Black-box α-divergence Minimization

José Miguel Hernández-Lobato, Yingzhen Li,
Daniel Hernández-Lobato, Thang Bui, Richard Turner,
Harvard University, University of Cambridge, Universidad Autónoma de Madrid.

Abstract

We present *black-box alpha* (BB-α), an approximate inference method based on the minimization of α-divergences between probability distributions. BB-α scales to large datasets since it can be implemented using stochastic gradient descent. BB-α can be applied to complex probabilistic models with little effort since it only requires as input the likelihood function and its gradients. These gradients can be easily obtained using automatic differentiation. By tuning the parameter α, we are able to interpolate between variational Bayes and an expectation propagation like algorithm. Experiments on probit and neural network regression problems illustrate the accuracy of the posterior approximations obtained with BB-α.

1 Introduction

Probabilistic models are very useful tools to make predictions from data. However, they require the computation of a posterior distribution which is often intractable. To avoid this, one can use approximate inference techniques. Two examples are expectation propagation (EP) and variational Bayes (VB) [1, 2]. These methods adjust the parameters of a tractable distribution so that it is close to the exact posterior. This is done by finding the stationary point of an energy function. In VB this involves solving a minimization problem. In EP this involves solving a *min-max* problem, which is more challenging. Both EP and VB are particular cases of local α-divergence minimization, where $\alpha \in (-\infty, +\infty) \setminus \{0\}$ is a parameter that specifies the divergence to be minimized [3]. If $\alpha = 1$, EP is obtained and $\alpha \to 0$ gives VB [3]. The optimal value for α may be model and/or dataset specific.

The energy function of EP (VB) includes expectations of (the logarithm of) the likelihood factors under a tractable distribution. This complicates the use of EP and VB in complex models for which there might not exist an analytic solution for these expectations. Monte Carlo sampling has been proposed to approximate the intractable expectations in VB [4]. This approach is known as *black-box* VB and it allows VB to be applied to almost any model. Here we propose a similar method for the more general problem of α-divergence minimization and call it *black-box alpha*. For this, we simplify the energy function of EP so that the *min-max* problem is transformed into a minimization problem. This enables the use of stochastic optimization methods for large scale learning. We then generalize the energy function so that it is based on the local minimization of α-divergences. Last, we show that the expectations within the resulting energy function can be approximated via Monte Carlo. This allows us to apply the resulting method to arbitrary probabilistic models. The difficulty of gradient computations is avoided by using automatic differentiation tools. The proposed method can interpolate between VB ($\alpha \to 0$) and an EP-like method ($\alpha = 1$).

2 The energy function of expectation propagation

Let us assume that we want to use EP to approximate the posterior distribution $p(\mathbf{x}|\mathcal{D})$ for some model parameters $\mathbf{x}$, where $\mathcal{D}$ is a dataset with N data points sampled independently. Then

$$p(\mathbf{x}|\mathcal{D}) \propto \left[\prod_{n=1}^{N} f_n(\mathbf{x})\right] p(\mathbf{x}) \,, \tag{1}$$

where each $f_n(\mathbf{x})$ is a likelihood factor, that is, $f_n(\mathbf{x}) = p(y_n|\mathbf{x})$ where y_n is the n-th data instance, and $p(\mathbf{x})$ is a prior distribution, which we assume to belong to the exponential family, that is, $p(\mathbf{x}) = \exp\{\mathbf{s}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\nu}_0 - \Phi(\boldsymbol{\nu}_0)\}$, where $\boldsymbol{\nu}$ and $\mathbf{s}(\mathbf{x})$ are vectors of natural parameters and sufficient statistics, respectively, and $\Phi(\boldsymbol{\nu}_0)$ is the logarithm of the normalization constant of $\exp\{\mathbf{s}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\nu}_0\}$.

We can use EP to approximate the posterior $p(\mathbf{x}|\mathcal{D})$ with a distribution within the same exponential family as the prior. Let $q(\mathbf{x}) = \exp\{\mathbf{s}(\mathbf{x})^{\mathrm{T}}(\boldsymbol{\nu} + \boldsymbol{\nu}_0)\}$ be the unnormalized posterior approximation

used by EP. We have used here the parameterization $(\boldsymbol{\nu}+\boldsymbol{\nu}_0)$ to make explicit the contribution of the prior (given by $\boldsymbol{\nu}_0$) and likelihood (given by $\boldsymbol{\nu}$) in q. EP also uses a *cavity* distribution with the same form as q for each likelihood factor. Each cavity distribution approximates the result of removing a likelihood factor from the true posterior. Let $q^{\setminus n}(\mathbf{x}) = \exp\{\mathbf{s}(\mathbf{x})^{\mathrm{T}}(\boldsymbol{\lambda}_n + \boldsymbol{\nu}_0)\}$ be the unnormalized cavity distribution that approximates (1) when the n-th likelihood factor is removed. Again, we have used here a parameterization that makes explicit the contribution of the prior to each cavity.

