Presentation at Workshop on Machine Learning and Applications at NUS, Oct 14, 2022

The Bayesian Learning Rule

Mohammad Emtiyaz Khan

RIKEN Center for AI Project, Tokyo

http://emtiyaz.github.io

Summary of research at https://emtiyaz.github.io/papers/symposium_2021.pdf

Slides available at https://emtiyaz.github.io/papers/Oct13_2022_NUS.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

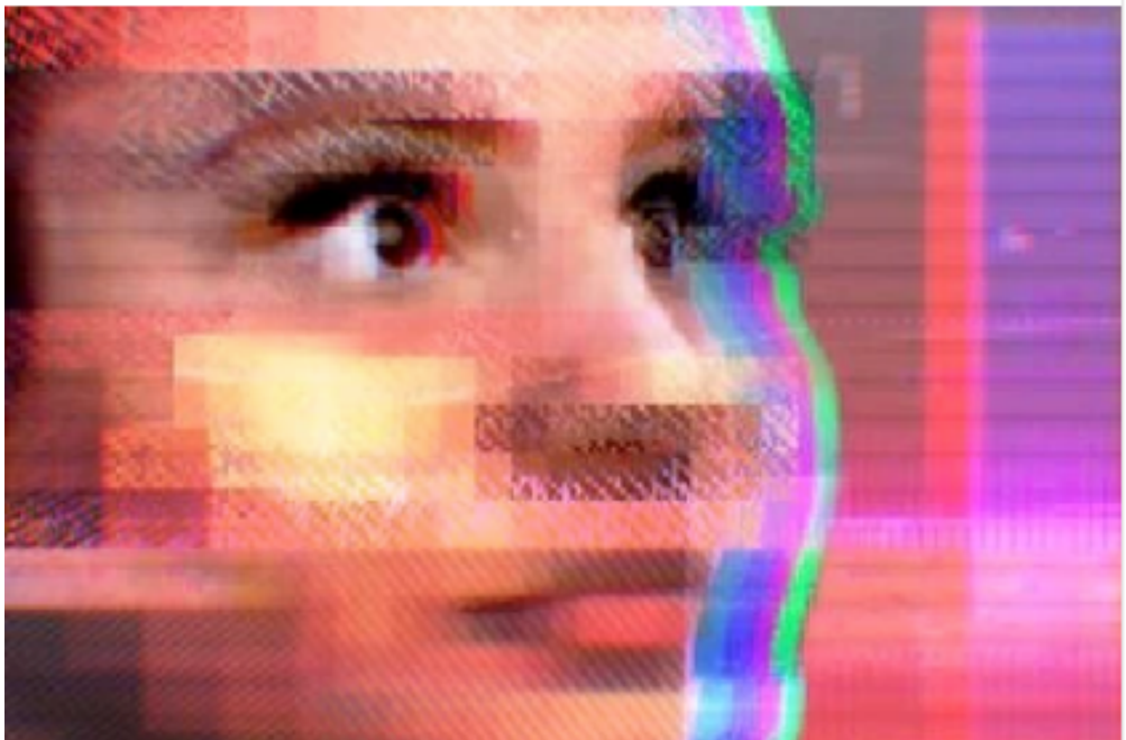
Transfer skills

at the age of 14 months

Fail because too quick to adapt

TayTweets: Microsoft AI bot manipulated into being extreme racist upon release

Posted Fri 25 Mar 2016 at 4:38am, updated Fri 25 Mar 2016 at 9:17am



TayTweets is programmed to converse like a teenage girl who has "zero chill", according to Microsoft. (Twitter: TayTweets)

https://www.abc.net.au/news/2016-03-25/microsoft-created-ai-bot-becomes-racist/7276266

Failure of AI in “dynamic” setting

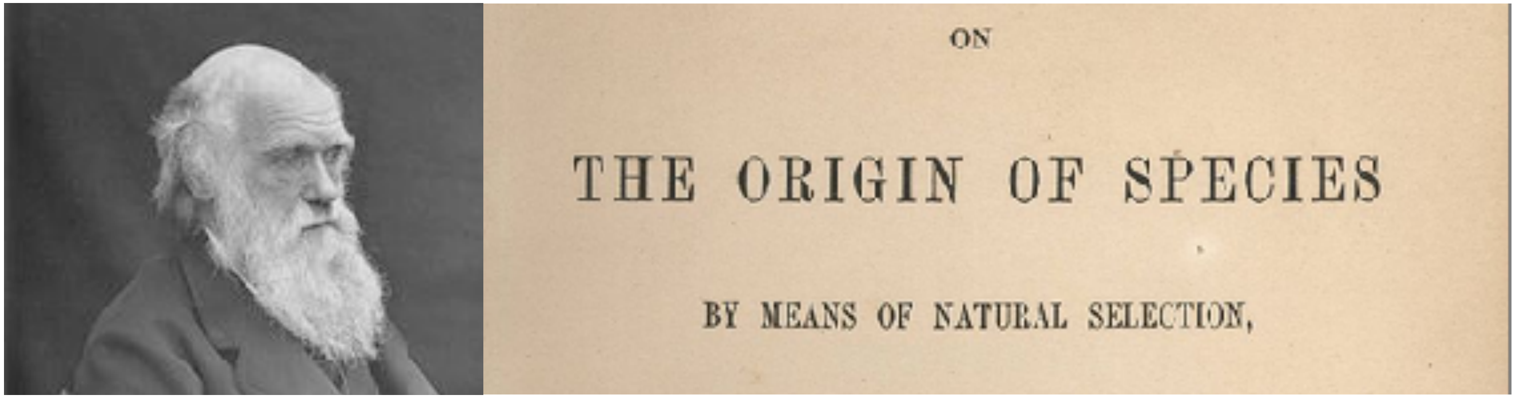
Robots need quick adaptation to be deployed
(for example, at homes for elderly care)

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

AI that learn like humans

Quickly adapt to learn new skills, throughout their lives

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



The Origin of Algorithms

What are the common principles behind popular algorithms?

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

Principles of “good” algorithms?

- Why Bayes?
- Information Geometry of Bayes
  - To unify/generalize/improve learning-algorithms
  - Optimize for “posterior approximations”
- Bayesian Learning rule (BLR)
  - Derive many algorithms from optimization, deep learning, and Bayesian inference
- Natural Gradients are Everywhere!

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

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.

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

Loss

Data

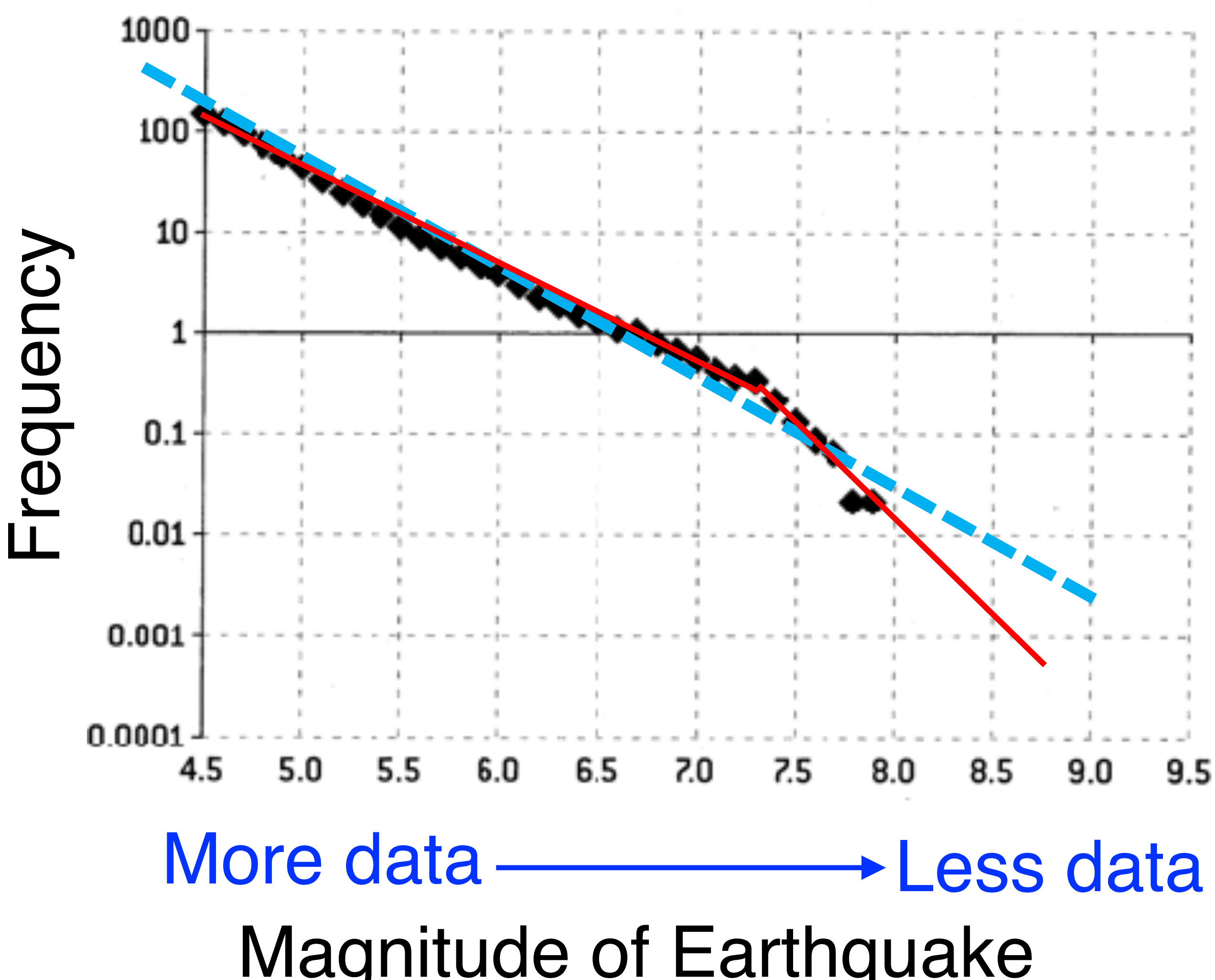
Model Params

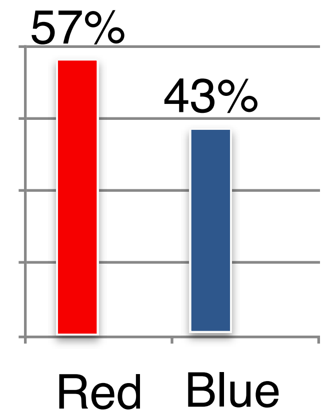
Deep Network

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?

