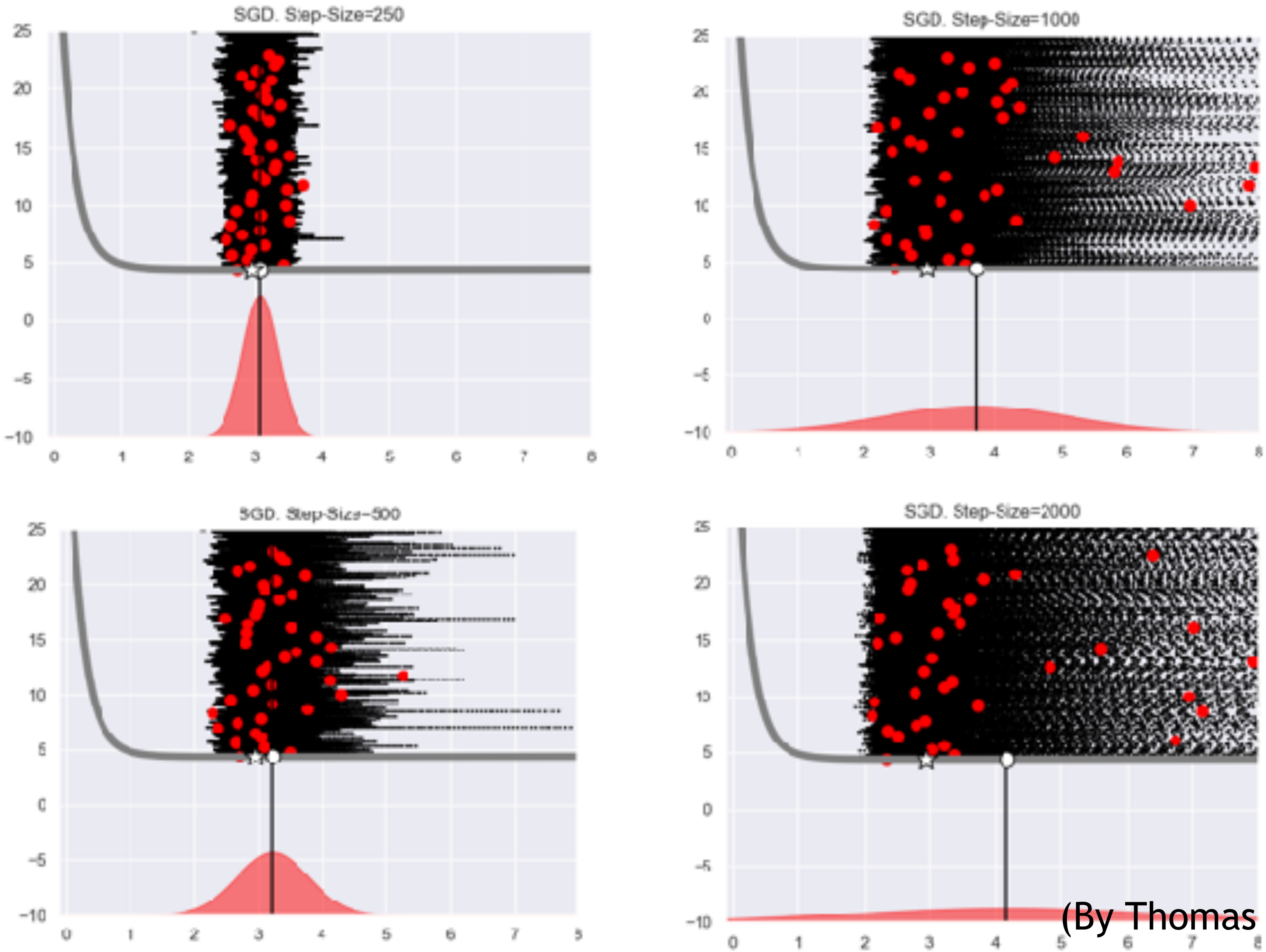


# SGD: Implicit Regularization




(By Thomas Moellenhoff)

