physical sciences forum

MDPI

Proceeding Paper

Variational Bayesian Approximation (VBA): A Comparison between Three Optimization Algorithms †

Seyedeh Azadeh Fallah Mortezanejad [1] and Ali Mohammad-Djafari [2,3,*]

[1] School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China
[2] International Science Consulting and Training (ISCT), 91440 Bures sur Yvette, France
[3] Shanfeng Company, Shaoxing 312352, China
* Correspondence: djafari@ieee.org
† Presented at the 41st International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Paris, France, 18–22 July 2022.

Abstract: In many Bayesian computations, we first obtain the expression of the joint distribution of all the unknown variables given the observed data. In general, this expression is not separable in those variables. Thus, obtaining the marginals for each variable and computing the expectations is difficult and costly. This problem becomes even more difficult in high dimensional quandaries, which is an important issue in inverse problems. We may then try to propose a surrogate expression with which we can carry out approximate computations. Often, a separable expression approximation can be useful enough. The variational Bayesian approximation (VBA) is a technique that approximates the joint distribution p with an easier, for example separable, distribution q by minimizing the Kullback–Leibler divergence $KL(q|p)$. When q is separable in all the variables, the approximation is also called the mean field approximation (MFA), and so q is the product of the approximated marginals. A first standard and general algorithm is the alternate optimization of $KL(q|p)$ with respect to q_i. A second general approach is its optimization in the Riemannian manifold. However, in this paper, for practical reasons, we consider the case where p is in the exponential family and so is q. For this case, $KL(q|p)$ becomes a function of the parameters θ of the exponential family. Then, we can use any other optimization algorithm to obtain those parameters. In this paper, we compare three optimization algorithms, namely a standard alternate optimization, a gradient-based algorithm and a natural gradient algorithm, and study their relative performances in three examples.

Keywords: variational Bayesian approach (VBA); Kullback–Leibler divergence; mean field approximation (MFA); optimization algorithm

Citation: Fallah Mortezanejad, S.A.; Mohammad-Djafari, A. Variational Bayesian Approximation (VBA): A Comparison between Three Optimization Algorithms. *Phys. Sci. Forum* **2022**, *5*, 48. https://doi.org/10.3390/psf2022005048

Academic Editors: Frédéric Barbaresco, Frank Nielsen and Martino Trassinelli

Published: 8 February 2023

1. Introduction

In many applications, with direct or indirect observations, the use of the Bayesian computations starts with obtaining the expression of the joint distribution of all the unknown variables given the observed data. Then, we must use it for inference. In general, this expression is not separable in all the variables of the problem. So, the computations become hard and costly. For example, obtaining the marginals for each variable and computing the expectations are difficult and costly. This problem becomes even more crucial in high dimensional quandaries, which is an important issue in inverse problems. We may then need to propose a surrogate expression with which we can carry out approximate computations.

The variational Bayesian approximation (VBA) is a technique that approximates the joint distribution p with an easier, for example a separable one, q, by minimizing the Kullback–Leibler divergence $KL(q|p)$, which makes the marginal computations much easier. For example, in the case of two variables, $p(x, y)$ is approximated by $q(x, y) = q_1(x)q_2(y)$ via minimizing $KL(q_1q_2|p)$. When q is separable in all the variables of p, the approximation is also called mean field approximation (MFA).

To obtain the approximate marginals q_1 and q_2, we have to minimize $KL(q_1q_2|p)$. The first standard and general algorithm is the alternate optimization of $KL(q_1q_2|p)$ with respect to q_1 and q_2. By finding the expression of the functional derivatives of $KL(q_1q_2|p)$ with respect to q_1 and q_2 and then equating them to zero alternatively, we obtain an iterative optimization algorithm. A second general approach is its optimization in the Riemannian manifold. However, in this paper, for practical reasons, we consider the case where p is in the exponential family and so are q_1 and q_2. For this case, $KL(q_1q_2|p)$ becomes a function of the parameters θ of the exponential family. Then, we can use any other optimization algorithm to obtain those parameters.

In this paper, we compare three optimization algorithms: a standard alternate optimization (Algorithm 1), a gradient-based algorithm [1,2] (Algorithm 2) and a natural gradient algorithm [3–5] (Algorithm 3). The aim of this paper is to consider the first algorithm as the VBA method and compare it with the two other algorithms.

Of the main advantages of the VBA for inference problems, such as inverse problems and machine learning, we can mention the following:

- First, VBA builds a sufficient model according to prior information and the final posterior distribution. Especially in the mean field approximation (MFA), the result ends in an explicit form for each unknown component using conjugate priors and works well for small sample sizes [6–8].
- The second benefit is, for example in machine learning, that it is a robust way for classification based on the predictive posterior distribution and diminishes over-trained parameters [7].
- The third privilege is that the target structure has uncertainty in the VBA recursive processes. This feature prevents further error propagation and increases the robustness of VBA [9].

Besides all these preponderances, the VBA has some weaknesses, such as difficulty regarding the solution of integrals and expectations in terms of obtaining a posterior distribution, and there is no evidence of finding an exact posterior [6]. Its most significant drawback arises when there are strong dependencies between unknown parameters, and the VBA ignores them. Then estimates, computed based on this approximation, may be very far from the exact values. However, it works well when the number of dependencies are low [8].