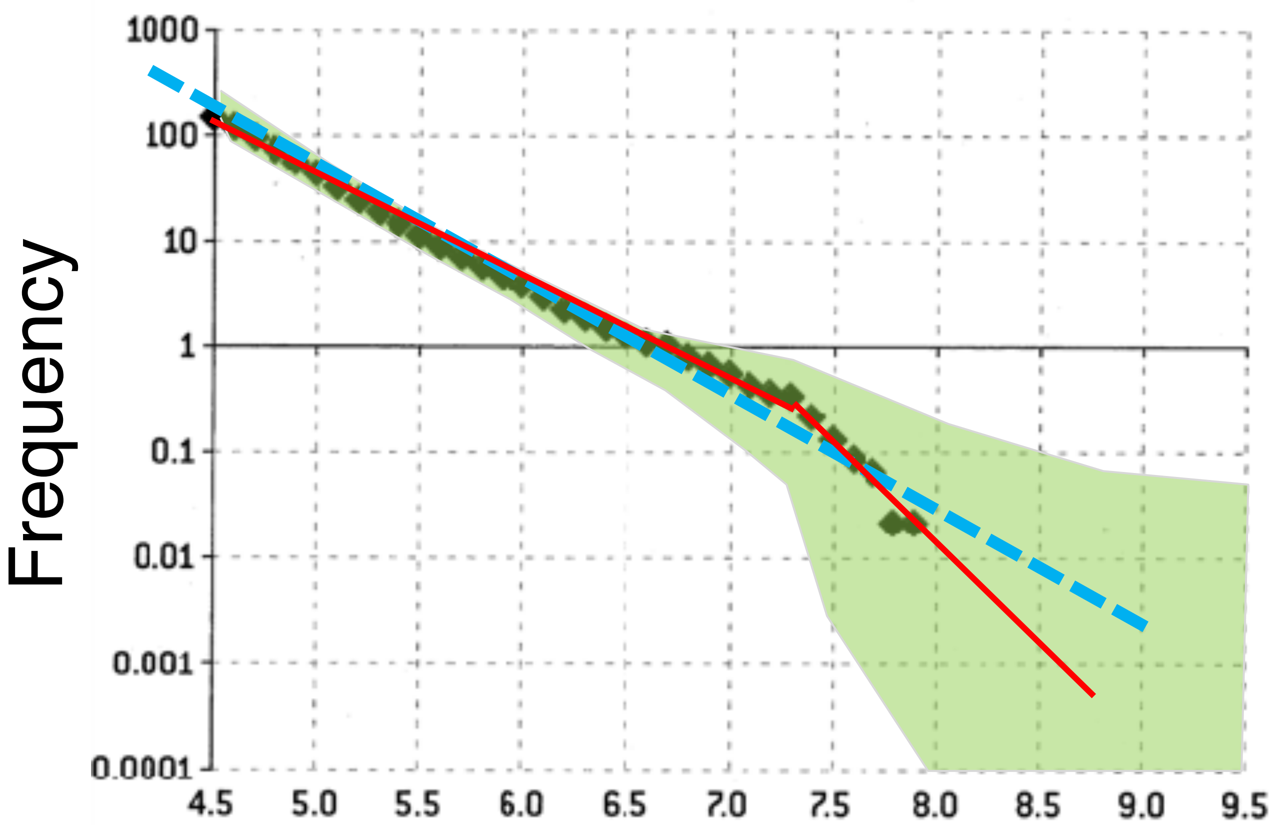




Red is more risky than the blue

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

Example: Which is a Better Fit?



More data → Less data

Magnitude of Earthquake

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

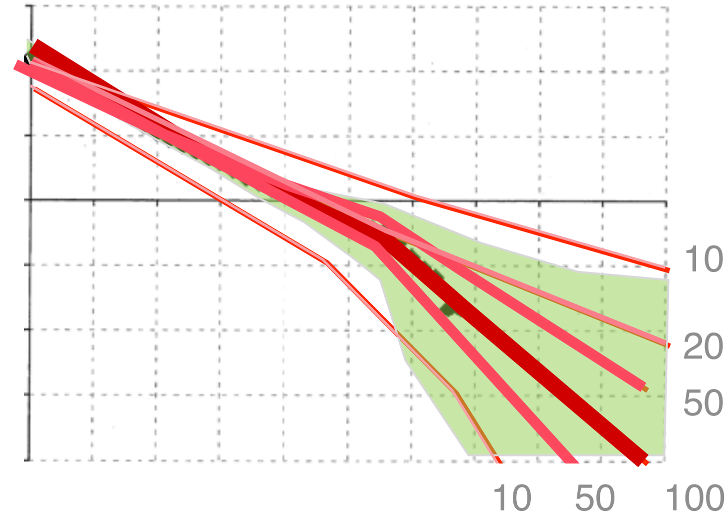
1. Sample $\theta \sim p(\theta)$ prior

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

3. Normalize

Posterior Likelihood x Prior

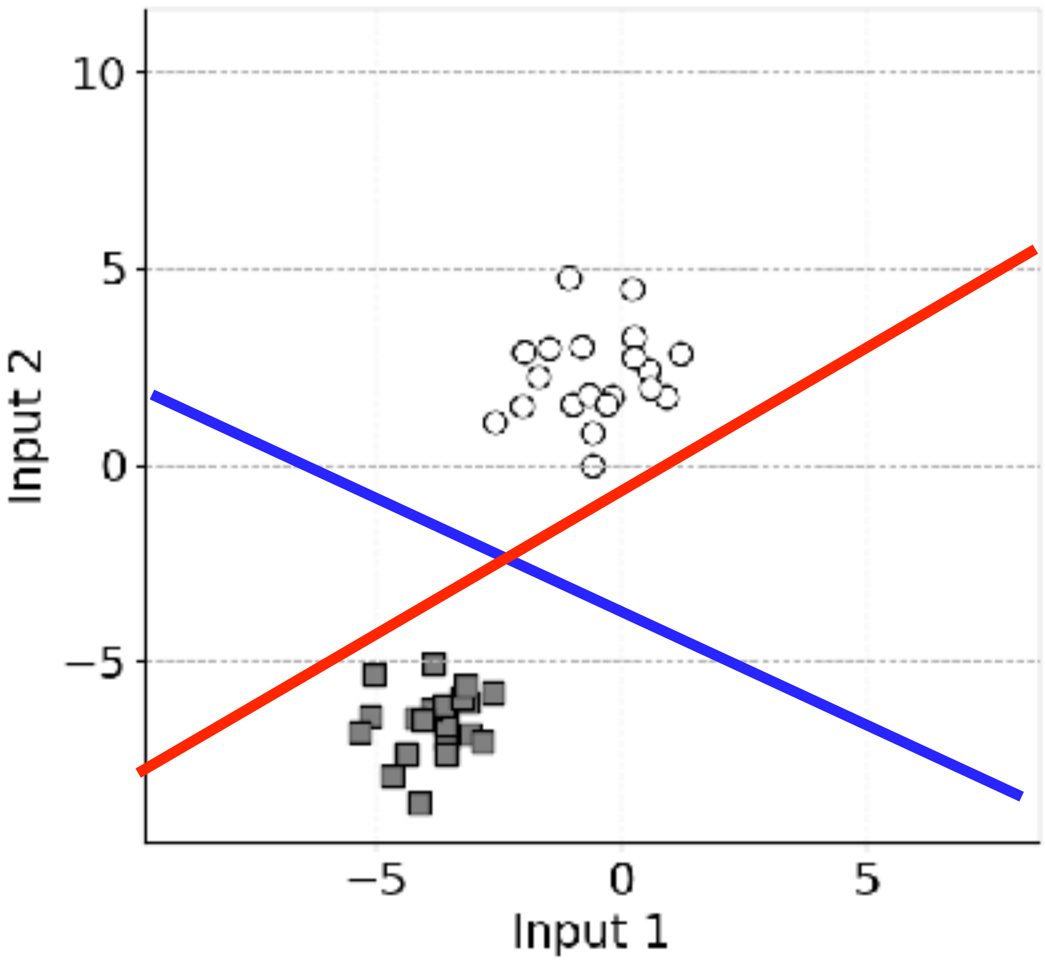
$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$$



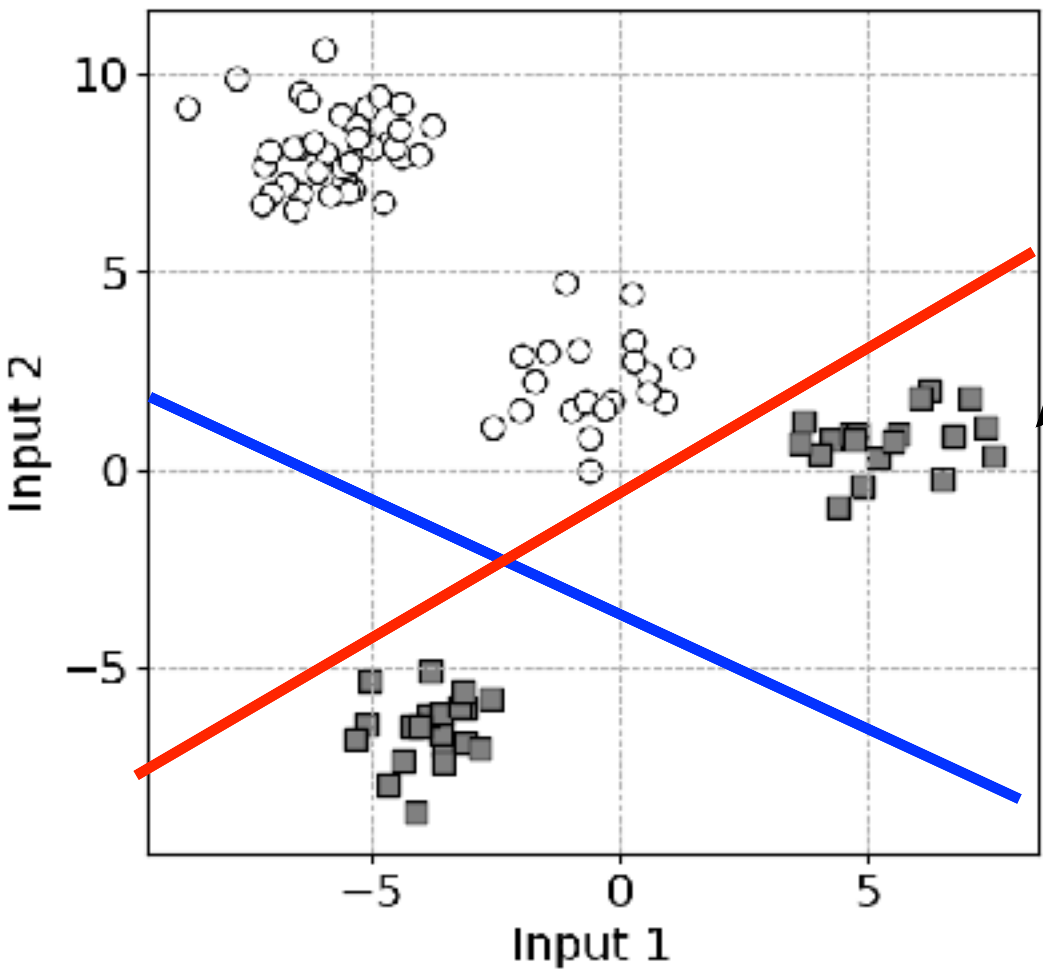
A global method: Integrates over all models

Does not scale to large problem

Which is a good classifier?