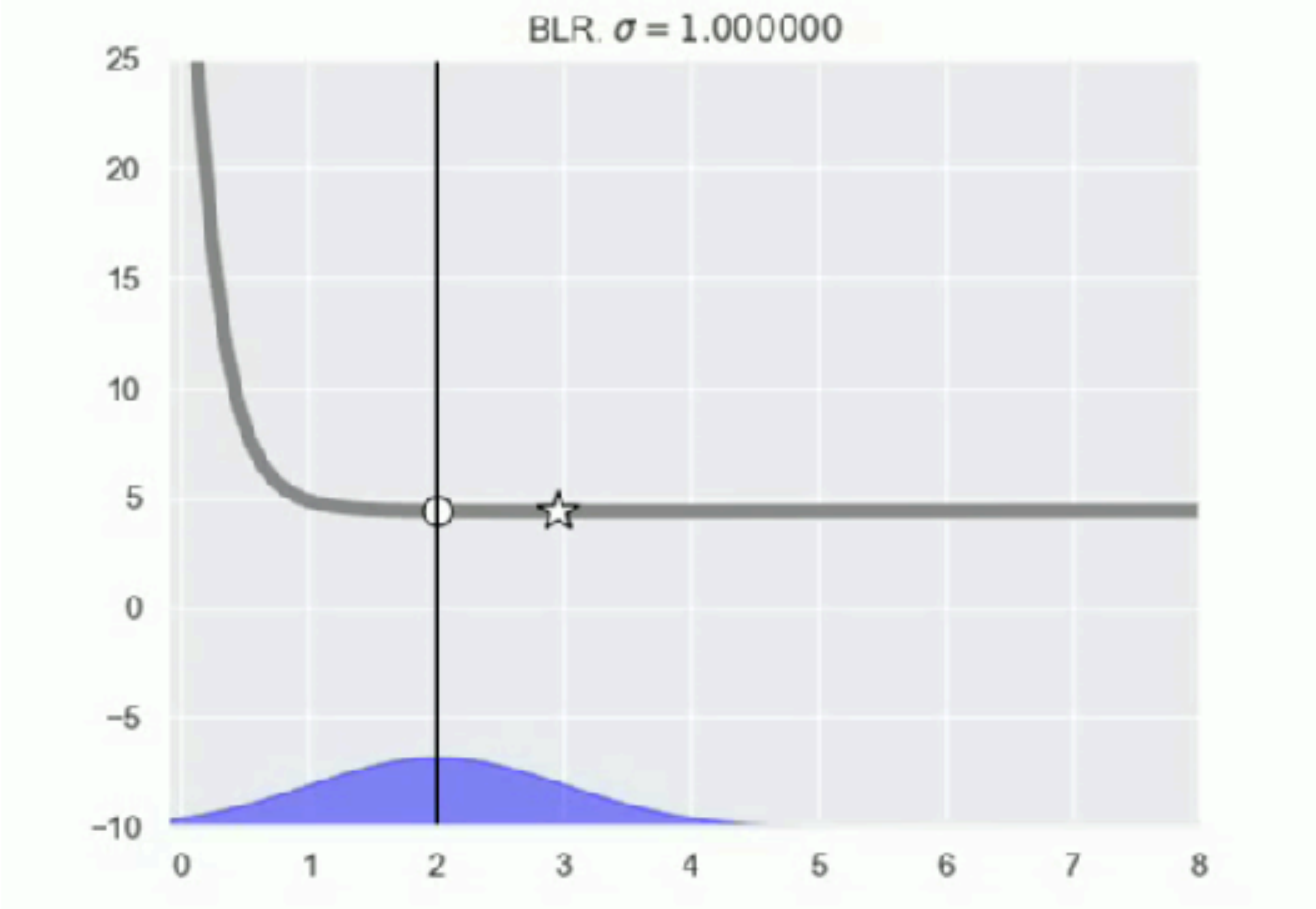
# SGD: Implicit Regularization




(By Thomas Moellenhoff)

# Bayes: Implicit Regularization

Estimating Gaussian posteriors where the variance is fixed, and only the mean is estimated