According to [1], the EP energy function is

$$E(\boldsymbol{\nu}, \{\boldsymbol{\lambda}_n\}) = \Phi(\boldsymbol{\nu}_0) + (N-1)\log \int q(\mathbf{x}) d\mathbf{x} - \textstyle\sum_{n=1}^{N} \log \int f_n(\mathbf{x}) q^{\setminus n}(\mathbf{x}) d\mathbf{x}\,. \tag{2}$$

This energy is equal to minus the logarithm of the EP approximation of the model evidence $p(\mathcal{D})$, that is, the normalizer of the right-hand side of (1) [3]. Therefore, minimizing (2) with respect to $\boldsymbol{\nu}$ and $\{\boldsymbol{\lambda}_n\}$ is arguably a sensible way to tune these variational parameters. In practice, EP finds a stationary solution to the constrained optimization problem

$$\min_{\boldsymbol{\nu}} \max_{\{\boldsymbol{\lambda}_n\}} E(\boldsymbol{\nu}, \{\boldsymbol{\lambda}_n\}) \quad \text{subject to} \quad (N-1)\boldsymbol{\nu} = \textstyle\sum_{n=1}^{N} \boldsymbol{\lambda}_n\,, \tag{3}$$

where the constraint in (3) guarantees that the $\{\boldsymbol{\lambda}_n\}$ are valid cavity parameters. That is, the product of all the cavities is equal to $N-1$ times the posterior approximation (leaving the prior factor aside).

By replacing $\boldsymbol{\nu}+\boldsymbol{\nu}_0$ with $1/(N-1)\sum_{n=1}^{N}(\boldsymbol{\lambda}_n + \boldsymbol{\nu}_0)$ in (2) and then taking derivatives with respect to each $\{\boldsymbol{\lambda}_n\}$, we obtain the conditions that a solution to (3) must satisfy:

$$\mathbf{E}_q[\mathbf{s}(\mathbf{x})] = \mathbf{E}_{f_n q^{\setminus n}}[\mathbf{s}(\mathbf{x})]\,, \quad n = 1, \ldots, N\,. \tag{4}$$

That is, the expectation of $\mathbf{s}(\mathbf{x})$ with respect to the posterior approximation $q(\mathbf{x})$ and with respect to the product of $f_n(\mathbf{x})$ and $q^{\setminus n}(\mathbf{x})$, often called the *tilted* distribution, must be the equal.

The problem in (3) can be solved using a double-loop algorithm [5, 6]. This algorithm alternates between an optimization of the cavity parameters $\{\boldsymbol{\lambda}_n\}$ in the inner loop and an optimization of the parameters of the posterior approximation $\{\boldsymbol{\nu}\}$ in the outer loop. Each iteration of the double-loop algorithm is guaranteed to minimize the energy in (2). However, the alternating optimization of $\{\boldsymbol{\nu}\}$ and $\{\boldsymbol{\lambda}_n\}$ is very inefficient and the double-loop algorithm often requires too many iterations to be useful in practice.

2.1 Energy optimization with tied parameters

To simplify the previous optimization problem, we assume that all the cavity parameters are equal and have a simple form. We assume that each $\boldsymbol{\lambda}_n$ is obtained as a function of $\boldsymbol{\nu}$ according to

$$\boldsymbol{\lambda}_n = \tfrac{N-1}{N}\boldsymbol{\nu}\,, \tag{5}$$

which guarantees that the constraint in the right-hand side of (3) is satisfied. All the cavities are now the same and are obtained by scaling $\boldsymbol{\nu}$ in q by $(N-1)/N$. This assumption was made in [7] to obtain a version of EP called *Stochastic Expectation Propagation* that has low memory consumption. Here, we use this assumption to obtain another version of EP that is guaranteed to converge and can be implemented by optimizing an energy function without requiring a double-loop algorithm. Recall that $\boldsymbol{\nu}$ captures the contribution of the likelihood to the posterior. Thus, with (5) we are assuming that each factor $f_n(\mathbf{x})$ in (1) has an equal contribution to the likelihood and when we remove that factor, we only have to scale $\boldsymbol{\nu}$ by $(N-1)/N$.