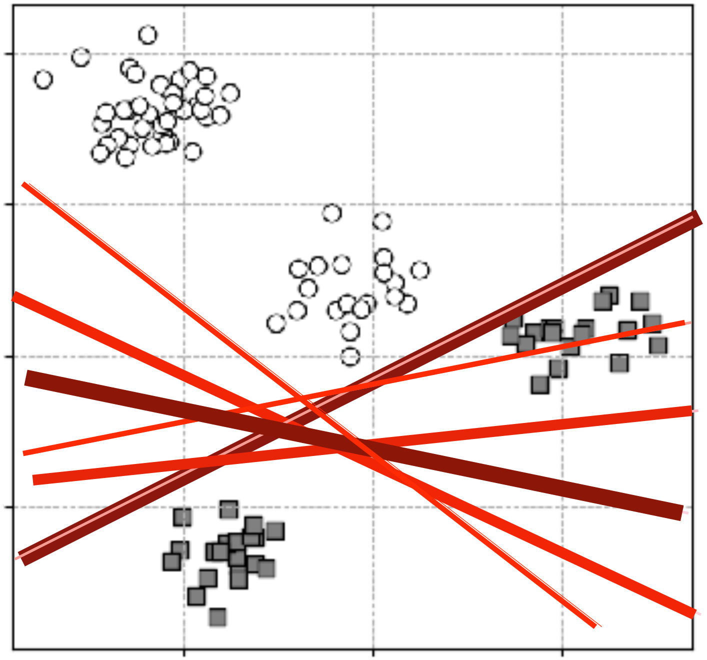
Which is a good classifier?



Misclassified by the red line, but not by the blue

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

Bayesian Principles



(1) Keep your options open

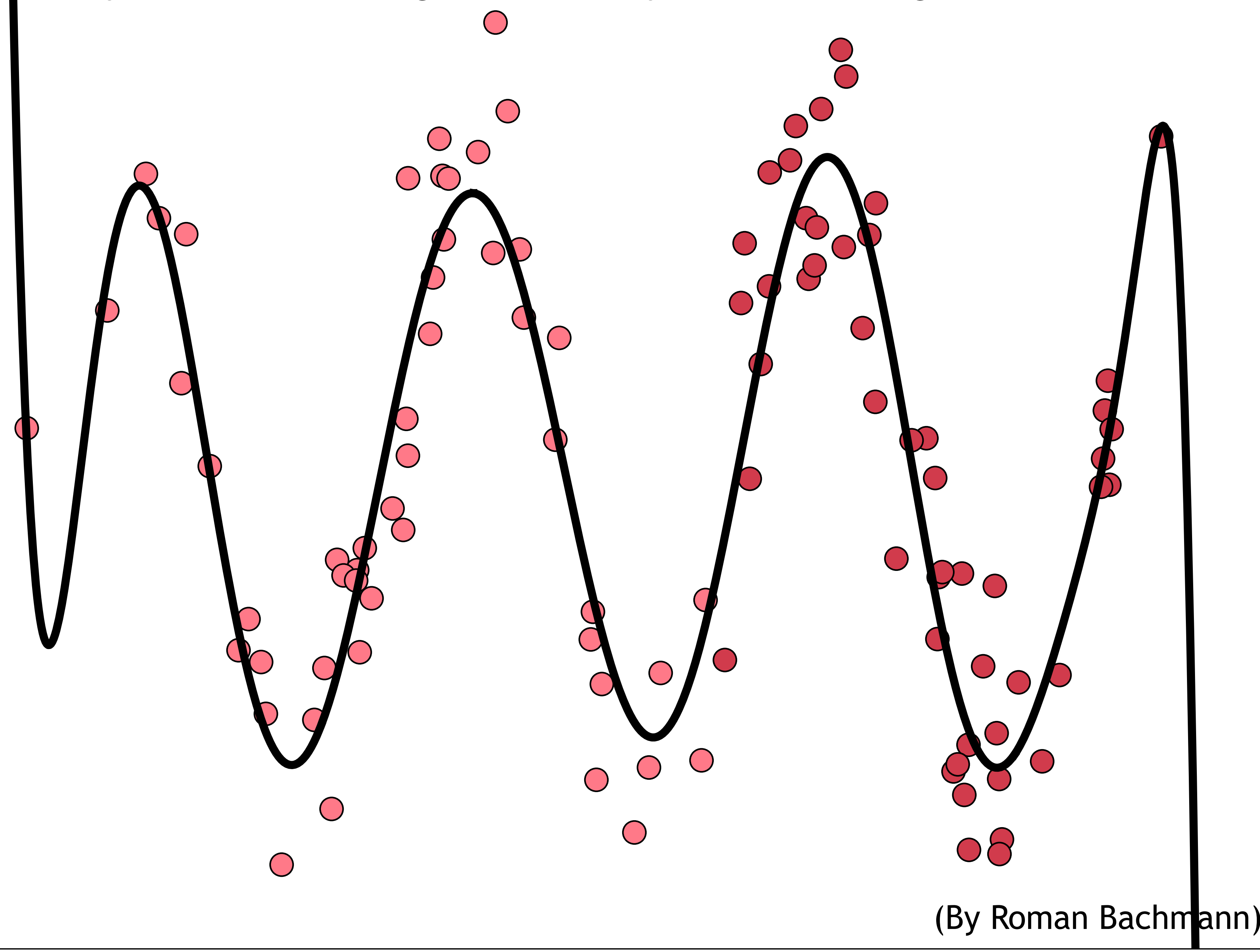
$$p(\theta|\mathcal{D}_1) = \frac{p(\mathcal{D}_1|\theta){\color{red}p(\theta)}}{\int p(\mathcal{D}_1|\theta)p(\theta)d\theta}$$

(2) Revise with new evidence

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

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

Bayesian Linear Regression (polynomials of degree 15)

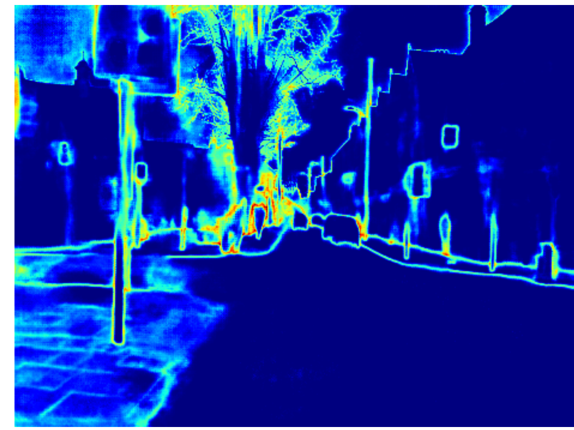


(By Roman Bachmann)

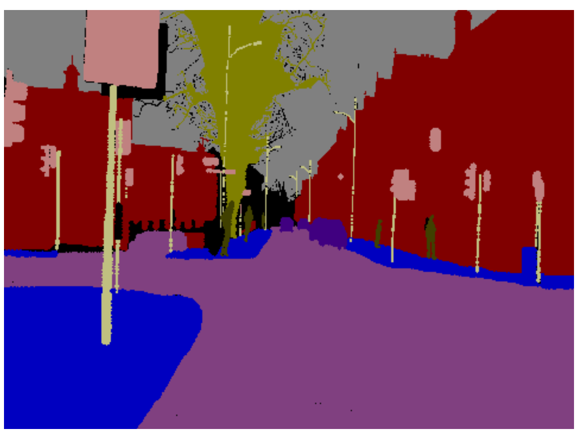
Uncertainty Estimates for Image Segmentation