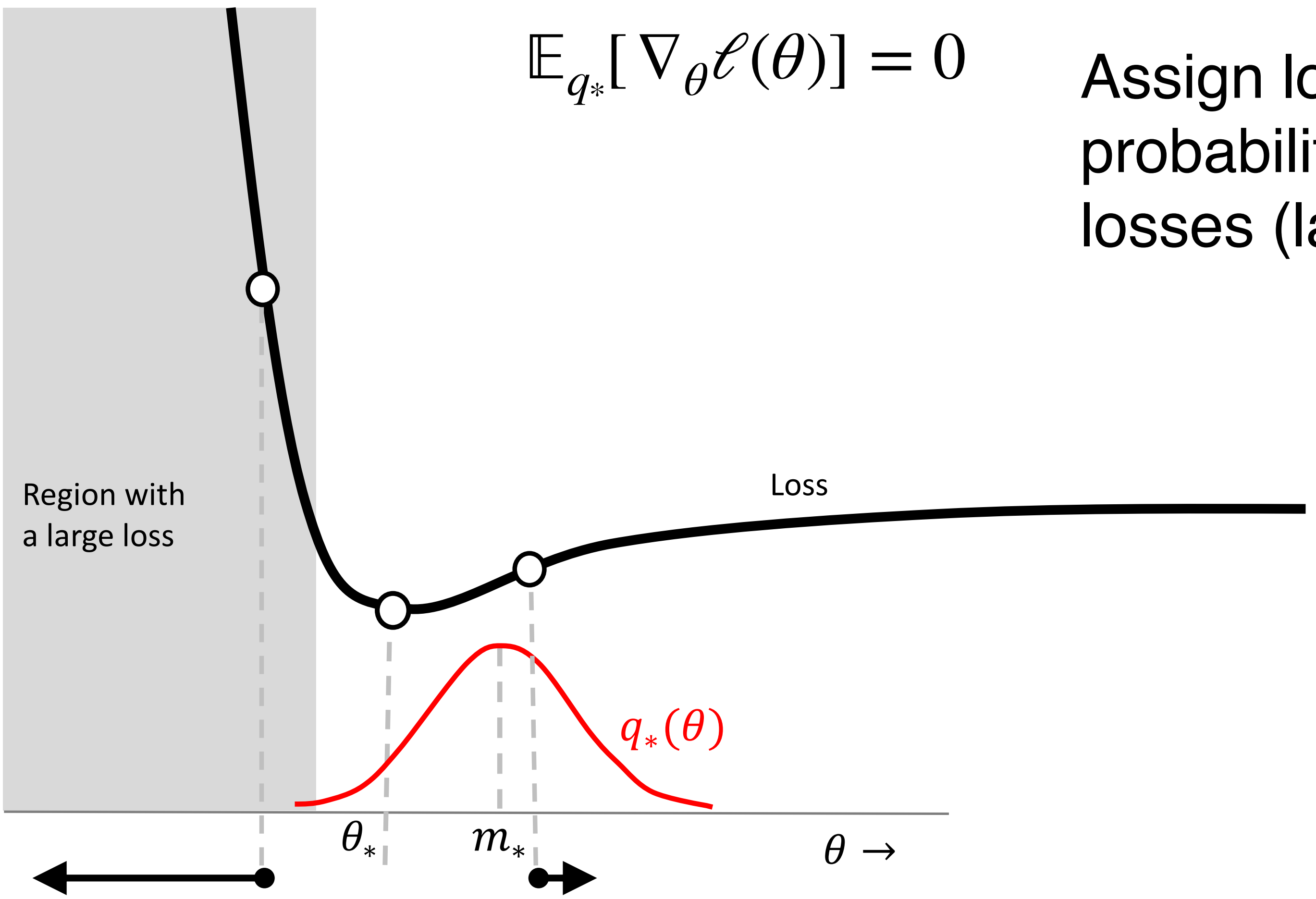


See Section 4.4 in Khan and Rue, 2021

# Bayes: Implicit Regularization

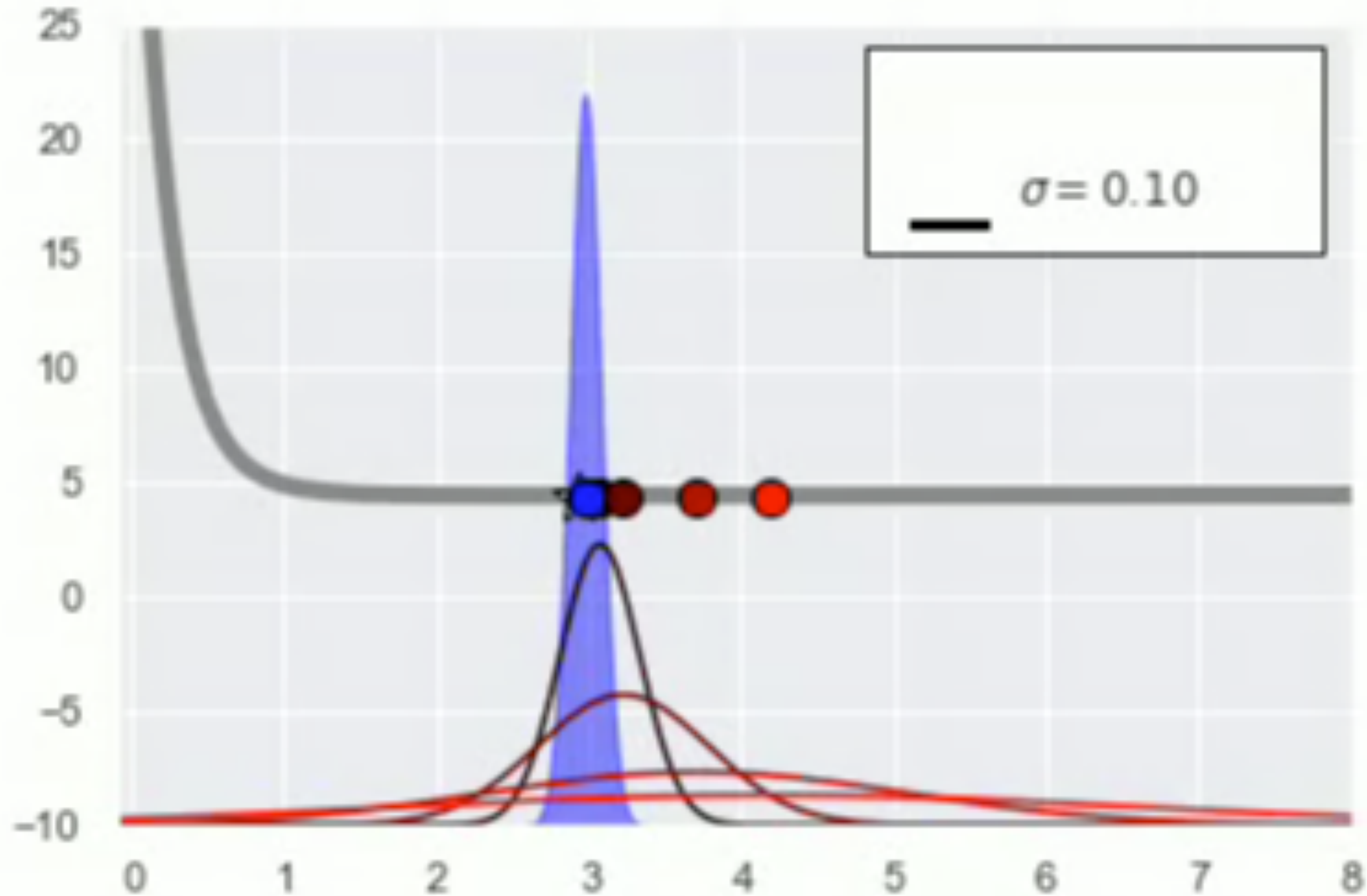
$$\mathbb{E}_{q_*}[\nabla_\theta \ell(\theta)] = 0$$