In this article, we examine three different estimating algorithms of the unknown parameters in a model concerning prior information. The first iterative algorithm is a standard alternate optimization based on VBA, which begins a certain initial points. Sometimes, the points are estimated from an available dataset, but most of the time, we do not have enough data on the parameters to make certain pre-estimations of them. To solve this obstacle, we can start the algorithm with certain desired points, and then by repeating the process, they approach the true values using the posterior distribution. The second two algorithms are gradient-based and natural gradient algorithms, whose base function is Kullback–Leibler divergence. First, the gradient of Kullback–Leibler for all unknown parameters must be found, then must start from points either estimated from data or desired choices. Then, we repeat the iterative algorithm until it converges to certain points. If we denote the unknown parameter space with θ, then the recursive formula is $\tilde{\theta}^{(k+1)} = \tilde{\theta}^{(k)} - \gamma \nabla KL(\tilde{\theta}^{(k)})$ for gradient-based and natural gradient algorithms with different values of γ.

Additionally, we consider three examples, normal-inverse-gamma, multivariate normal and linear inverse problem for checking the performance and convergence speed of the algorithms.

We propose the following organization of this paper: In Section 2, we present a brief explanation of the basic analytical aspect of VBA. In Section 3, we explain our first example related to normal-inverse-gamma distribution analytically and, in practice, explain the outcomes of three algorithms to estimate the unknown parameters. In Section 5, we study a more complex example of a multivariate normal distribution whose means and variance–covariance matrix are unknown and have normal-inverse-Wishart distribution.

The aim of this section is to demonstrate the marginal distributions of $\tilde{\mu}$ and $\tilde{\Sigma}$ using a set of multivariate normal observations using the mean and variance. In Section 6, the example is closer to realistic situations and is a linear inverse problem. In Section 7, we present a summary of the work carried out in the article and compare the three recursive algorithms through three different examples.

2. Variational Bayesian Approach (VBA)

As we mentioned previously, VBA uses Kullback–Leibler divergence. Kullback–Leibler divergence [10] $KL(q|p)$ is an information measure of discrepancy between two probability functions defined as follows. Let $p(x)$ and $q(x)$ be two density functions of a continuous random variable x with respect to support set $\mathbb{S}_X$. $KL(q|p)$ function is introduced as:

$$KL(q|p) = \int_{x\in\mathbb{S}_X} q(x) \ln \frac{q(x)}{p(x)} \mathrm{d}x. \tag{1}$$

For simplicity, we assume a bivariate case of distribution $p(x,y)$ and want to assess it via VBA; therefore, we have:

$$KL(q|p) = -H(q_1) - H(q_2) - \langle \ln\ p(x,y)\rangle_{q_1q_2}, \tag{2}$$

where

$$H(q_1) = -\int_{x\in\mathbb{S}_X} q_1(x) \ln\ q_1(x)\mathrm{d}x \quad \text{and} \quad H(q_2) = -\int_{y\in\mathbb{S}_Y} q_2(y) \ln\ q_2(y)\mathrm{d}y$$

are, respectively, the Shannon entropies of x and of y, and

$$\langle \ln p(x,y)\rangle_{q_1q_2} = \int\int_{(x,y)\in\mathbb{S}_{XY}} q_1(x)q_2(y) \ln\ p(x,y)\mathrm{d}x\mathrm{d}y.$$

Now, differentiating the Equation (2) with respect to q_1 and then with respect to q_2 and equating them to zero, we obtain:

$$q_1(x) \propto \exp\Big\{\langle \ln\ p(x,y)\rangle_{q_2(y)}\Big\} \quad \text{and} \quad q_2(y) \propto \exp\Big\{\langle \ln\ p(x,y)\rangle_{q_1(x)}\Big\} \tag{3}$$

These results can be easily extended to more dimensions [11]. They do not have any closed form because they depend on the expression of $p(x,y)$ and that of q_1 and q_2. An interesting case is that of exponential families and conjugate priors, where writing

$$p(x,y) = p(x|y)p(y), \text{ and } p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x|y)p(y)}{p(x)}, \tag{4}$$

we can consider $p(y)$ as prior, $p(x|y)$ as the likelihood, and $p(y|x)$ as the posterior distributions. Then, if $p(y)$ is a conjugate prior for the likelihood $p(x|y)$, then the posterior $p(y|x)$ will be in the same family as the prior $p(y)$. To illustrate all these properties, we provide details of these expressions for a first simple example of normal-inverse-gamma $p(x,y) = \mathcal{N}(x|\mu,y)\mathcal{IG}(y|\alpha,\beta)$ with $q_1(x) = \mathcal{N}(x|\mu,v)$ and $q_2 = \mathcal{IG}(y|\alpha,\beta)$. For this simple case, first we give the expression of $KL(q|p)$ with $q_1(x) = \mathcal{N}(x|\tilde{\mu},\tilde{v})$ and $q_2(y) = \mathcal{IG}(y|\tilde{\alpha},\tilde{\beta})$ as a function of the parameters $\theta = (\tilde{\mu},\tilde{v},\tilde{\alpha},\tilde{\beta})$ and then the expressions of the three above-mentioned algorithms; after which, we study their convergence.