Image

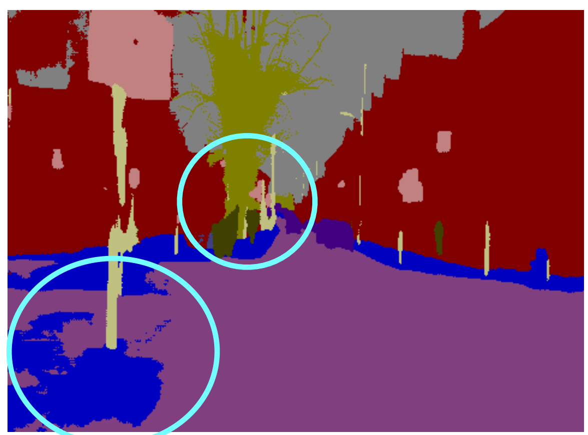
Uncertainty

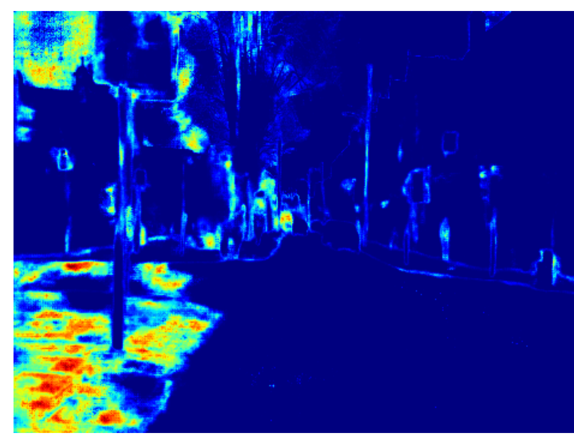


True Segments



Prediction

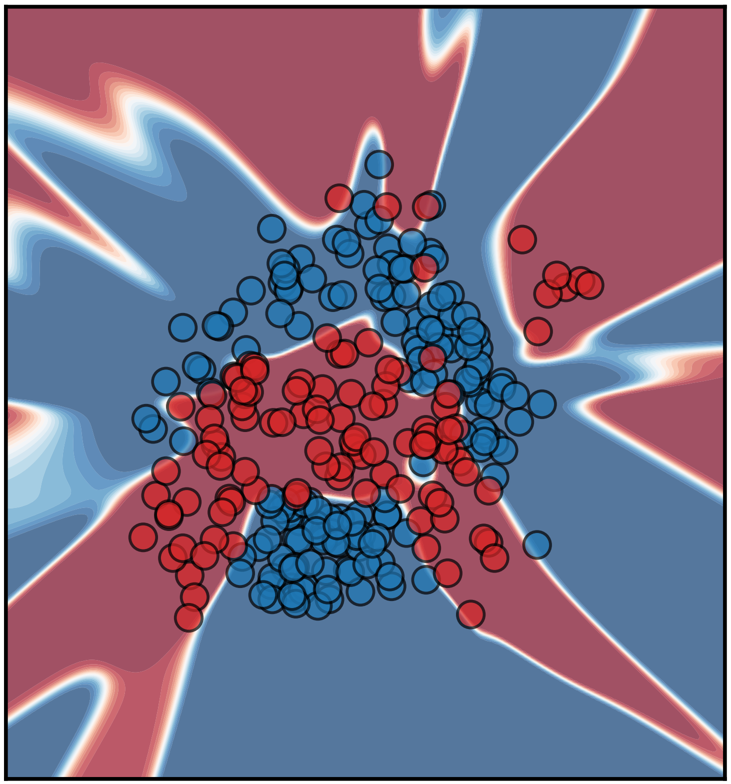




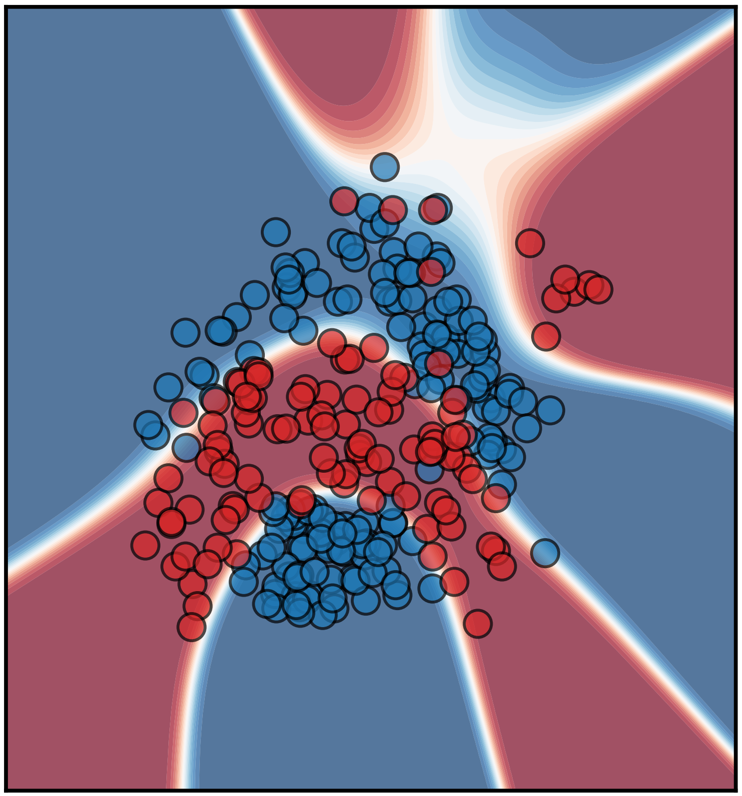
Kendall, Alex, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics." *CVPR*. 2018.

Reduce Overfitting

Standard DL



Bayesian DL

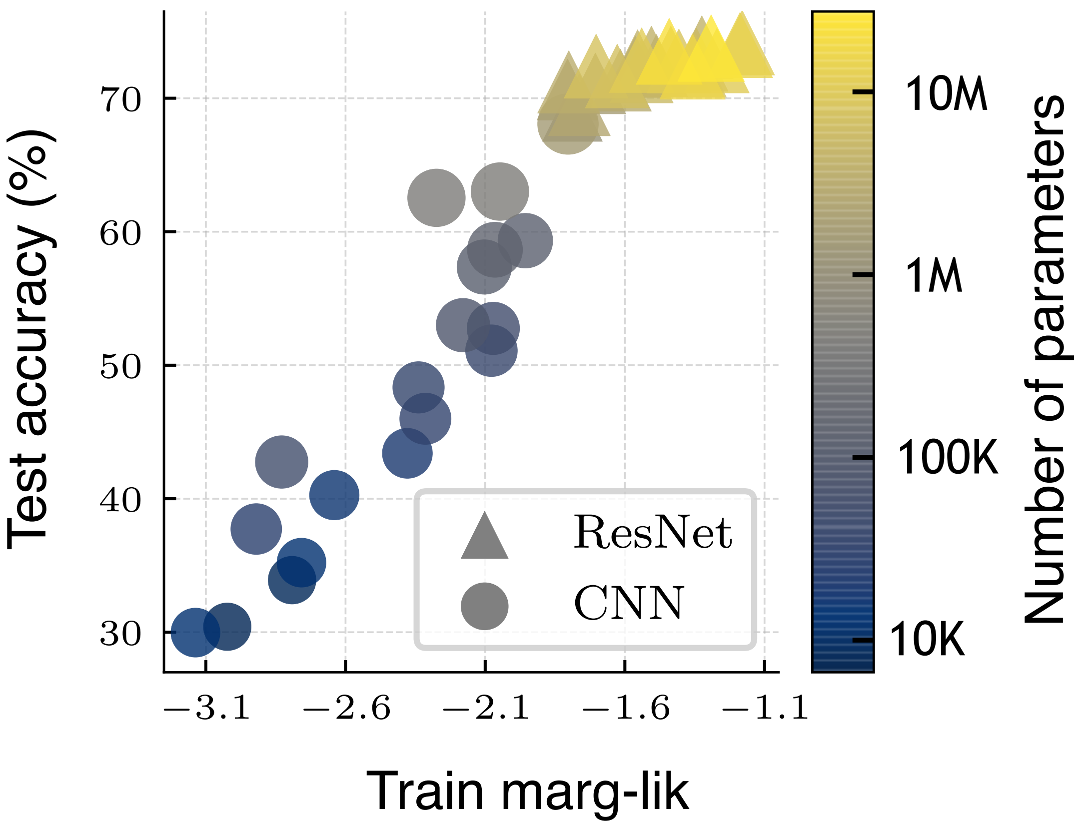


Left figure is cross-validation. Right figure is “Marginal Likelihood”.

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

Deep learning

Not scalable

Scalable

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

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

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

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

Bayesian Learning Rule

“Everything” from Bayes’ geometry
New information as natural gradients

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

Bayesian learning rule

See Table 1 in Khan and Rue, 2021

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

Principle of Trial-and-Error

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

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

Loss

Data

Model Params

Deep Network

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

We will derive them as special instances of a rule exploiting information geometry of Bayes.

Geometry of Exponential Family

We will exploit the geometry of "minimal" 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	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

1. Wainwright and Jordan, Graphical Models, Exp Fams, and Variational Inference Graphical models 2008
2. Malago et al., Towards the Geometry of Estimation of Distribution Algos based on Exp-Fam, FOGA, 2011

The Bayesian Learning Rule

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

Posterior approximation (expo-family)

Bayesian Learning Rule [1,2] (natural-gradient descent)

Natural and Expectation parameters of q

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

$$\lambda \leftarrow \underbrace{(1-\rho)\boxed{\lambda}}_{\text{Old belief}} - \rho \underbrace{\nabla_{\mu} \mathbb{E}_q[\ell(\theta)]}_{\text{New information = natural gradients}}$$

Exploiting posterior's information geometry to derive existing algorithms as special instances by approximating q and natural gradients.

1. Khan and Rue, The Bayesian Learning Rule, arXiv, https://arxiv.org/abs/2107.04562, 2021
2. Khan and Lin. "Conjugate-computation variational inference…." AIstats (2017).

Warning!

- This natural gradient might be different from the one what we (often) encounter in machine learning for Maximum-Likelihood
  - In MLE, the loss is the negative log probability distribution

$$\min_{\theta} - \log q(\theta) \Rightarrow F(\theta)^{-1} \nabla \log q(\theta)$$

  - Here, loss and distribution are two different entities, even possible unrelated

$$\min_{q} \mathbb{E}_q[\ell(\theta)] - \mathscr{H}(q) \Rightarrow F(\lambda)^{-1} \nabla_\lambda \mathbb{E}_q[\ell(\theta)]$$

Gradient Descent from Bayesian Learning Rule

(Euclidean) gradients as natural gradients

Bayesian learning rule:

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

See Section 1.3.1 in Khan and Rue, 2021

Gradient Descent from BLR

GD: $\theta \leftarrow \theta - \rho \nabla_\theta \ell(\theta)$

BLR: $m \leftarrow m - \rho \nabla_m \ell(m)$

"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:

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

Put the expectation (Bayes) back in and use the Bayesian averaging.

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).

