Learning-Algorithms from Bayesian Principles

Mohammad Emtiyaz Khan
RIKEN center for Advanced Intelligence Project
Tokyo, Japan

Haavard Rue
CEMSE Division
King Abdullah University of Science and Technology
Thuwal, Saudi Arabia

December 9, 2019

Abstract

Machine-learning algorithms are commonly derived using ideas from optimization and statistics, followed by an extensive empirical efforts to make them practical as there is a lack of underlying principles to guide this process. In this paper, we present a learning rule derived from Bayesian principles, which enables us to connect a wide-variety of learning algorithms. Using this rule, we can derive a wide-range of learning-algorithms in fields such as probabilistic graphical models, continuous optimization, deep learning, reinforcement learning, online learning, and black-box optimization. This includes classical algorithms such as least-squares, Newton's method, and Kalman filter, as well as modern deep-learning algorithms such as stochastic-gradient descent, RMSprop and Adam. Overall, we show that Bayesian principles not only unify, generalize, and improve existing learning-algorithms, but also help us design new ones. [**This is a working draft and a work in progress**]

1 Learning algorithms

Machine Learning (ML) methods have been extremely successful in solving many challenging problems in fields such as computer vision, natural-language processing, and artificial intelligence (AI). The main idea is to formulate those problems as *prediction problems*, and *learn* a *model* on existing data to predict the future outcomes. For example, to design an AI agent that can recongnize objects, we collect a dataset $\mathcal{D} := \{\mathbf{x}_i, y_i\}_{i=1}^N$ with images $\mathbf{x}_i \in \mathbb{R}^D$ and object labels $y_i \in \{1, 2, 3, \ldots, K\}$, and learn a model $f_{\mathbf{w}}(\mathbf{x})$ with parameters $\mathbf{w}$ to predict the labels for a new image $\mathbf{x}_*$. *Learning algorithms* are often employed to estimate the mode parameters $\mathbf{w}$ using the *princinple of trial-and-error*. For example, the well-known *Empricial Risk Minimization (ERM)* approach uses the following criteria:

$$\min_{\mathbf{w}} \bar{\ell}(\mathbf{w}) := \sum_{i=1}^{N} \ell(y_i, \mathbf{f}_{\mathbf{w}}(\mathbf{x}_i)) + \mathcal{R}(\mathbf{w}). \tag{1}$$

where $\ell(y, \mathbf{f}_{\mathbf{w}}(\mathbf{x}))$ is a loss function that encourages the model to predict well and $\mathcal{R}(\mathbf{w})$ is a regularizer that prevents it from overfitting.

A wide-variety of such learning-algorithms exist in the literature that minimize different types of loss functions, e.g., least-squares, Kalman filters, stochastic-gradient descent, and Newton's method etc. Design of such algorithms plays a key role behind the recent success of modern machine-learning algorithms.

2 Lack of common principles to derive learning-algorithms

Often, learning-algorithms are derived by borrowing ideas from a diverse set of fields, such as, statistics, optimization, statistical physics, information theory, and computer science. Borrowing and combinging ideas from such diverse fields is a strong point of the ML community, and has given rise to a wide-variety of learning algorithm. This diversity, however, comes with a price. The very nature of machine learning is data centric and the charasteristics of data is problem dependent. A successful application of learning algorithms, in many situations, becomes an art involving many tricks-of-the-trade which need subtle tuning for every specific pair of the problem and model. Discovery of such tricks is a costly process and typically requires extensive empirical investigations. For every new algorithm, researchers are often forced to reinvent these tricks from scratch. Deployment of these new algorithms also results in a huge change in practice, e.g., changes during data collection and preprocessing, as well as in codebases. Such costs discourage algorithmic innovations and forces researchers and practitioners to resort to simpler approximations, even when the approximations are suboptimal, brittle, and theoretically unsound. Currently, there is a lack of common principles that can guide the design and tuning process, and it is our opinion that this widens the gap between theory and practice, which in the long run would not be benefitial for ML research community.