Assign lower probability to higher losses (large -ve grad)



# Bayes: Implicit Regularization

Bayes solutions (blue) compared to SGD solutions (red lines)

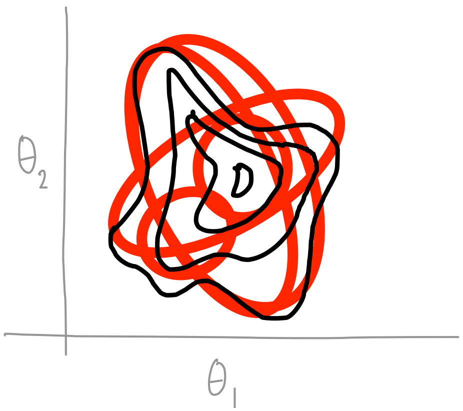


See Section 1.3 and 3.2 in Khan and Rue, 2021

# Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex ⟷ Simple




Bayes' rule

Mixture of Newton

Newton

Gradient Descent

See Section 1.3.2 in Khan and Rue, 2021

# Newton's Method from Bayes

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

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

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

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

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

Derived by choosing a multivariate Gaussian

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

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

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

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

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

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

Express in terms of gradient and Hessian of loss:

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

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

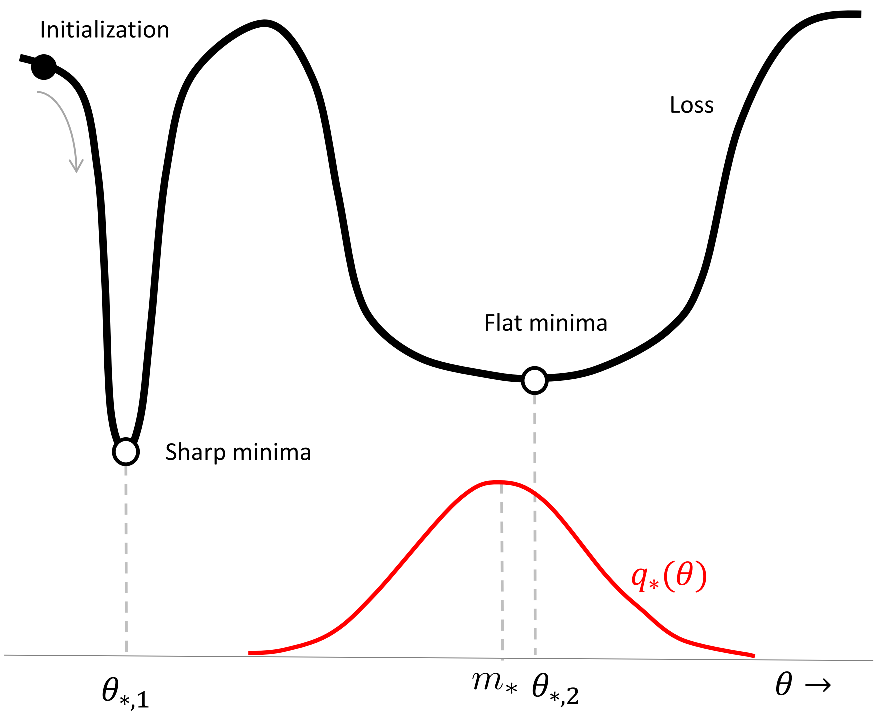
$$Sm \leftarrow (1-\rho) Sm - \rho \nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)]$$
$$S \leftarrow (1-\rho) S - \rho 2 \nabla_{\mathbb{E}_q(\theta\theta^\top)} \mathbb{E}_q[\ell(\theta)]$$