3. Normal-Inverse-Gamma Distribution Example

The purpose of this section is to explain in detail the process of performing calculations in VBA. For this we consider a simple case for which we have all the necessary expressions. The objective here is to compare the three different algorithms mentioned above. Additionally, its practical application can be explained as follows:

We have a sensor which measures a quantity X, N times $x_1, \ldots, x_N$. We want to model these data. In a first step, we model them as $N(x|\mu, v)$ with fixed μ and v. Then, it is easy to estimate the parameters (μ, v) either by maximal likelihood or Bayesian strategy. If we assume that the model is Gaussian with unknown variance and call this variance y and assign an $\mathcal{IG}$ prior to it, then we have a model $\mathcal{NIG}$ for $p(x, y)$. The $\mathcal{NIG}$ priors were applied to the wavelet context with correlated structures because they were conjugated with normal priors [12]. We chose the normal-inverse-gamma distribution because of this conjugated property and ease of handling.

We showed that the margins are St and $\mathcal{IG}$. Working directly with St is difficult. So, we want to approximate it with a Gaussian $q_1(x)$. This is equivalent to approximating $p(x, y)$ with $q_1(x)q_2(y)$. Now, we want to find the parameters μ, v, α, and β, which minimize $KL(q_1q_2|p)$. This process is called VBA. Then, we want to compare three algorithms to obtain the parameters which minimize $KL(\cdot|\cdot)$. $KL(\cdot|\cdot)$ is convex with respect to q_1 if q_2 is fixed and is convex with respect to q_2 if q_1 is fixed. So, we hope that the iterative algorithm converges. However, $KL(\cdot|\cdot)$ may not be convex in the space of parameters. So, we have to study the shape of this criterion concerning the parameters $\tilde{v}$, $\tilde{\alpha}$ and $\tilde{\beta}$.

The practical problem considered here is the following: A sensors delivers a few samples $x = \{x_1, x_2, \cdots, x_N\}$ of a physical quantity X. We want to find $p(x)$. For this process, we assume a simple Gaussian model but with unknown variance y. Thus, the forward model can be written as $p(x, y) = \mathcal{N}(x|\mu, y)\mathcal{IG}(y|\alpha, \beta)$. In this simple example, we know that $p(x)$ is Student's t-distribution obtained by:

$$S(x|\mu, \alpha, \beta) = \int \mathcal{N}(x|\mu, y)\mathcal{IG}(y|\alpha, \beta)\mathrm{d}y \tag{5}$$

Our objective is to find the three parameters $\theta = (\mu, \alpha, \beta)$ from the data x and an approximate marginal $q(x)$ for $p(x)$.

The main idea is to find such $q_1(x)q_2(y)$ as an approximation of $p(x, y)$. Here, we show the VBA, step by step. For this, we start by choosing the conjugate families $q_1(x) = \mathcal{N}(x|\tilde{\mu}, \tilde{v})$ and $q_2(y) = \mathcal{IG}(y|\tilde{\alpha}, \tilde{\beta})$.

In the first step, we have to calculate $\ln p(x, y)$

$$\ln p(x, y) = c - \frac{1}{2}\ln y - \frac{1}{2y}(x - \tilde{\mu})^2 - (\tilde{\alpha} + \frac{1}{2})\ln y - \frac{\tilde{\beta}}{y}. \tag{6}$$

where c is a constant value term independent of x and y. First of all, to use the iterative algorithm given in (3), starting by $q_1 = N(x|\mu', v')$ we have to find $q_2(y)$, so we have to start by finding $q_2(y)$. The integration of $\ln p(x, y)$ concerns $q_1(x)$

$$\langle \ln\ p(x, y)\rangle_{q_1} = c - \frac{1}{2y}\langle (x - \tilde{\mu})^2\rangle_{q_1} - (\tilde{\alpha} + 1)\ln\ y - \frac{\tilde{\beta}}{y}. \tag{7}$$

Since the mean of x is the same in prior and posterior distribution, $\tilde{\mu} = \tilde{\mu}'$ then $\langle (x - \tilde{\mu})^2\rangle_{q_1} = \tilde{v}$, otherwise $\langle (x - \tilde{\mu})^2\rangle_{q_1} = \tilde{v} + (\tilde{\mu} - \tilde{\mu}')^2$. Thus when $\tilde{\mu} = \tilde{\mu}'$

$$q_2(y) \propto \exp[-(\tilde{\alpha} + 1)\ln\ y - (\frac{\tilde{v}}{2} + \tilde{\beta})\frac{1}{y}]. \tag{8}$$

Thus, the function $q_2(y)$ is equivalent to an inverse gamma distribution $\mathcal{IG}(\tilde{\alpha}, \frac{\tilde{v}}{2} + \tilde{\beta})$. Similarly, $q_2(y)$ is $\mathcal{IG}(\tilde{\alpha}, \frac{\tilde{v} + (\tilde{\mu} - \tilde{\mu}')^2}{2} + \tilde{\beta})$ when $\tilde{\mu} \neq \tilde{\mu}'$. We have to take integral of $\ln\ p(x, y)$ over q_2 to find q_1

$$\langle \ln\ p(x, y)\rangle_{q_2} = c - (\tilde{\alpha} + 1)\langle \ln\ y\rangle_{q_2} - (\tilde{\beta} + \frac{1}{2}(x - \tilde{\mu})^2)\langle \frac{1}{y}\rangle_{q_2} \tag{9}$$