3 Learning-algorithms from Bayesian Principles

The goal of this paper is to introduce Bayesian principles as a common set of principle to derive a wide-variety of learning algorithms. Our key idea is to use the following Bayesian reformulation of the learning problem such as (1) where the optimization is over distributions $q(\mathbf{w})$ instead of $\mathbf{w}$:

$$\min_{q(\mathbf{w})\in\mathcal{Q}} \mathcal{L}(q) := \mathbb{E}_{q(\mathbf{w})}\left[\sum_{i=1}^{N} \ell(\mathbf{y}_i, \mathbf{f}_{\mathbf{w}}(\mathbf{x}_i))\right] + \mathbb{D}_{KL}[q(\mathbf{w}) \,\|\, p(\mathbf{w})], \tag{2}$$

where $\mathcal{Q} \subseteq \mathcal{P}$ is a set of probability distribution, $p(\mathbf{w}) \propto \exp\left[-\mathcal{R}(\mathbf{w})\right]$, and $\mathbb{D}_{KL}[\,\|\,]$ is the Kullback-Leibler divergence. We refer to this as the *Bayesian learning problem*.

When the loss corresponds to the log of a probability distribution $\ell(\mathbf{y}, f_w(\mathbf{x})) := -\log p(\mathbf{y}|f_w(\mathbf{x}))$, then the above minimization results in the optimal $q_*(\mathbf{w})$ to be equal to the posterior distribution. Fortunately, the above objective holds for a more general class of loss functions and prior distributions (Zellner, 1988; Bissiri et al., 2016). In such cases, when $\mathcal{Q} \equiv \mathcal{P}$, the optimum is the following the following posterior distribution:

$$q_*(\mathbf{w}) \propto \prod_{i=1}^{N} e^{-\ell(y_i, \mathbf{f}_{\mathbf{w}}(\mathbf{x}_i))} p(\mathbf{w}) \tag{3}$$

We show that by restricting the set $\mathcal{Q} \subset \mathcal{P}$ to a carefully chosen approximation, we can obtain a variety of existing learning algorithms from the above Bayesian principle.

4 Bayesian approximations

Throughout the paper, we will focus on cases where $\mathcal{Q}$ is the set of a class of *minimal* exponential family approximations:

$$q(\mathbf{w}) := h(\mathbf{w}) \exp\Big[\boldsymbol{\lambda}^\top \boldsymbol{\phi}(\mathbf{w}) - A(\boldsymbol{\lambda})\Big] \tag{4}$$

where $\boldsymbol{\lambda} \in \Omega$ is a natural parameter with Ω being the set of valid natural parameterization, $\boldsymbol{\phi}(\mathbf{w})$ is a vector containing sufficient statistics, $A(\mathbf{w})$ is the log-partition function, and $h(\mathbf{w})$ is the base measure. Our result holds in more general setting where $\mathcal{Q}$ can be a class of mixture distribution as well as kernel exponential family, but minimal exponential families are sufficient to demonstrate the generality of our Bayesian principle.

5 Bayesian learning rule

Given a learning problem, such as (1), and an exponential-family approximation with a constant base measure $h(\mathbf{w}) \equiv 1$, such as (4), a variety of learning algorithms can be obtained as special cases of the following update, which we call the *Bayesian learning rule*:

$$\boldsymbol{\lambda}_{t+1} \leftarrow (1-\rho_t)\boldsymbol{\lambda}_t - \rho_t \nabla_{\boldsymbol{\mu}} \mathbb{E}_{q_t(\mathbf{w})}\left[\bar{\ell}(\mathbf{w})\right], \tag{5}$$

where t denotes the t'th iteration, $\boldsymbol{\mu} := \mathbb{E}_q[\boldsymbol{\phi}(\mathbf{w})]$ is the *expectation parameter* of the exponential family $q(\mathbf{w})$, $q_t(\mathbf{w})$ denotes $q(\mathbf{w})$ with natural parameters $\boldsymbol{\lambda}_t$, and $\rho_t > 0$ is a sequence of scalar step-sizes (learning rates). When the base measure is not a constant, it needs to be accounted in the rule, in which case, the rule takes the following form:

$$\boldsymbol{\lambda}_{t+1} \leftarrow (1-\rho_t)\boldsymbol{\lambda}_t - \rho_t \nabla_{\boldsymbol{\mu}} \mathbb{E}_{q_t(\mathbf{w})}\left[\bar{\ell}(\mathbf{w}) + \log h(\mathbf{w})\right], \tag{6}$$

The rule is derived using ideas from information geometry, and is essentially a mirror descent algorithm where the geometry is dictated by the log-partition function $A(\boldsymbol{\lambda})$ of the approximation (4) (originally derived in (Khan and Lin, 2017) for nonconjugate variational inference; see the reference for a detailed derivation). The Bayesian learning rule converges to the solution of the Bayesian learning problem (2).