After replacing $\boldsymbol{\lambda}_n + \boldsymbol{\nu}_0$ in (2) with the right-hand side of (5) and taking derivatives with respect to $\boldsymbol{\nu}$, we obtain the new stationary conditions:

$$\mathbf{E}_q[\mathbf{s}(\mathbf{x})] = \tfrac{1}{N}\textstyle\sum_{n=1}^{N} \mathbf{E}_{f_n q^{\setminus n}}[\mathbf{s}(\mathbf{x})]\,. \tag{6}$$

Therefore, the expectation of $\mathbf{s}(\mathbf{x})$ with respect to $q(\mathbf{x})$ is going to be equal to the average of the expectation of $\mathbf{s}(\mathbf{x})$ across the different tilted distributions $f_n(\mathbf{x})q^{\setminus n}(\mathbf{x})$, for $n = 1, \ldots, N$.

The solution that minimizes (2) with respect to $\boldsymbol{\nu}$ subject to (5) is expected to converge to the solution of (3) when more and more data are available. As N grows, we expect q and the cavities to become very peaked. When this happens, the contribution of each likelihood factor $f_n(\mathbf{x})$ to the tilted distribution $f_n(\mathbf{x})q^{\setminus n}(\mathbf{x})$ becomes very small because $f_n(\mathbf{x})$ is a rather flat function when compared to $q^{\setminus n}(\mathbf{x})$, which is very peaked. Therefore, as the amount of data N increases, we expect all the terms $\mathbf{E}_{f_n q^{\setminus n}}[\mathbf{s}(\mathbf{x})]$ in (6) to be very similar to each other. When all of them are equal, we have that (6) implies (4).

The convergence of a method that minimizes (2) under (5) is guaranteed if (2) is bounded below. This is going to be the case if we assume that all the factors $\{f_n(\mathbf{x})\}$ in (1) are bounded, that is, $f_n(\mathbf{x}) \leq c$ for some $c > 0$. See [1] for further details.

We can optimally adjust the prior hyper-parameters $\boldsymbol{\nu}_0$ by minimizing (2) with respect to $\boldsymbol{\nu}_0$ under the new form for the cavities given by (5). After taking gradients with respect to $\boldsymbol{\nu}_0$ and equating them to zero, we obtain the stationary condition $\mathbf{E}_q[\mathbf{s}(\mathbf{x})] = \mathbf{E}_p[\mathbf{s}(\mathbf{x})]$. That is, at convergence, the expectations of $\mathbf{s}(\mathbf{x})$ with respect to the posterior approximation $q(\mathbf{x})$ and with respect to the prior $p(\mathbf{x})$ should be the same.

3 Local minimization of α-divergences

The energy function given in (2) can be derived as the approximation to the logarithm of the marginal likelihood given by a message passing algorithm. This algorithm locally minimizes the Kullback-Leibler (KL) distance between the tilted distributions and the posterior approximation q. A generalized version of (2) is obtained by running a message passing algorithm that locally minimizes a generalization of the KL divergence called the α-divergence [3]. The new energy function is

$$E_\alpha(\boldsymbol{\nu}, \{\boldsymbol{\lambda}_n\}) = \Phi(\boldsymbol{\nu}_0) + \left(\tfrac{N}{\alpha} - 1\right) \log \int q(\mathbf{x}) d\mathbf{x} - \textstyle\sum_{n=1}^N \frac{1}{\alpha} \log \int f_n(\mathbf{x})^\alpha q_\alpha^{\backslash n}(\mathbf{x}) d\mathbf{x} \,. \quad (7)$$

where $q_\alpha^{\backslash n}(\mathbf{x}) = \left[q^{\backslash n}(\mathbf{x})/q(\mathbf{x})\right]^\alpha q(\mathbf{x})$ and $\alpha \in (-\infty, \infty) \setminus \{0\}$ is a parameter that determines the form of the α-divergence that is used to measure the distance between the n-th tilted distribution $f_n(\mathbf{x})q^{\backslash n}(\mathbf{x})$ and the posterior approximation $q(\mathbf{x})$. Again, we can simplify the optimization of this new energy function by assuming that all the cavity parameters are equal and have the form given by (5). This results in $q_\alpha^{\backslash n}(\mathbf{x}) = \exp\{\mathbf{s}(\mathbf{x})^\mathrm{T} \boldsymbol{\beta}_\alpha\}$ where $\boldsymbol{\beta}_\alpha = (N - \alpha)/N\boldsymbol{\nu} + \boldsymbol{\nu}_0$.

When α takes value one we have that the minimization of (7) under (5) is equivalent to the minimization of (2) under (5). In the limit that α approaches zero, we have that the minimizer of (7) under (5) converges to the solution that maximizes the variational lower bound. Therefore, by adjusting the α parameter, we can interpolate between the solutions given by variational Bayes and the solutions given by the approximation to expectation propagation described in Section 2.1.