Note that the first term does not depend on x and $\langle\frac{1}{y}\rangle_{q_2} = \frac{2\tilde{\alpha}}{2\tilde{\beta}+\tilde{v}}$, so

$$q_1(x) \propto \exp[-\frac{2\tilde{\alpha}}{2\tilde{\beta}+\tilde{v}}(\tilde{\beta}+\frac{1}{2}(x-\tilde{\mu})^2)] \propto \exp[-\frac{(x-\tilde{\mu})^2}{2\frac{2\tilde{\beta}+\tilde{v}}{2\tilde{\alpha}}}]. \tag{10}$$

We see that q_1 is, again, a normal distribution but with updated parameters $\mathcal{N}(\tilde{\mu}, \frac{2\tilde{\beta}+\tilde{v}}{2\tilde{\alpha}})$, so $\tilde{v} = \frac{2\tilde{\beta}+\tilde{v}}{2\tilde{\alpha}}$. Note that we obtained the conjugacy property: if $p(x|y) = \mathcal{N}(x|\mu, y)$ and $p(y) = \mathcal{IG}(y|\alpha, \beta)$, then $p(y|x) = \mathcal{IG}(y|\alpha', \beta')$ where μ', α' and β' are $\mu' = \mu$, $\alpha' = \alpha$, $\beta' = \beta + \frac{2\beta+v}{2\alpha}$. In this case, we also know that $p(x|\alpha, \beta) = St(x|\mu', \alpha, \beta)$.

In standard alternate optimization based on VBA (Algorithm 1), there is no need for an iterative process for $\tilde{\mu}$ and $\tilde{\alpha}$, which are approximated by $\tilde{\mu} = \mu_0$ and $\tilde{\alpha} = \alpha_0$, respectively. The situation for $\tilde{\beta}$ and $\tilde{v}$ is different because there are circular dependencies among them. So, the approximation needs an iterative process, staring from $\tilde{\mu}^{(1)} = \mu_0$, $\tilde{v}^{(1)} = v_0$, $\alpha^{(1)} = \alpha_0$ and $\beta^{(1)} = \beta_0$. As a conclusion for this case is that the values of α and μ do not change during the iterations, and so depend on the initial values. However, the values of β and v are interdependent and change during the iterations. This algorithm is summarized below.

Alternate optimization algorithm based on VBA (Algorithm 1):

$$\begin{array}{rcl} \tilde{\alpha}^{(k+1)} & = & \tilde{\alpha}^{(k)}, \\ \tilde{\beta}^{(k+1)} & = & \tilde{\beta}^{(k)} + \frac{\tilde{v}^{(k)}}{2}, \\ \tilde{\mu}^{(k+1)} & = & \tilde{\mu}^{(k)}, \\ \tilde{v}^{(k+1)} & = & \frac{2\tilde{\beta}^{(k)}+\tilde{v}^{(k)}}{2\tilde{\alpha}^{(k)}}. \end{array}$$

This algorithm converges to $v = (2\tilde{\beta}+\tilde{v})/(2\tilde{\alpha})$, which gives $\tilde{v} = (2\tilde{\beta})/(2\tilde{\alpha}-1)$ and $\tilde{\beta} = 0$, so $\tilde{v} = 0$, which is a degenrate solution.

The two other algorithms, gradient- and natural gradient-based, require to find the expression of $KL(q_1q_2 : p)$ as a function of the parameters $\theta = (\alpha, \beta, \mu, v)$:

$$KL(\tilde{\theta}) = 2\ln\,\Gamma(\tilde{\alpha}-\frac{1}{2}) - (2\tilde{\alpha}+\frac{3}{2})\psi_0(\tilde{\alpha}-\frac{1}{2}) + \frac{(2\tilde{\alpha}-1)(\tilde{v}+2\tilde{\beta})}{4\tilde{\beta}} + \tilde{\alpha} + \frac{5}{2}\ln\tilde{\beta} - \frac{1}{2}\ln\,\tilde{v} - 1. \tag{11}$$

We also need the gradient expression of $\nabla KL(\tilde{\theta})$ for $\tilde{\theta}$:

$$\nabla KL(\tilde{\theta}) = \left(\frac{\tilde{v}}{2\tilde{\beta}} - (2\tilde{\alpha}+\frac{3}{2})\psi_1(\tilde{\alpha}-\frac{1}{2}) + 2, \quad -(2\tilde{\alpha}-1)\frac{\tilde{v}}{4\tilde{\beta}^2} + \frac{5}{2\tilde{\beta}}, \quad 0, \quad \frac{2\tilde{\alpha}-1}{4\tilde{\beta}} - \frac{1}{2\tilde{v}}\right). \tag{12}$$

As we can see, these expressions do not depend on $\tilde{\mu}$, so their derivatives with respect to $\tilde{\mu}$ are zero. The means are preserved.