Even though the algorithm is designed to optimize the Bayesian learning problem, with an appropriate choice of the approximation $q(\mathbf{w})$, it reduces to learning algorithms that optimize the ERM loss (1). This is a surprising and remarkable result, and similar results of this type show that the Bayesian principle indeed is a fundamental principle behind a variety of learning algorithms.

6 Illustrative example (least-squares)

The simplest learning problem to illustrate this point is linear regression where the Bayesian learning rule reduces to ridge regression. The loss function is given as follows:

$$\bar{\ell}(\mathbf{w}) := \tfrac{1}{2}\left[\sum_{i=1}^{N}(y_i - \mathbf{x}_i^\top \mathbf{w})^2 + \delta \mathbf{w}^\top \mathbf{w}\right], \tag{7}$$

with $\delta > 0$ as a scalar regularization parameter. The minimizer is $\mathbf{w}_* := (\mathbf{X}^\top\mathbf{X} + \delta\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$, where $\mathbf{X}$ is a $N \times D$ matrix with $\mathbf{x}_i^\top$ as rows and $\mathbf{y}$ is the vector of all y_i. We can derive this from Bayesian principle by choosing $q(\mathbf{w}) := \mathcal{N}(\mathbf{w}|\mathbf{m}, \mathbf{S}^{-1})$ with mean $\mathbf{m}$ and precision matrix $\mathbf{S}$ (i.e., inverse of the covariance) and estimate its parameters using Bayesian learning rule. For Gaussians, there are two natural parameters and corresponding expectation parameters.

$$\boldsymbol{\lambda} := \begin{bmatrix} \mathbf{S}\mathbf{m} \\ -\frac{1}{2}\mathbf{S} \end{bmatrix} \quad \boldsymbol{\phi}(\mathbf{w}) := \begin{bmatrix} \mathbf{w} \\ \mathbf{w}\mathbf{w}^\top \end{bmatrix}, \quad \boldsymbol{\mu} = \mathbb{E}_q \begin{bmatrix} \mathbf{w} \\ \mathbf{w}\mathbf{w}^\top \end{bmatrix} = \begin{bmatrix} \mathbf{m} \\ \mathbf{S}^{-1} + \mathbf{m}\mathbf{m}^\top \end{bmatrix} \tag{8}$$

There is a one-to-one mapping between the natural ane expectation parameters, and as we will see later the two parameters lives in separates spaces that are duals of each other. The Bayesian learning rule

requires the gradient of $\mathbb{E}_q[\bar{\ell}(\mathbf{w})]$ with respect to $\boldsymbol{\mu}$. This can be easily obtained by noticing that the expectated loss is in fact linear in $\boldsymbol{\mu}$:

$$\mathbb{E}_q\left[\bar{\ell}(\mathbf{w})\right] = -\mathbf{y}^\top\mathbf{X}\mathbb{E}_q\left(\mathbf{w}\right) + \tfrac{1}{2}\mathrm{Tr}\left[\left(\mathbf{X}^\top\mathbf{X} + \delta\mathbf{I}\right)\mathbb{E}_q\left(\mathbf{w}\mathbf{w}^\top\right)\right] + \mathrm{cnst}. \tag{9}$$

Therefore the gradients are simply given as follows:

$$\nabla_{\boldsymbol{\mu}}\mathbb{E}_q\left[\bar{\ell}(\mathbf{w})\right] = \begin{bmatrix} -\mathbf{X}^\top\mathbf{y} \\ \frac{1}{2}\left(\mathbf{X}^\top\mathbf{X} + \delta\mathbf{I}\right) \end{bmatrix}. \tag{10}$$

The right-hand side above already contains the components requires to obtain $\mathbf{w}_*$. Since the gradient does not depend on $\boldsymbol{\lambda}$, we can use a learning-rate $\rho_t = 1$ and set $\boldsymbol{\lambda}_* = -\nabla_{\boldsymbol{\mu}}\mathbb{E}_q\left[\bar{\ell}(\mathbf{w})\right]$ to get the following solution:

$$\mathbf{m}_* := \left(\mathbf{X}^\top\mathbf{X} + \delta\mathbf{I}\right)^{-1}\mathbf{X}^\top\mathbf{y}, \qquad \mathbf{S}_* := \left(\mathbf{X}^\top\mathbf{X} + \delta\mathbf{I}\right), \tag{11}$$

As promised, the mean $\mathbf{m}_*$ is equal to $\mathbf{w}_*$ recovering the ridge regression solution.