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

See Section 4.4 in Khan and Rue, 2021

# Bayes leads to robust solutions

## Avoiding sharp minima



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

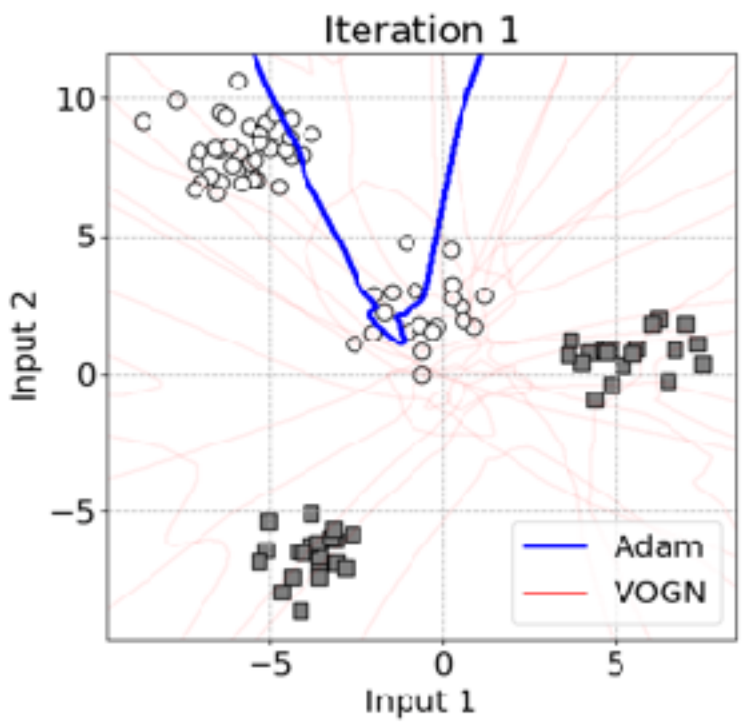
| Learning Algorithm | Posterior Approx. | Natural-Gradient Approx. | Sec. |
|---|---|---|---|
| **Optimization Algorithms** | | | |
| Gradient Descent | Gaussian (fixed cov.) | Delta method | 1.3 |
| Newton's method | Gaussian | ——"—— | 1.3 |
| Multimodal optimization (New) | Mixture of Gaussians | ——"—— | 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 |

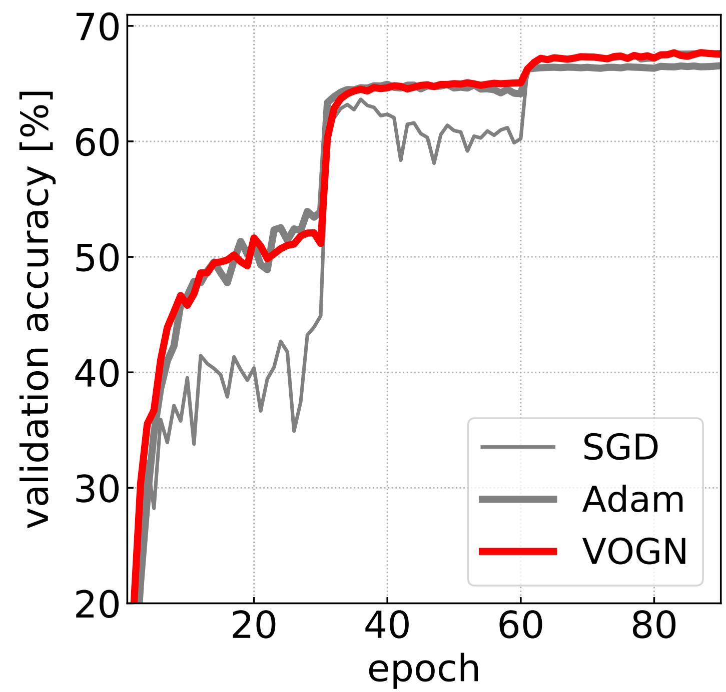
We can compute uncertainty using a variant of Adam.


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).