Gradient and natural gradient algorithms (Algorithms 2 and 3):

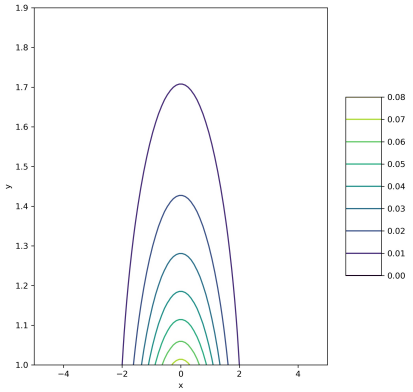
$$\begin{array}{rcl} \tilde{\alpha}^{(k+1)} & = & \tilde{\alpha}^{(k)} - \gamma\Big(\frac{\tilde{v}^{(k)}}{2\tilde{\beta}^{(k)}} - (2\tilde{\alpha}^{(k)}+\frac{3}{2})\psi_1(\tilde{\alpha}^{(k)}-\frac{1}{2}) + 2\Big), \\ \tilde{\beta}^{(k+1)} & = & \tilde{\beta}^{(k)} - \gamma\Big(-(2\tilde{\alpha}^{(k)}-1)\frac{\tilde{v}^{(k)}}{4[\tilde{\beta}^{(k)}]^2} + \frac{5}{2\tilde{\beta}^{(k)}}\Big), \\ \tilde{\mu}^{(k+1)} & = & \tilde{\mu}^{(k)}, \\ \tilde{v}^{(k+1)} & = & \tilde{v}^{(k+1)} - \gamma\Big(\frac{2\tilde{\alpha}^{(k)}-1}{4\tilde{\beta}^{(k)}} - \frac{1}{2\tilde{v}^{(k)}}\Big). \end{array}$$

Here, γ is fixed for the gradient algorithm and is proportional to $1/\|\nabla KL\|$ for the natural gradient algorithm.

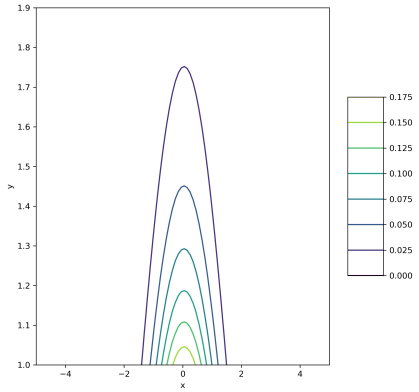
4. Numerical Experimentations

To show the relative performances of these algorithms, we generate $n = 100$ samples from the model $p(x, y) = \mathcal{N}(x|1, y)\mathcal{IG}(y|3, 1)$ for the numerical computations. Thus, it should be noted that we know the exact values of the unknown parameters which can be used to show the performances of the proposed algorithms. The following results: $\tilde{\theta}_1$, $\tilde{\theta}_2$ and $\tilde{\theta}_3$ are the estimated parameters using, respectively, Algorithm 1, Algorithm 2 and Algorithm 3. The contour plots of the corresponding probability density functions are shown in Figure 1 compared with original model.

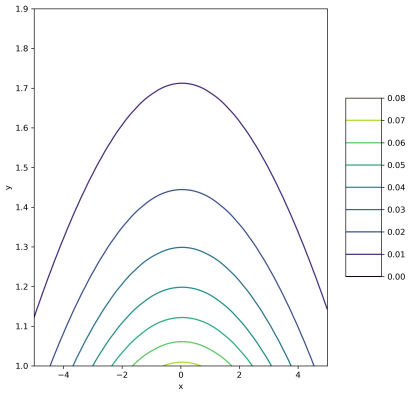
$$\tilde{\theta}_1 = \begin{cases} \tilde{\mu} = 0.044724, \\ \tilde{v} = 0.590894, \\ \tilde{\alpha} = 3.792337, \\ \tilde{\beta} = 1.765735 \end{cases} \quad \tilde{\theta}_2 = \begin{cases} \tilde{\mu} = 0.044724, \\ \tilde{v} = 7.910504, \\ \tilde{\alpha} = 4.991621, \\ \tilde{\beta} = 3.002594 \end{cases} \quad \tilde{\theta}_3 = \begin{cases} \tilde{\mu} = 0.044724, \\ \tilde{v} = 0.423761, \\ \tilde{\alpha} = 4.415239, \\ \tilde{\beta} = 0.706049. \end{cases}$$



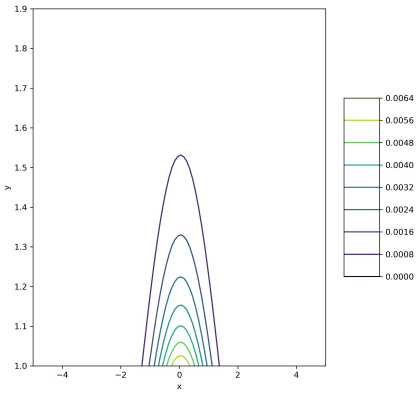
(**a**) True contour plot of the model



(**b**) The VBA approximation



(**c**) The gradient-based algorithm



(**d**) The natural gradient algorithm

Figure 1. The true model is $\mathcal{N}(x|0, y)\mathcal{IG}(y|3, 1)$. The numbers of iterations until the convergence are different in the algorithms.