7 Bayesian inference on Conjugate models

This simple examples illustrates the main idea of the Bayesian learning rule. The loss (7) is quadratic and therefore is linear in sufficient statistics of q, making the expected loss be linear in $\boldsymbol{\mu}$. In such cases, the gradient is is independent of $\boldsymbol{\lambda}$, and the solution can be obtained in closed-form by setting $\rho_t = 1$. This generalizes to all loss functions that correspond to the log-likelihood of a *conjugate* exponential-family model. In such cases, the loss can be written as $\bar{\ell}(\mathbf{w}) = \boldsymbol{\lambda}_{\mathcal{D}}^\top\boldsymbol{\phi}(\mathbf{w})$ for some $\boldsymbol{\lambda}_{\mathcal{D}}$ which only depends on the dataset $\mathcal{D}$ and is independent of $\mathbf{w}$. For example, for the least-square case, we have,

$$\boldsymbol{\lambda}_{\mathcal{D}} = \begin{bmatrix} -\mathbf{X}^\top\mathbf{y} \\ \frac{1}{2}\left(\mathbf{X}^\top\mathbf{X} + \delta\mathbf{I}\right) \end{bmatrix}. \tag{12}$$

The solution is then recovered by setting $\boldsymbol{\lambda}_* = \boldsymbol{\lambda}_{\mathcal{D}}$, which then corresponds to the classical *forward-backward* algorithm (Koller and Friedman, 2009).

8 Gradient-Descent from Bayesian principles

Gradient-based algorithms, such as gradient descent (GD) and Newton's method, are perhaps one of the most popular algorithms for unconstrained learning problems. We can derive both of these algorithms from the Bayesian learning rule by choosing $q(\mathbf{w})$ to be Gaussian. The GD update takes the following form:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \rho_t\nabla_{\mathbf{w}}\bar{\ell}(\mathbf{w}_t) \tag{13}$$

To derive this algorithm from Bayesian learning rule, we let $q(\mathbf{w}) := \mathcal{N}(\mathbf{m}, \mathbf{I})$ where $\mathbf{m}$ is the parameter that needs to be estimate and the covariance is set to $\mathbf{I}$ (the choice of covariance as well as prior can be arbitrary and does not change the form of the update). For this choice of q, both the natural and expectation parameters are equal to the mean $\mathbf{m}$, i.e., $\boldsymbol{\lambda} := \mathbf{m}$ and $\boldsymbol{\mu} := \mathbb{E}_q(\mathbf{w}) = \mathbf{m}$. The base measure is $h(\mathbf{w}) := (2\pi)^{-D/2}\exp\left(-\frac{1}{2}\mathbf{w}^\top\mathbf{w}\right)$, which is not equal to 1. Therefore, we need to use the update (6):

$$\boldsymbol{\lambda}_{t+1} \leftarrow (1-\rho_t)\boldsymbol{\lambda}_t - \rho_t\nabla_{\boldsymbol{\mu}}\mathbb{E}_{q_t(\mathbf{w})}\left[\bar{\ell}(\mathbf{w}) + \log h(\mathbf{w})\right]. \tag{14}$$

Now, by simply substituting the definition of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ and noting that the derivative of $\mathbb{E}_q[h(\mathbf{w})]$ is equal to $-\mathbf{m}$, we get the following update:

$$\mathbf{m}_{t+1} \leftarrow \mathbf{m}_t - \rho_t \nabla_{\mathbf{m}} \mathbb{E}_{q_t(\mathbf{w})} \left[\bar{\ell}(\mathbf{w}) \right], \tag{15}$$

which is similar to the GD update but now the gradients are obtained at the expectation $\mathbb{E}_{q(\mathbf{w})} \left[\bar{\ell}(\mathbf{w}) \right]$ instead of the loss itself. To recover the GD update, we can approximate the gradient at the mean $\mathbf{m}$:

$$\nabla_{\mathbf{m}} \mathbb{E}_{q(\mathbf{w})} \left[\bar{\ell}(\mathbf{w}) \right] \approx \nabla_{\mathbf{w}} \bar{\ell}(\mathbf{w}) \big|_{\mathbf{w}=\mathbf{m}} . \tag{16}$$

This approximation, also known as the zeroth-order delta approximation, implies that we resort to a greedy approximation that does not employ the averaging property of the Bayesian principle. Essentially, by giving away the Bayesian principles, we obtain a non-Bayesian one. Using this approximation and then referring to the mean $\mathbf{m}_t$ as the iterate $\mathbf{w}_t$, we recover the gradient descent update (13).