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

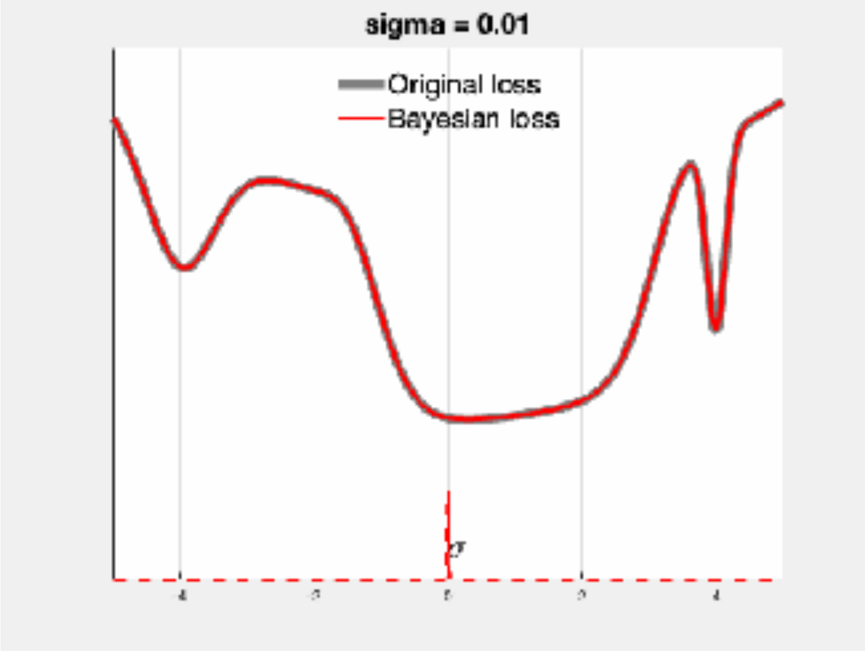
Why use Bayesian averaging?

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

Entropy

Gaussian approximation

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



- First term "smooths" the loss to favor "flatter" regions [1]
- Second term figures out how much to smooth (find σ)
- Similar mechanisms are used in DL [2], RL, search, robust optimization.

1. Foret et al. Sharpness-Aware Minimization for Efficiently Improving Generalization, ICLR, 2021
2. Smith et al., On the Origin of Implicit Regularization in Stochastic Gradient Descent, ICLR, 2021

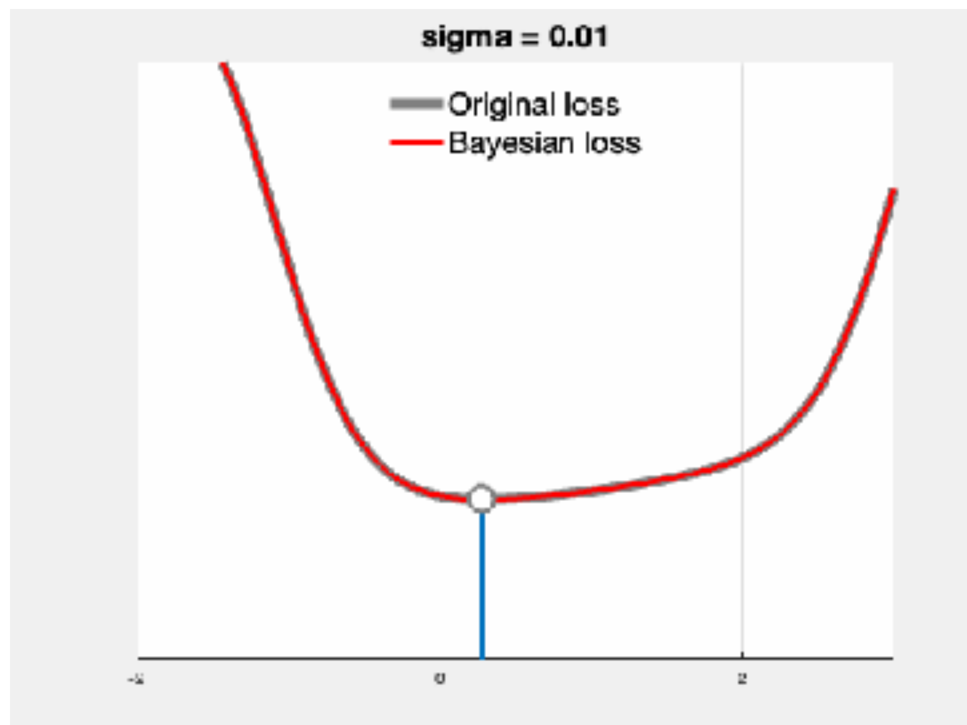
See Eq 25 in Khan and Rue, 2021 (Bonnet's theorem)

Bayes Prefers Flatter directions

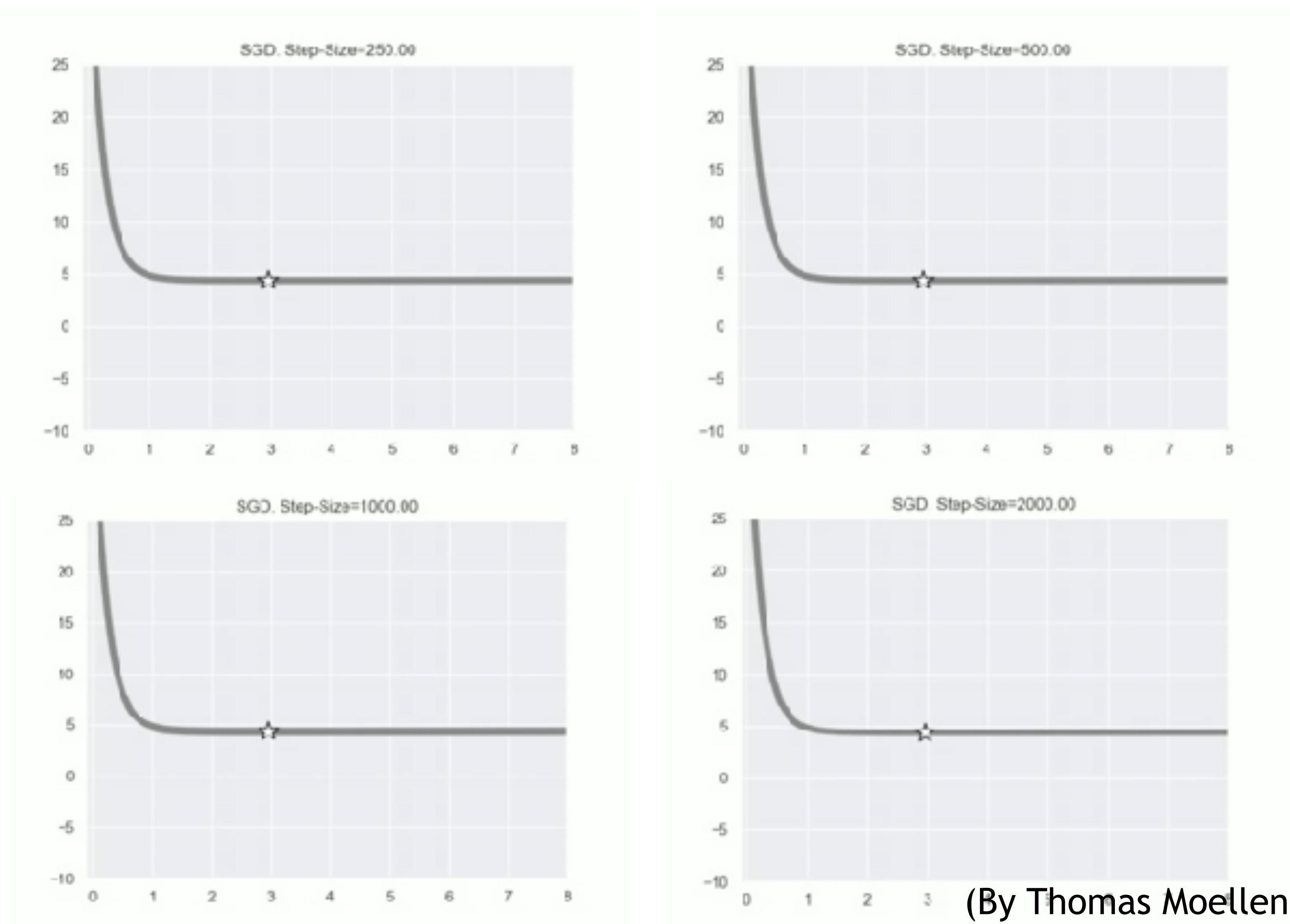
GD: $\theta \leftarrow \theta - \rho \nabla_\theta \ell(\theta) \quad\quad \Longrightarrow \nabla_\theta \ell(\theta_*) = 0$

BLR: $m \leftarrow m - \rho \nabla_m \mathbb{E}_q[\ell(\theta)] \quad \Longrightarrow \nabla_m \mathbb{E}_{q_*}[\ell(\theta)] = 0$