# Uncertainty of Deep Nets

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




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

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

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 Newton Methods

RMSprop

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

Variational Online Gauss-Newton

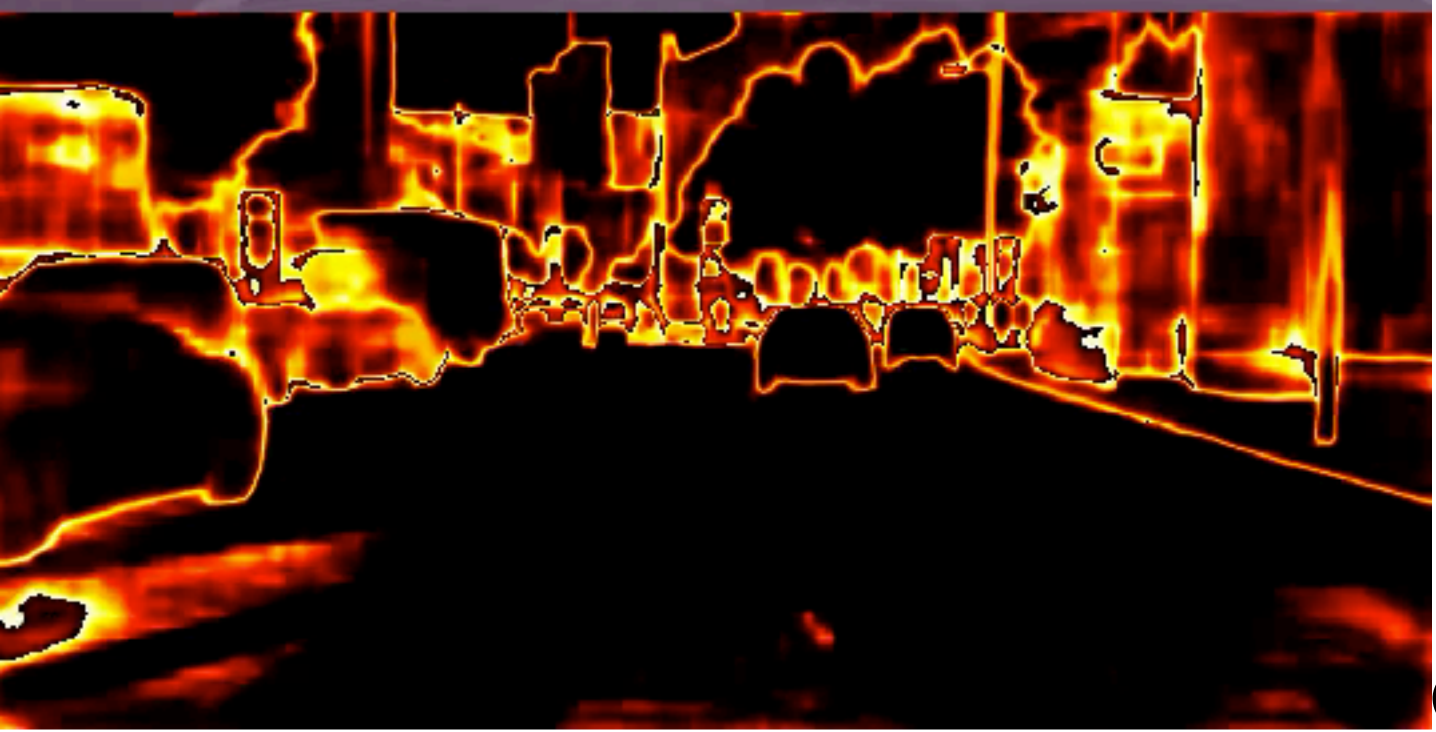
$$g \leftarrow \hat{\nabla}\ell(\theta), \text{ where } \theta \sim \mathcal{N}(m, \sigma^2)$$
$$s \leftarrow (1-\rho)s + \rho(\Sigma_i g_i^2)$$
$$m \leftarrow m - \alpha(s+\gamma)^{-1}\nabla_\theta \ell(\theta)$$
$$\sigma^2 \leftarrow (s+\gamma)^{-1}$$

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

The BLR variant from [3] led to the winning solution for the NeurIPS 2021 challenge for “approximate inference in deep learning”. Watch Thomas Moellenhoff’s talk at https://www.youtube.com/watch?v=LQInlN5EU7E.

