Variational Bayesian methods

Contents

We introduce the variational Bayesian approach, of which the goal consists in approximating an untractable Bayesian posterior with other probability distributions. This approach also takes into account the cost of encoding parameters, and thus protects against overfitting.

1 Variational Bayesian approach

We are given a dataset $\mathcal{D} = (x_1, ..., x_n)$, and we want to model $\mathcal{D}$ using $(p_\theta)_{\theta \in \Theta}$.

An usual technique is the maximum likelihood: use $\theta^* := \mathrm{argmax}_\theta p_\theta(\mathcal{D})$. The problem with this approach is overfitting: while this approach gives a high probability to the training set, it often performs poorly when predicting new data points.

The MDL approach consists in taking into account the cost of describing the model, and a first remark is that if θ^* is a real number, the cost of encoding θ^* is infinite. Consequently, it is necessary to encode a whole region of Θ.

More precisely, the cost of describing the data with model θ is

$$L_\theta(\mathcal{D}) = -\ln p_\theta(\mathcal{D}). \tag{1}$$

Now, let us choose α a Bayesian prior on θ (i.e. $-\ln \alpha(\theta)$ is the cost of describing θ using the prior α). A way of describing data (or a generative model) is thus the following: pick $\theta \sim \alpha$, and then pick $\mathcal{D} \sim p_\theta$.

With this model, we get:

$$p(\mathcal{D}) = \int_\theta p_\theta(\mathcal{D}) \alpha(\theta) \mathrm{d}\theta \tag{2}$$

The compressed length of the data is therefore $-\ln p(\mathcal{D})$.

This model can also be used for prediction: consider $\mathcal{D}$ a training set, and $\mathcal{D}'$ a test set. We have:

$$p(\mathcal{D}'|\mathcal{D}) = \int_\theta p_\theta(\mathcal{D}')p(\theta|\mathcal{D}) \tag{3}$$

$$= \int p_\theta(\mathcal{D}')\frac{\alpha(\theta)p_\theta(\mathcal{D})}{\int_{\theta'}\alpha(\theta')p_{\theta'}(\mathcal{D})\mathrm{d}\theta'} \tag{4}$$

We can see in (4) that the θ used to compute the prediction are concentrated around the region where $\alpha(\theta)p_\theta(\mathcal{D})$ is large (and in the case where α is uniform, most of the mass will be around the maximum likelihood estimate θ^*).

Now, we need a practical way to evaluate the compressed length of the data $-\ln p(\mathcal{D})$ and the smoothed[1] prediction $p(\mathcal{D}'|\mathcal{D})$.

1.1 The variational bound

Notation 1. *We denote by $\pi(\theta)$ the posterior Bayesian distribution, namely:*

$$\pi(\theta) := p(\theta|\mathcal{D}) = \frac{\alpha(\theta)p_\theta(\mathcal{D})}{\int_{\theta'}\alpha(\theta')p_{\theta'}(\mathcal{D})\mathrm{d}\theta'} \tag{5}$$

However, most of the time, π is impossible to compute, and even sampling from π can be impossible. So it can be reasonable to use another probability distribution β instead, which would be "close enough" to π, but computable (as we will show later, some Gaussian β is reasonable). The following result (variational bound) gives a bound on the loss in compression provoked by this change.

Proposition 2 (Variational bound)**.** *For all β probability distribution on Θ, we have:*

$$-\ln p(\mathcal{D}) = -\mathbb{E}_{\theta\sim\beta}\ln p_\theta(\mathcal{D}) + \mathrm{KL}(\beta\|\alpha) - \mathrm{KL}(\beta\|\pi). \tag{6}$$

In particular, since $\mathrm{KL} \geqslant 0$*, we have*

$$-\ln p(\mathcal{D}) \leqslant -\mathbb{E}_{\theta\sim\beta}\ln p_\theta(\mathcal{D}) + \mathrm{KL}(\beta\|\alpha) \tag{7}$$