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

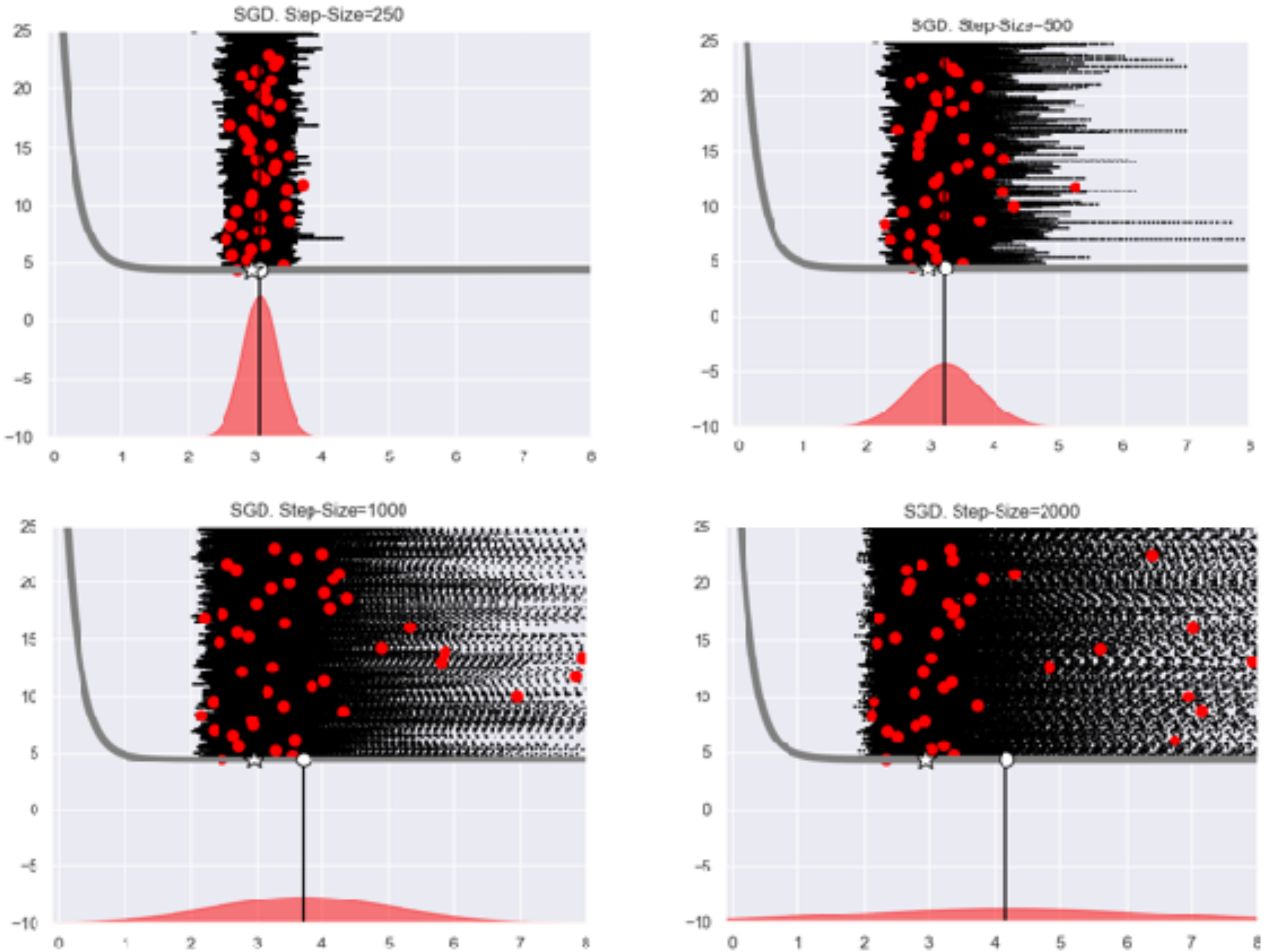


SGD: Implicit Regularization



(By Thomas Moellenhoff)

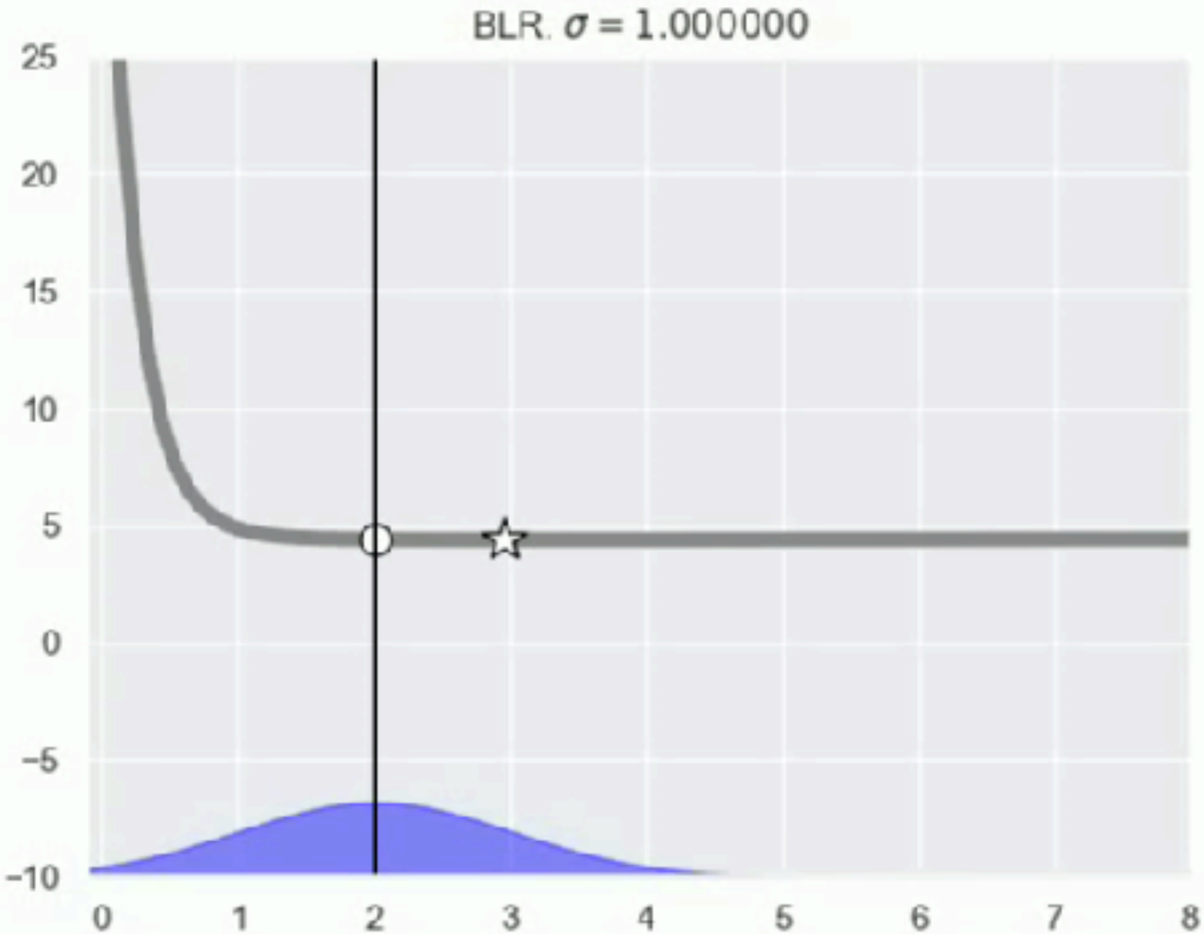
SGD: Implicit Regularization



Bayes: Explicit Regularization

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

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



By increasing the variance, we can move the mode arbitrarily far.

Bayesian"noise" has a similar regularization to the SGD noise.

It prefers "flatter" directions.

Newton's method from Bayesian Learning Rule

(Gradient, Hessian) as natural gradients

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from BLR

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

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

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

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

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

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

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

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 BLR

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

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

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

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 4.2 in Khan and Rue, 2021

RMSprop/Adam from BLR

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).

Practical DL with Bayes

RMSprop

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

BLR variant called VOGN

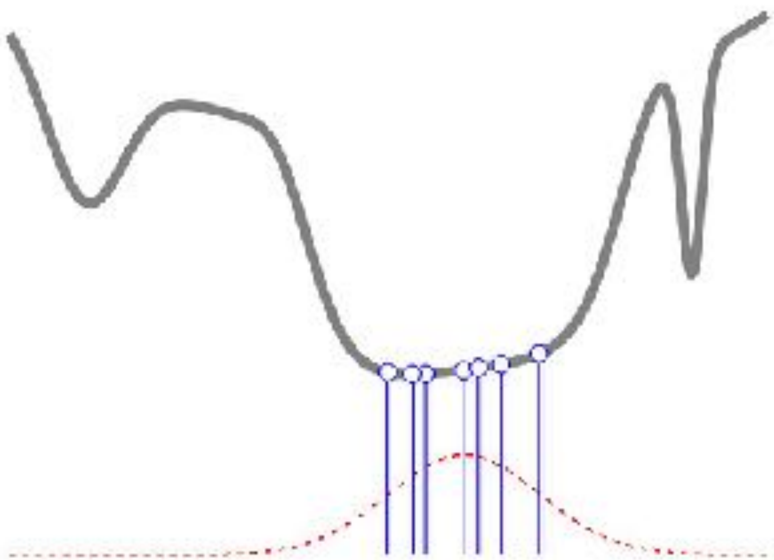
$$g \leftarrow \hat{\nabla}\ell(\theta), \text{ where } \theta \sim \mathcal{N}(m, \sigma^2)$$
$$s \leftarrow (1-\rho)s + \rho(\Sigma_i g_i^2)$$
$$m \leftarrow m - \alpha(s+\gamma)^{-1}\nabla_\theta \ell(\theta)$$
$$\sigma^2 \leftarrow (s+\gamma)^{-1}$$

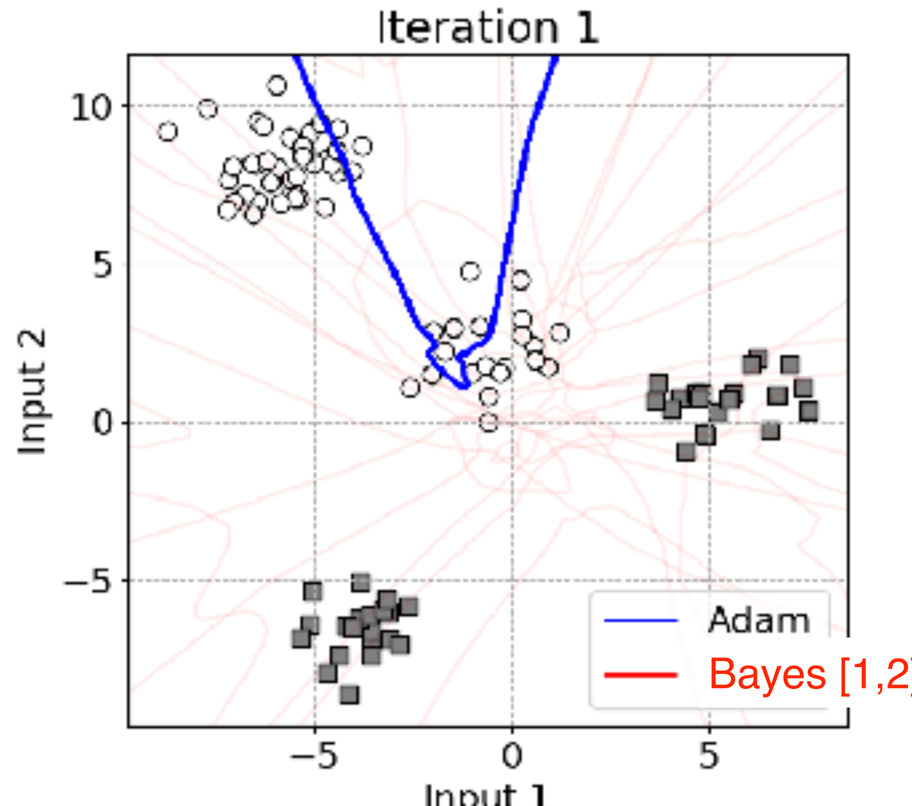
Available at https://github.com/team-approx-bayes/dl-with-bayes

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).
3. Lin et al. “Handling the positive-definite constraints in the BLR.” ICML (2020).