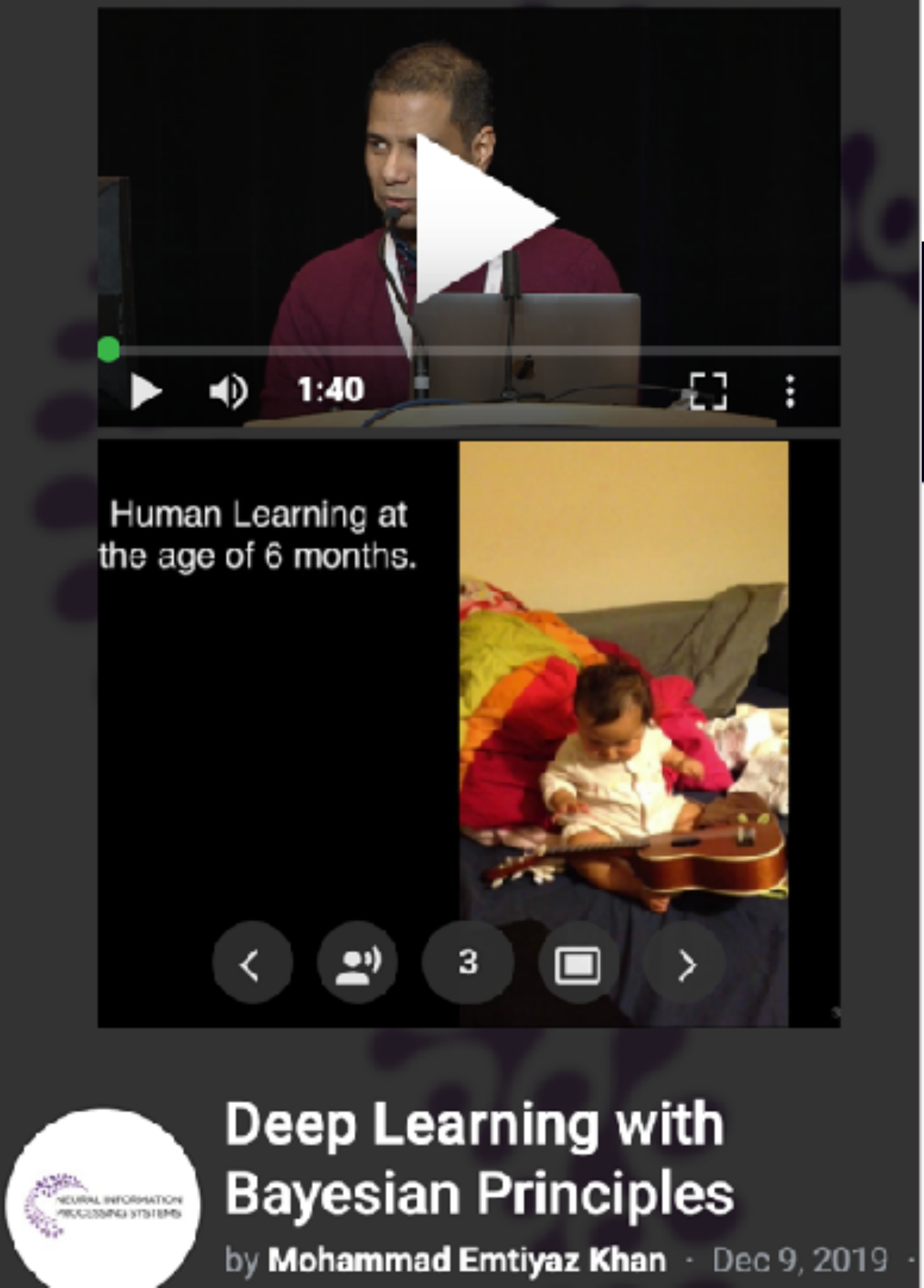

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. “Practical Deep Learning with Bayesian Principles.” NeurIPS (2019).
3. Lin et al. “Handling the positive-definite constraints in the BLR.” ICML (2020).