3.1 Large scale learning

When N is very large, we can minimize (7) under (5) using stochastic optimization techniques. In particular, we can uniformly sample a mini-batch of data $\mathbf{S} \subseteq \{1, \ldots, N\}$ and construct the noisy estimate of the energy function given by

$$\hat{E}_\alpha(\boldsymbol{\nu}) = \Phi(\boldsymbol{\nu}_0) + \left(\tfrac{N}{\alpha} - 1\right) \log \int q(\mathbf{x}) d\mathbf{x} - \tfrac{N}{|\mathbf{S}|} \textstyle\sum_{n \in \mathbf{S}} \frac{1}{\alpha} \log \int f_n(\mathbf{x})^\alpha q_\alpha^{\backslash n}(\mathbf{x}) d\mathbf{x} \,. \quad (8)$$

The gradients of (8) can then used to minimize the original objective by stochastic gradient descent.

3.2 Black-box α-divergence minimization

In complicated probabilistic models, we might not be able to analytically solve the integrals in (8) involving the likelihood factors. However, we can obtain an estimate of these integrals by Monte Carlo. For this, we draw K samples $\mathbf{x}_1, \ldots, \mathbf{x}_K$ from $q_\alpha^{\backslash n}(\mathbf{x})$ and then approximate the integrals by expectations with respect to those samples. This produces the following new noisy estimate of the energy function

$$\hat{E}_\alpha(\boldsymbol{\nu}) = \Phi(\boldsymbol{\nu}_0) + \left(\tfrac{N}{\alpha} - 1\right) \log \int q(\mathbf{x}) d\mathbf{x} - \tfrac{N}{|\mathbf{S}|} \textstyle\sum_{n \in \mathbf{S}} \left[\frac{1}{\alpha} \log \frac{1}{K} \sum_{k=1}^K f_n(\mathbf{x}_k)^\alpha\right] - \frac{N}{\alpha} \Phi(\boldsymbol{\beta}_\alpha) \,, \quad (9)$$

where $\Phi(\boldsymbol{\beta}_\alpha)$ is the log-normalizer of $q_\alpha^{\backslash n}(\mathbf{x})$. Since $q_\alpha^{\backslash n}(\mathbf{x})$ and its samples $\mathbf{x}_1, \ldots, \mathbf{x}_K$ depend on $\boldsymbol{\nu}$, we can then use the reparametrization trick described in [8] to obtain the correct gradients of the Monte Carlo estimator. Note, however, that the resulting stochastic gradients will be biased because the energy function (9) applies a non-linear transformation (the logarithm) to the Monte Carlo estimator of the integrals. Nevertheless, this bias can be reduced by increasing the number of samples K.

Given a new probabilistic model, one can then use the proposed approach to quickly implement, in an automatic manner, an inference algorithm based on the local minimization of α-divergences. For this one only needs to write code that evaluates the likelihood factors $f_1, \ldots, f_N$ in (9). After this, the most difficult task is the computation of the gradients of (9) so that stochastic gradient descent with minibatches can be used to optimize the energy function. However, the computation of these gradients can be easily automated by using automatic differentiation tools such as autograd (`http://github.com/HIPS/autograd`) or theano [9]. This approach allows us to quickly implement and test different modeling assumptions when solving applied problems.

Table 1: Average Test Log-likelihood and Standard Errors, Probit Regression.

Dataset	**WB-α=1.0**	**BB-α=1.0**	**BB-α=10^{-6}**	**BB-VB**	**EP**
Ionosphere	-0.3211±0.0134	-0.3206±0.0134	-0.3204±0.0134	-0.3204±0.0134	**-0.3186±0.0187**
Madelon	-0.6771±0.0021	-0.6764±0.0019	-0.6763±0.0012	-0.6763±0.0012	**-0.6758±0.0017**
Pima	**-0.4993±0.0098**	-0.4997±0.0099	-0.5001±0.0099	-0.5001±0.0099	-0.4994±0.0098
Avg. Rank	3.1565±0.1280	3.0000±0.1053	3.1701±0.1196	3.2041±0.0931	2.4694±0.1245

Table 2: Average Test Log-likelihood and Standard Errors, Neural Networks.