Why use Bayesian averaging?

Choose an "ensemble" of almost equally good models (similar to sampling in SGD trajectories)

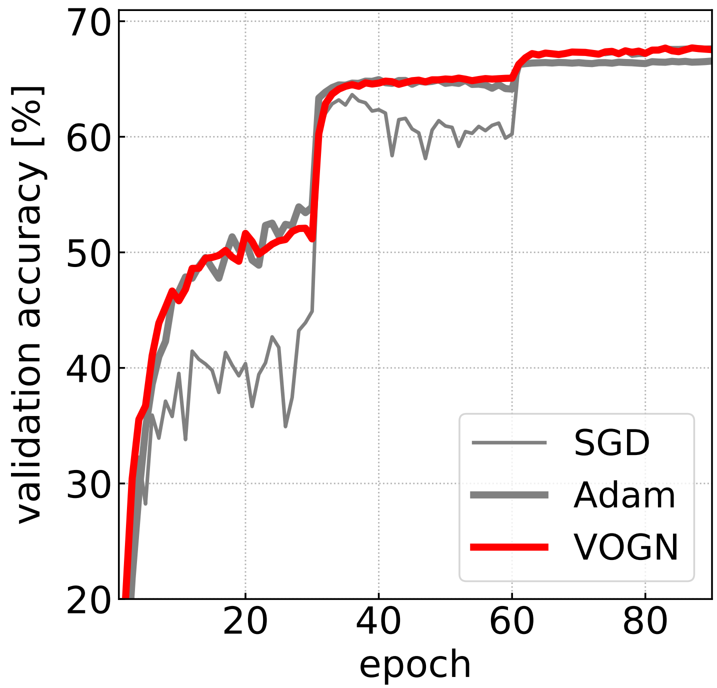




1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. "Practical Deep Learning with Bayesian Principles." NeurIPS (2019).

Uncertainty of Deep Nets

VOGN: A modification of Adam with similar performance on ImageNet, but better uncertainty



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

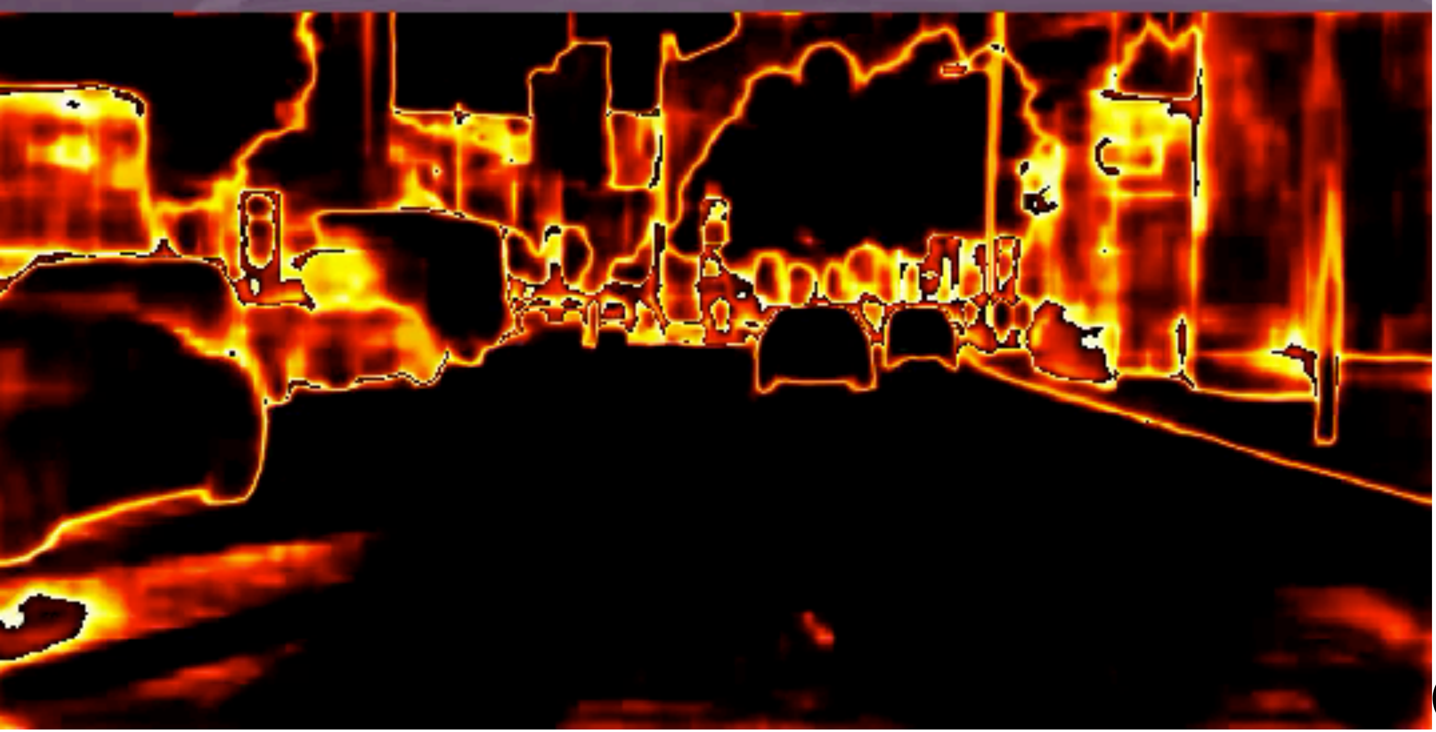
1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. “Practical Deep Learning with Bayesian Principles.” NeurIPS (2019).

BLR variant [3] got 1st prize in NeurIPS 2021 Approximate Inference Challenge

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)

Summary

- Gradient descent is derived using a Gaussian with fixed covariance, and estimating the mean
- Newton's method is derived using multivariate Gaussian
- RMSprop is derived using diagonal covariance
- Adam is derived by adding heavy-ball momentum term
- For "ensemble of Newton", use Mixture of Gaussians [1]
- To derive DL algorithms, we need to use the Delta method (a local approximation) $\mathbb{E}_q[\ell(\theta)] \approx \ell(m)$
- Then, to improve DL algorithms, we just need to add some "global" touch by relaxing the local approximation

1. Lin, Wu, Mohammad Emtiyaz Khan, and Mark Schmidt. "Fast and Simple Natural-Gradient Variational Inference with Mixture of Exponential-family Approximations." *ICML* (2019).

Bayes' Rule from Bayesian Learning Rule

"Messages" as natural gradients

Bayesian Inference as Optimization

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

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

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

All distribution Distribution Entropy

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

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

Good news: This holds for a generic loss function!

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

See Section 1.3.3 and Section 5.1 in Khan and Rue, 2021

Bayesian Inference from BLR

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

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

Sufficient statistics of q

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

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

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \boldsymbol{\rho}\left(\nabla_{\mu}\mathbb{E}_q[\ell(\theta)] + \mathcal{H}(q)\right) \implies \lambda_* = \boxed{\lambda_{\mathcal{D}}}$$

Messages

Forward-backward, SVI, Variational message passing etc. are special cases of the BLR

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

The Bayesian "Principle"

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

All distribution

Distribution

Entropy

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

Restrict the set of distributions (change P to Q)

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

Often we call this variational Inference, but it's not just a method, rather a general Bayesian principle.

Similar to IGO [1] but the entropy is essential!

1. Ollivier et al. "Information-geometric optimization algorithms: A unifying picture via invariance principles." JMLR (2017).

Our use of natural-gradients here is not a matter of choice. In fact, natural-gradients are inherently present in *all solutions of the Bayesian objective* in Eq. 2. For example, a solution of Eq. 2 or equivalently a fixed point of Eq. 3, satisfies the following,

$$\nabla_{\mu} \mathbb{E}_{q_*}[\bar{\ell}(\boldsymbol{\theta})] = \nabla_{\mu} \mathcal{H}(q_*), \text{ which implies } \tilde{\nabla}_{\lambda} \mathbb{E}_{q_*}[-\bar{\ell}(\boldsymbol{\theta})] = \boldsymbol{\lambda}_*, \tag{5}$$

for candidates with constant base-measure. This is obtained by setting the gradient of Eq. 2 to 0, then noting that $\nabla_{\mu}\mathcal{H}(q) = -\boldsymbol{\lambda}$ (App. B), and then interchanging ∇_{μ} by $\tilde{\nabla}_{\lambda}$ (because of Eq. 4). In other words, natural parameter of the best $q_*(\boldsymbol{\theta})$ is equal to the natural gradient of the expected negative-loss. The importance of natural-gradients is entirely missed in the Bayesian/variational inference literature, including textbooks, reviews, tutorials on this topic [Bishop, 2006, Murphy, 2012, Blei et al., 2017, Zhang et al., 2018a] where natural-gradients are often put in a special category.

We will show that natural gradients retrieve essential higher-order information about the loss landscape which are then assigned to appropriate natural parameters using Eq. 5. The information-matching is due to the presence of the entropy term there, which is an important quantity for the optimality of Bayes in general [Jaynes, 1982, Zellner, 1988, Littlestone and Warmuth, 1994, Vovk, 1990], and which is generally absent in non-Bayesian formulations (Eq. 1). The entropy term in general leads to exponential-weighting in Bayes' rule. In our context, it gives rise to natural-gradients and, as we will soon see, automatically determines the complexity of the derived algorithm through the complexity of the class of distributions $\mathcal{Q}$, yielding a principled way to develop new algorithms.