Image Segmentation

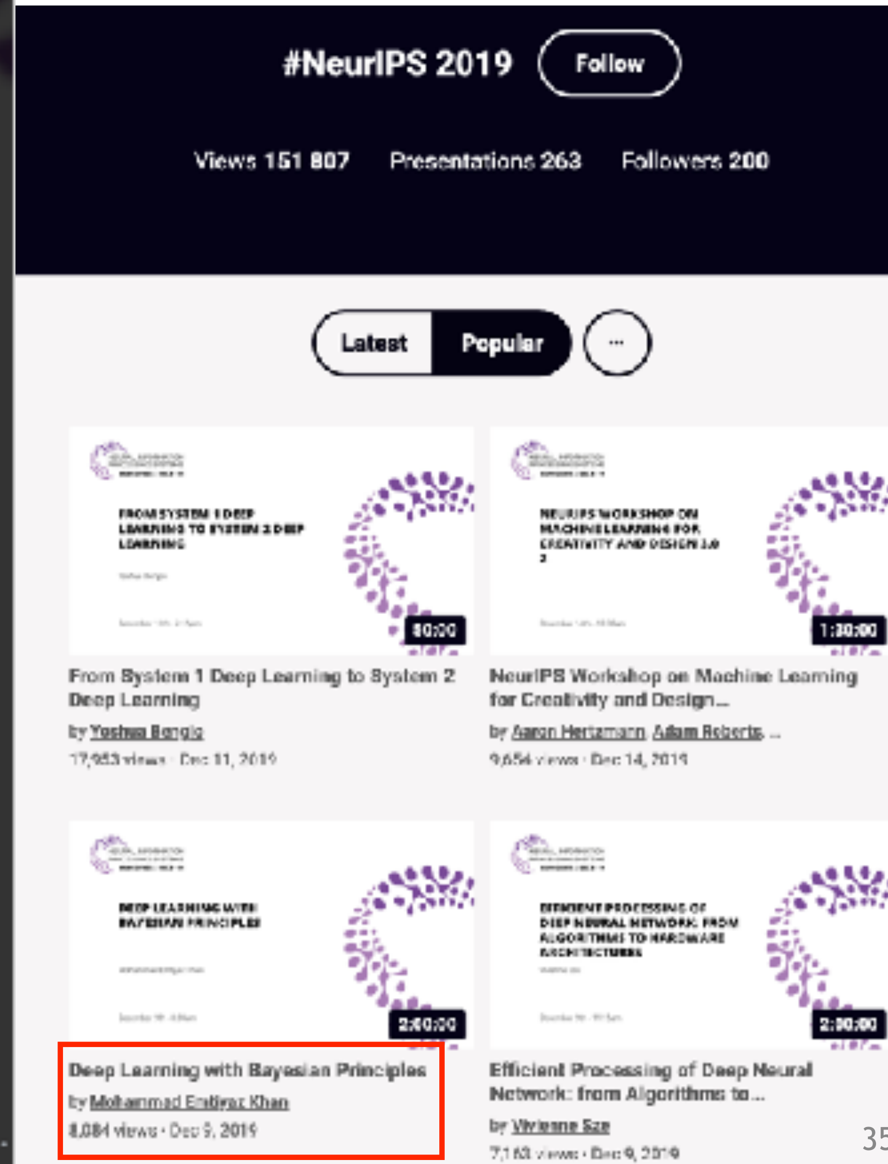


Uncertainty (entropy of class probs)

(By Roman Bachmann)




# NeurIPS 2019 Tutorial



# 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 $\longleftrightarrow$ Newton-variants

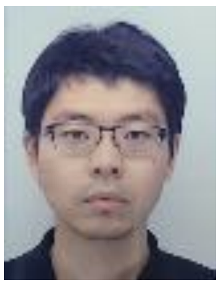

Wu Lin (UBC)


Mark Schmidt (UBC)

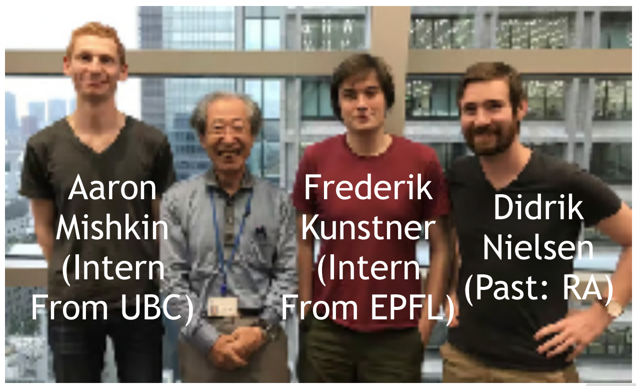

Frank Nielsen (Sony)

# Gaussian Approximation and DL

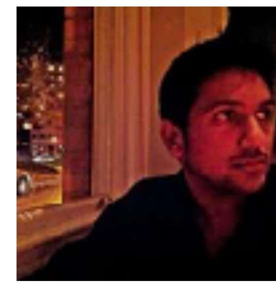

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Mishkin et al. "SLANG: Fast Structured Covariance Approximations for Bayesian Deep Learning with Natural Gradient" NeurIPS (2018).
3. Osawa et al. "Practical Deep Learning with Bayesian Principles."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

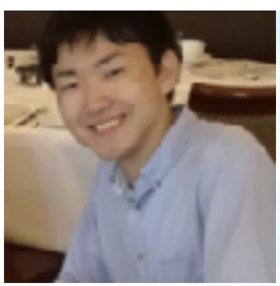
Voot Tangkaratt
(Postdoc, RIKEN-AIP)




Yarin Gal
(UOxford)



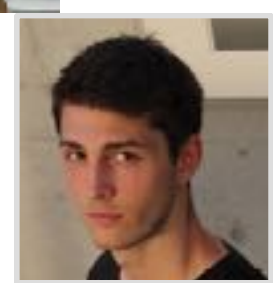
Akash Srivastava
(UEdinburgh)



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?

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

# Approximate Bayesian Inference Team

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

We have many open positions!
Come, join us.


Emtiyaz Khan
Team Leader


Pierre Alquier
Research Scientist


Gian Maria Marconi
Postdoc


Thomas Möllenhoff
Postdoc


Lu Xu
Postdoc


Jooyeon Kim
Postdoc


Wu Lin
PhD Student
*University of British Columbia*


David Tomàs Cuesta
Rotation Student, *Okinawa Institute of Science and Technology*


Dharmesh Tailor
Remote Collaborator
*University of Amsterdam*

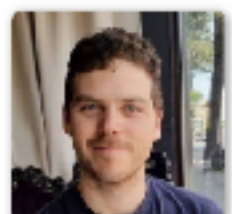

Erik Daxberger
Remote Collaborator
*University of Cambridge*


Tojo Rakotoaritina
Rotation 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 Cambridge*


Alexandre Piché
Remote Collaborator
*MILA*


Paul Chang
Remote Collaborator
*Aalto University*