All three algorithms try to minimize the same criterion. So, the objectives are all the same, but the number of steps may differ. The requirements must reach the minimum $KL(\cdot)$. In this simple example of the normal-inverse-gamma distribution, the convergence step numbers of VBA, gradient-based and natural gradient are 1, 2 and 1 using moment initializations. The overall performance of the standard alternate optimization (VBA) is

more precise than any other. The poorest estimation is from a gradient-based algorithm. So, the algorithms are able to approximate the joint density function with a separable one but with different accuracy. In the following section, we will tackle a more complex model.

5. Multivariate Normal-Inverse-Wishart Example

In previous section, we explained how to preform VBA in order to approximate a complicated joint distribution function by tractable marginal factorials using a simple case study. In this section, a multivariate normal case $p(x) = \mathcal{N}(x|\tilde{\mu}, \tilde{\Sigma})$ is considered, which is approximated by $q(x) = \prod_i \mathcal{N}(x_i|\tilde{\mu}_i, \tilde{v}_i)$ for different shapes of the covariance matrix $\tilde{\Sigma}$.

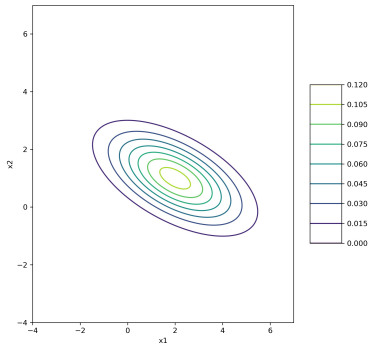
We assume that the basic structure of an available dataset is multivariate normal with unknown mean vector $\tilde{\mu}$ and variance–covariance matrix $\tilde{\Sigma}$. Their joint prior distribution is a normal-inverse-Wishart distribution of $\mathcal{NIW}(\tilde{\mu}, \tilde{\Sigma}|\tilde{\mu}_0, \tilde{\kappa}, \tilde{\Psi}, \tilde{\nu})$, which is the generalized form of the classical $\mathcal{NIG}$. The posteriors are multivariate normal for mean vector and inverse-Wishart in the variance–covariance matrix. Since the normal-inverse-Wishart distribution is a conjugate prior distribution for multivariate normal, the posterior distribution of $\tilde{\mu}$ and $\tilde{\Sigma}$ again belongs to the same family, and their corresponding margins are

$$\mathcal{MN}\left(\frac{\tilde{\kappa}\tilde{\mu}_0 + n\overline{x}}{\tilde{\kappa}+n}, \frac{1}{\tilde{\kappa}+n}\tilde{\Lambda}\right), \quad \mathcal{IW}\left(\tilde{\Lambda} + \tilde{\Psi} + \sum_{i=1}^{n}(x_i - \overline{x})(x_i - \overline{x})^T, \tilde{\nu} + n\right), \tag{13}$$

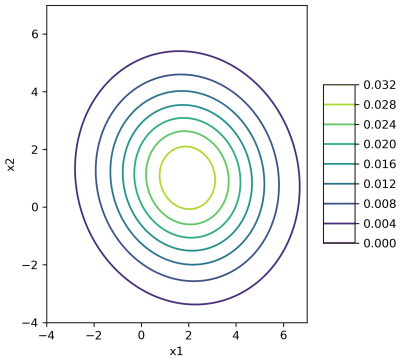
where n is the sample size. To present the performance of the three algorithms, we examined on a dataset based on $x \sim \mathcal{NIW}(x|\mu, \Sigma)$, whose parameters have the following low-density structure:

$$\mu \sim \mathcal{MN}\left(\mu|\begin{bmatrix}2\\1\end{bmatrix}, \frac{1}{2}\begin{bmatrix}3 & -1\\-1 & 1\end{bmatrix}\right), \quad \Sigma \sim \mathcal{IW}\left(\Sigma|\begin{bmatrix}3 & -1\\-1 & 1\end{bmatrix}, 6\right).$$

We used only the data of x in the estimation processes. The results of algorithms are presented in Figure 2 along with the true contour plot of the model. The VBA estimation is the most separable distribution compared with gradient and natural gradient methods. The next best case is the natural gradient algorithm, but its weakness is transferring the dependency slightly to the approximation. The results for the gradient-based algorithm show the dependency completely, along with its inability to obtain a separable model.

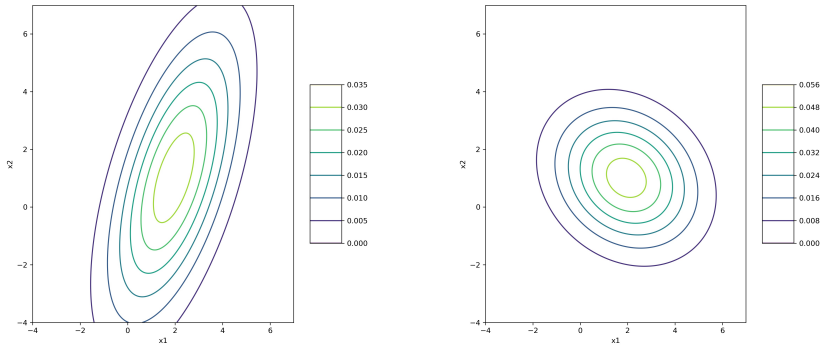


(**a**) True contour plot of the model



(**b**) The VBA approximation

Figure 2. *Cont.*



(**c**) The gradient-based algorithm (**d**) The natural gradient algorithm

Figure 2. In this example, the numbers of iterations to obtain the convergence are different between the three algorithms. These numbers are, respectively, 2, 280 and 4. The VBA and natural gradient algorithms estimate the distribution with separable functions in fewer steps than the gradient which seems not converged even up to 280 iterations.