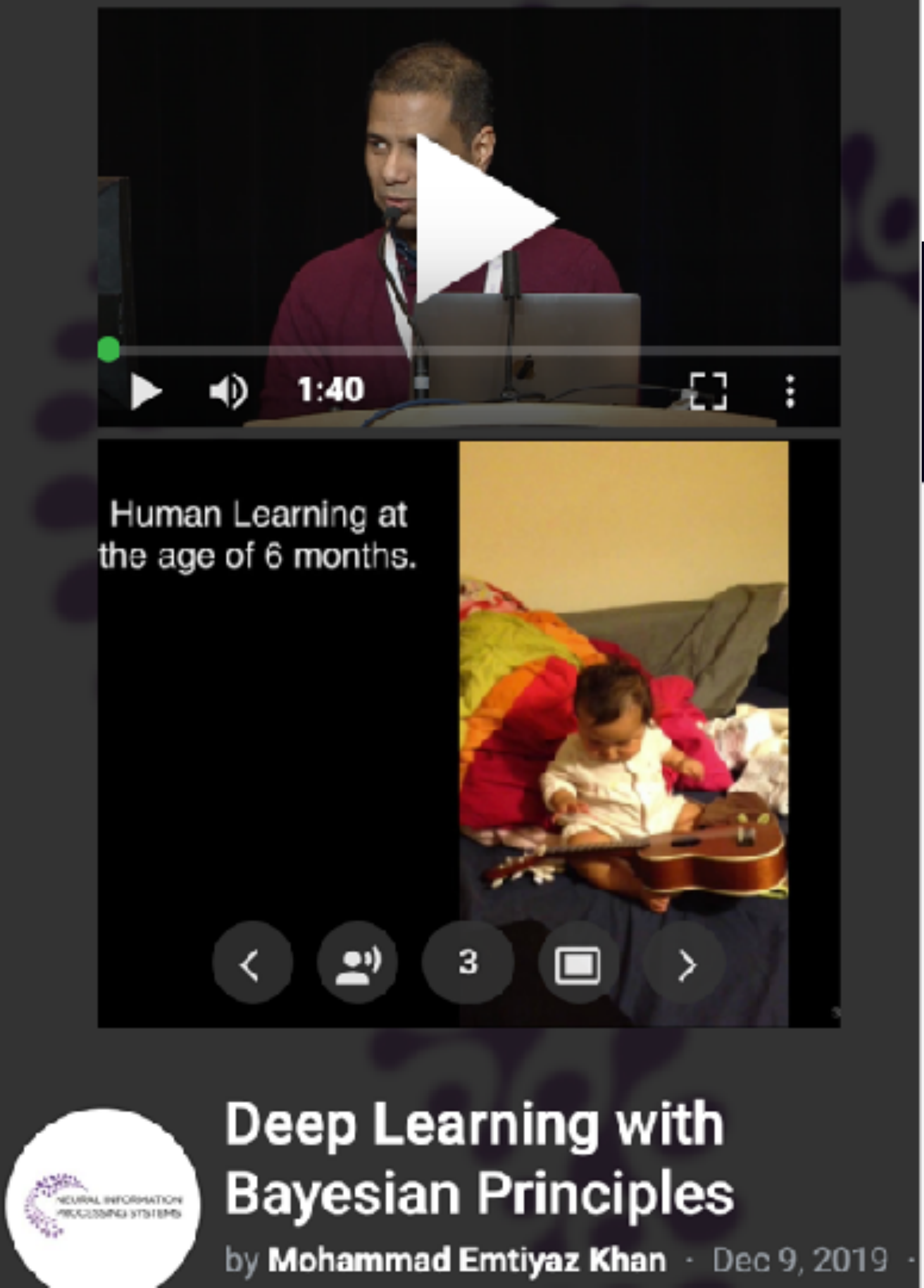
Overall, our work demonstrates the importance of natural-gradients and information geometry for algorithm design in ML. This is similar in spirit to Information Geometric Optimization [Ollivier et al., 2017], which focuses on the optimization of black-box, deterministic functions. In contrast, we derive generic learning algorithms by using the same Bayesian principles. The BLR we use is a generalization of the method proposed in Khan and Lin [2017], Khan and Nielsen [2018] specifically for approximate Bayesian inference. Here, we establish it as a general learning rule to derive many old and new learning algorithms, which include both Bayesian and non-Bayesian ones, way beyond its original proposal. We do not claim that these successful algorithms work well because they are derived from the BLR. Rather, we use the BLR to simply unravels the inherent Bayesian nature of these "good" algorithms. In this sense, the BLR can be seen as a variant of Bayes' rule, useful for generic algorithm design.

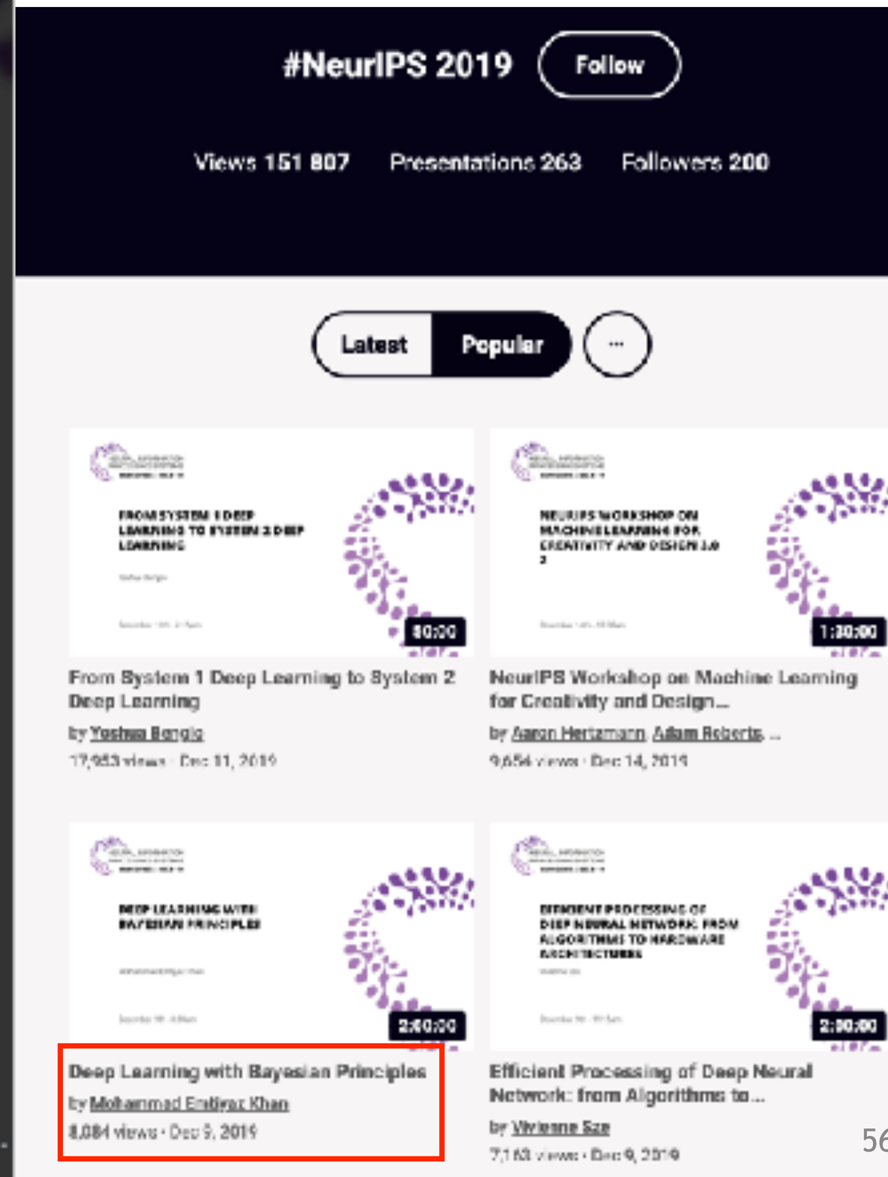
Principles of “good” algorithms?

- Information Geometry of Bayes
  - To unify/generalize/improve learning-algorithms
  - Optimize for “posterior approximations”
- Bayesian Learning rule (BLR)
  - Derive many algorithms from optimization, deep learning, and Bayesian inference
- Natural Gradients are Everywhere!

Khan and Rue, The Bayesian Learning Rule, arXiv, https://arxiv.org/abs/2107.04562, 2021
Lin et al., Tractable structured natural gradient descent using local parameterizations, ICML 2021

NeurIPS 2019 Tutorial





What's Next

- Bayesian "Duality" Principle
  - The BLR unravels a duality perspective of good algorithms
  - Unifies many results from many fields
    - convex duality, Kernel methods, Bayesian nonparametric methods, Deep Learning, Robust statistics, and Information Geometry
  - Helps to "solve" the Adaptation problem
- My talk on Monday will show two examples of this principle
  - Robust deep learning as "convex relaxation" of Bayes
  - Principle of Adaptive learning through K-priors

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

Emtiyaz Khan

Research director
(Japan side)

Approx-Bayes team at RIKEN-AIP and OIST

Julyan Arbel

Research director
(France side)

Statify-team, Inria Grenoble Rhône-Alpes

Kenichi Bannai

Co-PI (Japan side)

Math-Science Team at RIKEN-AIP and Keio University

Rio Yokota

Co-PI
(Japan side)

Tokyo Institute of Technology

Received total funding of around USD 3 million through JST's CREST-ANR and Kakenhi Grants.

Approximate Bayesian Inference Team

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


Emtiyaz Khan
Team Leader


Pierre Alquier
Research Scientist


Hugo Monzón Maldonado
Postdoc


Happy Buzaaba
Postdoc


Erik Daxberger
Remote Collaborator
University of Cambridge


Paul Chang
Remote Collaborator
Aalto University


Gian Maria Marconi
Postdoc

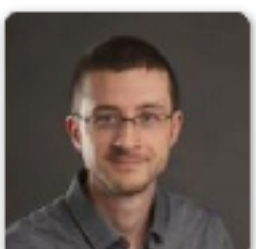

Thomas Möllenhoff
Postdoc


Lu Xu
Postdoc

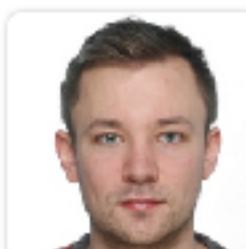

Jooyeon Kim
Postdoc


Alexandre Piché
Remote Collaborator
MILA


Ali Unlu
Intern, *Okinawa Institute of Science*


Geoffrey Wolfer
Postdoc


Wu Lin
PhD Student
University of British Columbia


Peter Nickl
Research Assistant


Dharmesh Tailor
Remote Collaborator
University of Amsterdam


Ang Mingliang
Remote Collaborator
National University of Singapore


Kenneth Chen
Intern, *Okinawa Institute of Science and Technology*