Dataset	**BB-α=BO**	**BB-α=1**	**BB-α=10^{-6}**	**BB-VB**	**Avg. α**
Boston	**-2.549±0.019**	-2.621±0.041	-2.614±0.021	-2.578±0.017	0.45±0.04
Concrete	**-3.104±0.015**	-3.126±0.018	-3.119±0.010	-3.118±0.010	0.72±0.03
Energy	-0.979±0.028	-1.020±0.045	**-0.945±0.012**	-0.994±0.014	0.72±0.03
Wine	-0.949±0.009	**-0.945±0.008**	-0.967±0.008	-0.964±0.007	0.86±0.04
Yacht	**-1.102±0.039**	-2.091±0.067	-1.594±0.016	-1.646±0.017	0.48±0.01
Avg. Rank	1.835±0.065	2.504±0.080	2.766±0.061	2.895±0.057	

4 Experiments

We perform experiments with a Bayesian probit regression model to validate the black-box approach described in Section 3.2. We call this method black-box alpha (BB-α). We compare BB-α with a method that uses the analytic solution to the integrals, which is possible in the probit model. This method optimizes the energy function given by (8) and it is called white-box alpha (WB-α). We also compare with a method that optimizes a Monte Carlo approximation to the variational lower bound [4]. This approximation is obtained in a similar way as the one described in Section 3.2. We call this method black-box variational Bayes (BB-VB). In all the methods, the posterior approximation q is a factorized Gaussian. The prior $p(\mathbf{x})$ is a factorized Gaussian with zero mean. We optimize the different objective functions using minibatches of size 32 and Adam [10] with its default parameter values during 200 epochs. We optimize the prior variance as described at the end of Section 2.1. The prior variance is optimized after completing each epoch, but only during the last 150 epochs. BB-α and BB-VB are implemented by drawing $K = 100$ Monte Carlo samples for each minibatch. We try the values $\alpha = 1$ and $\alpha = 10^{-6}$ for BB-α and the value $\alpha = 1$ for WB-α. We also compare with a batch implementation of EP. Table 1 shows the average test log-likelihood obtained by each technique in the probit regression datasets. We also show the average rank obtained by each method across all the training/test splits. EP is the best method since it does not use stochastic optimization. Nevertheless, the other methods obtain results similar to those of EP. They also perform similarly among them, with BB-α=10^{-6} obtaining the same results as BB-VB. BB-α=1 also performs similarly to WB-α=1, which uses the exact gradients for the minibatches.

We also performed additional experiments with neural networks for regression with 50 hidden units and Gaussian additive noise at the output. The prior is also zero mean Gaussian. In this case we tune the learning rate and the prior variance for the network weights using Bayesian optimization (BO) techniques [11]. We also tune $\alpha \in (0, 1]$ in BB-α using BO methods. The noise variance is tuned by optimizing each method's objective function. Table 2 shows the results obtained by each method in the experiments with neural networks. The last column in this table shows the average value for α selected by the BO routine. In this case, the best method is the version of BB-α that tunes α with BO, that is, BB-α=BO. Here we do not consider EP or WB-α because the required integrals are not tractable.

5 Conclusions and future work

We have shown that, by considering the energy function used by expectation propagation (EP) and constraining the form of the cavity distributions in this method, we obtain an approximation to EP that is guaranteed to converge and can be implemented by optimizing an energy function without having to use inefficient double-loop algorithms. The proposed approach can be easily extended to minimize, instead of the original Kullback-Leibler divergence used by EP, a generalization of this distance called the α-divergence. Scalability to large datasets can be achieved by using stochastic gradient descent with minibatches. Furthermore, a combination of a Monte Carlo approximation and automatic differentiation methods allows our technique to be applied in a straightforward manner to arbitrary probabilistic models with complex likelihood factors. Experiments with probit regression models illustrate the accuracy of the proposed approach. Experiments with neural networks show that one can obtain gains by tuning the value of α to the data. The proposed approach can be easily extended to consider a different value of α for each likelihood factor or data point. We plan to design methods for optimally choosing these α values.

References

[1] Thomas P Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

[2] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

[3] Tom Minka et al. Divergence measures and message passing. Technical report, Technical report, Microsoft Research, 2005.

[4] Rajesh Ranganath, Sean Gerrish, and David Blei. {Black Box Variational Inference}. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 814–822, 2014.

[5] Tom Heskes and Onno Zoeter. Expectation propagation for approximate inference in dynamic bayesian networks. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 216–223. Morgan Kaufmann Publishers Inc., 2002.

[6] Manfred Opper and Ole Winther. Expectation consistent approximate inference. *The Journal of Machine Learning Research*, 6:2177–2204, 2005.

[7] Yingzhen Li, Jose Miguel Hernandez-Lobato, and Richard E Turner. Stochastic expectation propagation. *arXiv preprint arXiv:1506.04132*, 2015.

[8] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations*, 2014.

[9] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[10] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. pages 2951–2959, 2012.