6. Simple Linear Inverse Problem

Finally the third example is the case of linear inverse problems with $g = Hf + \epsilon$ with priors $p(\epsilon) = \mathcal{N}(\epsilon|0, v_\epsilon I)$, $p(f) = \mathcal{N}(f|0, \text{diag}[v])$ and $p(v|\alpha, \beta) = \prod_j \mathcal{IG}(v_j|\alpha, \beta)$, where $f = [f_1, f_2, \cdots, f_N]$ and $v = [v_{f_1}, v_{f_2}, \cdots, v_{f_N}]$. Using these priors, we get $p(f, v|g) \propto p(g|f, v_\epsilon)p(f|v)p(v)$ with $p(g|f, v_\epsilon) = \mathcal{N}(g|Hf, v_\epsilon I)$. See [13] for details.

Thus, the joint distribution of g, f, and v:

$$p(g, f, v) \propto p(g|f, v_\epsilon)p(f|v)p(v). \tag{14}$$

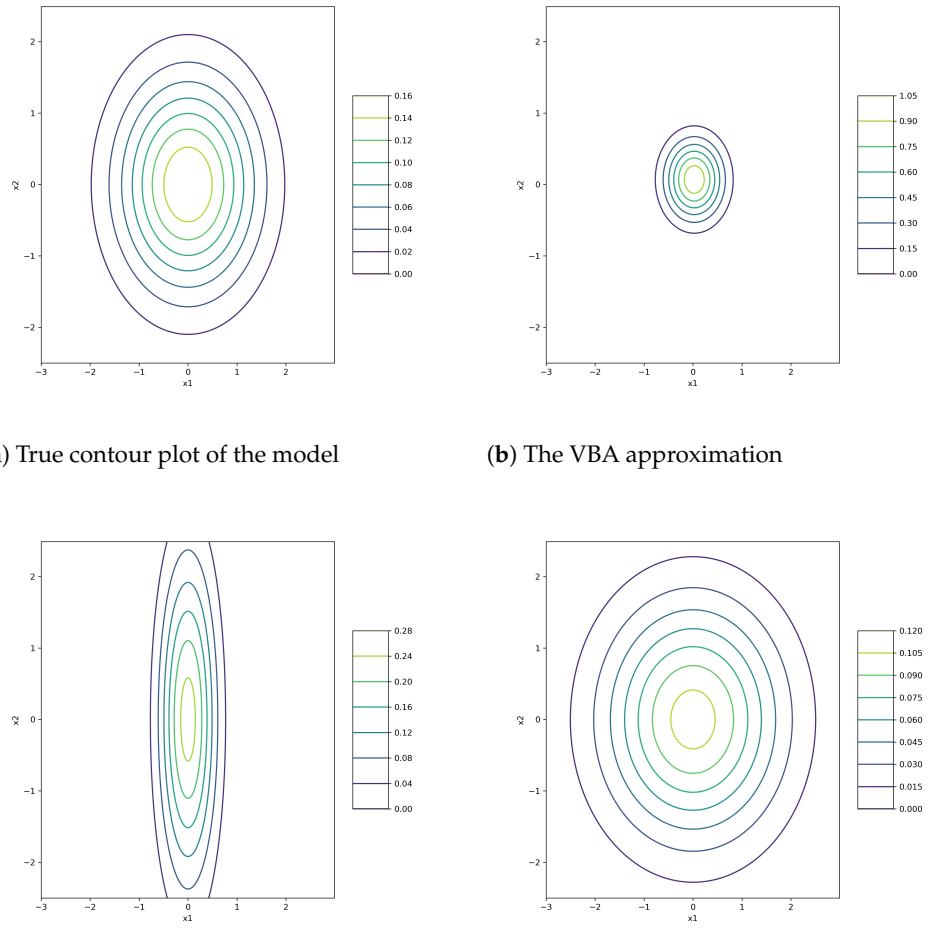
is approximated by $q(g, ft, \tilde{v}) = q_1(g|ft)q_2(ft)q_3(\tilde{v})$ using the VBA method. Even if the main interest is the estimation of $q_2(ft)$, but in the recursive process, $q_1(g|ft)$ and $q_3(\tilde{v})$ are also updated. For simplicity, we suppose that the transfer matrix H is an identical matrix I. The final outputs are as follows:

$$\begin{aligned} ft &\sim \mathcal{MN}\left(\frac{\tilde{\mu}_{ft}}{1 + \frac{2\tilde{v_\epsilon}\tilde{\alpha}}{n(\tilde{v} + \tilde{\mu}^2_{\tilde{f}} + 2\tilde{\beta})}}, \text{diag}[\frac{\tilde{v_\epsilon}(\tilde{v} + \tilde{\mu}^2_{\tilde{f}} + 2\tilde{\beta})}{n(\tilde{v} + \tilde{\mu}^2_{\tilde{f}}) + 2n\tilde{\beta} + 2\tilde{v_\epsilon}\tilde{\alpha}}]\right), \\ g &\sim \mathcal{N}(\tilde{\mu}_{ft}, \tilde{v_\epsilon} I), \quad \text{and} \quad \tilde{v}_j \sim \mathcal{IG}\left(\tilde{\alpha}_j, \frac{\tilde{v}_j + \tilde{\mu}^2_{\tilde{f}_j}}{2} + \tilde{\beta}_j\right),\ j = 1, \cdots, N. \end{aligned} \tag{15}$$

For simulations, we chose a model to examine the performance of these margins and compare them with gradient and natural gradient algorithms. The selected model is $g = Hf + \epsilon$ with the following assumptions:

$$H = I,\ f \sim \mathcal{MN}(f|0, \text{diag}[v_1, v_2]),\ v_1 \sim \mathcal{IG}(v_1|3, 2),\ v_2 \sim \mathcal{IG}(v_2|4, 3),\ \epsilon \sim \mathcal{MN}(\epsilon|0, b). \tag{16}$$

In the assessment procedure, we do not apply the above information. The output of algorithms are shown in Figure 3, as well as the actual contour plot. In this example, the best diagnosis is from the natural gradient algorithm. The VBA by construction is separable and cannot be the same as the original.



(**a**) True contour plot of the model

(**b**) The VBA approximation

(**c**) The gradient-based algorithm

(**d**) The natural gradient algorithm

Figure 3. The true model is almost separable. The natural gradient algorithm works better in this example, and the poorest approximation is that of the gradient-based algorithm.

7. Conclusions

This paper presents an approximation method of the unknown density functions for hidden variables called VBA. It is compared with gradient and natural gradient algorithms. We also consider three examples normal-inverse-gamma, normal-inverse-Wishart and linear inverse problem. We provided details of the first model and showed examples of two other examples throughout the whole paper. In all three models, the parameters are unexplored and need to be estimated by recursive algorithms. We attempted to approximate the joint complex distribution with a simpler version of the margin factorials that appeared to be independent cases. The VBA and natural gradient converged fairly early. The major discrepancy in algorithms comes from the accuracy of the results. They estimate the intricate joint distribution with separable ones. Here, the best overall performance of VBA is demonstrated.

Author Contributions: Both authors contributed equally. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Acerbi, L. Variational bayesian monte carlo. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montréal, QC, Canada, 3–8 December 2018; pp. 8223–8233.
2. Kuusela, M.; Raiko, T.; Honkela, A.; Karhunen, J. A gradient-based algorithm competitive with variational Bayesian EM for mixture of Gaussians. In Proceedings of the 2009 International Joint Conference on Neural Networks, Atlanta, GA, USA, 14–19 June 2009; pp. 1688–1695.
3. Gharsalli, L.; Duchêne, B.; Mohammad-Djafari, A.; Ayasso, H. A gradient-like variational Bayesian approach: Application to microwave imaging for breast tumor detection. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 1708–1712.
4. Zhang, G.; Sun, S.; Duvenaud, D.; Grosse, R. Noisy natural gradient as variational inference. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5852–5861.
5. Lin, W.; Khan, M.E.; Schmidt, M. Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 3992–4002.
6. Ghahramani, Z.; Beal, M. Variational inference for Bayesian mixtures of factor analysers. *Adv. Neural Inf. Process. Syst.* **1999**, *12*, 449–455.
7. Watanabe, S.; Minami, Y.; Nakamura, A.; Ueda, N. Variational Bayesian estimation and clustering for speech recognition. *IEEE Trans. Speech Audio Process.* **2004**, *12*, 365–381. [CrossRef]
8. Montesinos-López, O.A.; Montesinos-López, A.; Crossa, J.; Montesinos-López, J.C.; Luna-Vázquez, F.J.; Salinas-Ruiz, J.; Herrera-Morales, J.R.; Buenrostro-Mariscal, R. A variational Bayes genomic-enabled prediction model with genotype× environment interaction. *G3 Genes Genomes Genet.* **2017**, *7*, 1833–1853. [CrossRef] [PubMed]
9. Babacan, S.D.; Molina, R.; Katsaggelos, A.K. Variational Bayesian super resolution. *IEEE Trans. Image Process.* **2010**, *20*, 984–999. [PubMed]
10. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
11. Fox, C.W.; Roberts, S.J. A tutorial on variational Bayesian inference. *Artif. Intell. Rev.* **2012**, *38*, 85–95. [CrossRef]
12. De Canditiis, D.; Vidakovic, B. Wavelet Bayesian block shrinkage via mixtures of normal-inverse gamma priors. *J. Comput. Graph. Stat.* **2004**, *13*, 383–398. [CrossRef]
13. Ayasso, H.; Mohammad-djafari, A. Joint image restoration and segmentation using Gauss-Markov-Potts prior models and variational Bayesian computation. In Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 7–10 November 2009; pp. 1297–1300. [CrossRef]