9 Newton's method from Bayesian principles

The above example illustrates one of the key ideas behind the Bayesian learning rule – the gradient $\nabla_{\boldsymbol{\mu}} \mathbb{E}_{q(\mathbf{w})}[\bar{\ell}(\mathbf{w})]$ enables us to probe gradient information of the $\bar{\ell}(\mathbf{w})$. The gradient with respect to the mean $\mathbf{m}$ of the Gaussian, we get first-order information about the $\bar{\ell}(\mathbf{w})$. In general, by increasing the complexity of the approximating distribution, we can dig-out higher-order information about $\bar{\ell}(\mathbf{w})$. In fact, if we use a Gaussian approximation $\mathcal{N}(\mathbf{w}|\mathbf{m}, \mathbf{S}^{-1})$ and estimate both its mean $\mathbf{m}$ and precision $\mathbf{S}$ (i.e., inverse of the covariance), then we recover an online version of Newton's method, as shown in (Khan et al., 2018):

$$\mathbf{m}_{t+1} \leftarrow \mathbf{m}_t - \rho_t \mathbf{S}_{t+1}^{-1} \mathbb{E}_{q_t} \left[\nabla_{\mathbf{w}} \bar{\ell}(\mathbf{w}_t) \right], \qquad \text{where } \mathbf{S}_{t+1} \leftarrow (1-\rho_t)\mathbf{S}_t + \rho_t \mathbb{E}_{q_t} \left[\nabla^2_{\mathbf{ww}} \bar{\ell}(\mathbf{w}_t) \right]. \tag{17}$$

where $q_t := \mathcal{N}(\mathbf{w}|\mathbf{m}_t, \mathbf{S}_t)$. The update above uses gradients and Hessians that are evaluated at samples from q_t. By making the approximation (16) again and denoting $\mathbf{m}_t$ by $\mathbf{w}_t$, we get the following update that resorts to a greedy approximation, rather than the Bayseian one:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \rho_t \mathbf{S}_{t+1}^{-1} \left[\nabla_{\mathbf{w}} \bar{\ell}(\mathbf{w}_t) \right], \qquad \text{where } \mathbf{S}_{t+1} \leftarrow (1-\rho_t)\mathbf{S}_t + \rho_t \nabla^2_{\mathbf{ww}} \bar{\ell}(\mathbf{w}_t). \tag{18}$$

The precision $\mathbf{S}_t$ contains a moving average of the past Hessians. Newton's method can be obtained as a special case where the learning-rate is set to be 1, which corresponds to a perfect Newton step:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \left[\nabla^2_{\mathbf{ww}} \bar{\ell}(\mathbf{w}_t) \right]^{-1} \left[\nabla_{\mathbf{w}} \bar{\ell}(\mathbf{w}_t) \right], \tag{19}$$

This is valid when the loss is strongly convex and the algorithm is initialized close to the solution. When the updates employ stochastic gradients, we should resort to a step size of less than 1 since we do not have correct second-order information. In such cases, the update resembles RMSprop algorithm that is popular in deep learning. This is shown in Khan et al. (2018).

10 Complexity of approximation and learning-algorithm

The above two examples illsutrate a strong link between the complexity of the approximation q and that of the learning algorithms. The Gaussian approximation with unknown mean parameter results in a first-order method, while adding the covariance parameter results in a second-order method. By increasing the number of parameters of q or equivalently a richer sufficient statistics, we can therefore

obtain learning-algorithm that use higher-order information. A detailed discussion will be added in the future version where we will show that, by using kernel exponential families where natural parameters lie in a Reproducing-Kernel Hilbert Space (RKHS), arbitrarily complex optimization algorithms can be obtained. Bayesian principles therefore carry an immense potential to enable researchers invent new optimization algorithms that go beyond first and second-order algorithms.

References

Bissiri, P. G., Holmes, C. C., and Walker, S. G. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130.

Khan, M., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. (2018). Fast and scalable Bayesian deep learning by weight-perturbation in Adam. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2611–2620, Stockholmsmässan, Stockholm Sweden. PMLR.

Khan, M. E. and Lin, W. (2017). Conjugate-computation variational inference: converting variational inference in non-conjugate models to inferences in conjugate models. In *International Conference on Artificial Intelligence and Statistics*, pages 878–887.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Zellner, A. (1988). Optimal information processing and Bayes's theorem. *The American Statistician*, 42(4):278–280.