Proof. By expanding the KL divergences, we find that the right-hand term in (6) is equal to:

$$-\mathbb{E}_{\theta\sim\beta}\ln p_\theta(\mathcal{D}) + \int_\theta \beta(\theta)\ln\beta(\theta)\mathrm{d}\theta - \int_\theta \beta(\theta)\ln\alpha(\theta)\mathrm{d}\theta - \int_\theta \beta(\theta)\ln\beta(\theta)\mathrm{d}\theta + \int_\theta \beta(\theta)\ln\pi(\theta)\mathrm{d}\theta, \tag{8}$$

and by definition, we have:

$$\ln\pi(\theta) = \ln\alpha(\theta) + \ln p_\theta(\mathcal{D}) - \ln\int_{\theta'}\alpha(\theta')p_{\theta'}(\mathcal{D})\mathrm{d}\theta'. \tag{9}$$

By substituting (9) into (8), we find that the right-hand side is equal to:

$$-\int_\theta \beta(\theta)\ln\int_{\theta'}\alpha(\theta')p_{\theta'}(\mathcal{D})\mathrm{d}\theta'\mathrm{d}\theta = -\ln\int_{\theta'}\alpha(\theta')p_{\theta'}(\mathcal{D})\mathrm{d}\theta' = -\ln p_\theta(\mathcal{D}), \tag{10}$$

since the integral over θ is equal to 1. □

[1] "smoothed" because of the average on θ in (4).

This bound has a practical intepretation, because it is always possible to encode the distribution β with $\mathrm{KL}(\beta\|\alpha)$ nats[2] if we know α, by using the so called *bits-back* technique, as explained in [4]. An important use of this bound is that minimizing it is equivalent to minimizing $\mathrm{KL}(\beta\|\pi)$, which means that β can be used instead of π if sampling from the posterior distribution is needed. Consequently, minimizing this bound is interesting for two reasons: it gives a better final codelength, and it gives a better approximation of the posterior π.

Let us now optimize the bound (7) over β. We have:

$$
\begin{aligned}
L_\beta(\mathcal{D}) &:= -\mathbb{E}_{\theta\sim\beta}\ln p_\theta(\mathcal{D}) + \mathrm{KL}(\beta\|\alpha) && (11)\\
&= -\mathbb{E}_{\theta\sim\beta}\ln p_\theta(\mathcal{D}) - \mathbb{E}_{\theta\sim\beta}\ln\alpha(\theta) - \mathrm{Ent}(\beta), && (12)
\end{aligned}
$$

where $\mathrm{Ent}(\beta) := -\int_\theta \beta(\theta)\ln\beta(\theta)\mathrm{d}\theta$ is the entropy of β. Notice that the contribution of α has the same form as additional points in $\mathcal{D}$. The main regularization comes from $\mathrm{Ent}(\beta)$.

Let us consider for example that we encode explicitly an exact value of θ, i.e. $\beta := \delta_{\theta^*}$ is the Dirac mass at θ^*. The cost of describing θ^* as a real number is infinite, but even if we use the machine precision instead of a real Dirac (which is more realistic), the bound remains finite, but is very large. As we are going to see, a more reasonable choice would be a Gaussian β.

We want to optimize:

$$
L_\beta(\mathcal{D}) = -\mathbb{E}_{\theta\sim\beta}\ln p_\theta(\mathcal{D}) - \mathbb{E}_{\theta\sim\beta}\ln\alpha(\theta) - \mathrm{Ent}(\beta) \tag{13}
$$

Here, the second and third terms are respectively a penalty for "large" θ, and a penalty for too precise θ.

By definition, $\pi(\theta) \propto \alpha(\theta)p_\theta(\mathcal{D})$, so we have: $\ln\pi(\theta) = \ln\alpha(\theta) + \ln p_\theta(\mathcal{D}) +$ cst.

Now, if α is uniform[3], we have, around $\theta \approx \theta^*$:

$$
\ln\pi(\theta^*+\delta\theta) = \ln\pi(\theta^*) + \delta\theta\frac{\partial\ln p_\theta(\mathcal{D})}{\partial\theta}|_{\theta=\theta^*} + \frac{1}{2}\delta\theta^T[\mathrm{Hess}(\ln p_\theta(\mathcal{D}))]\delta\theta + o(\|\delta\theta\|^2) \tag{14}
$$

The first-order term is equal to zero (it is the derivative at the optimum), so we get:

$$
\pi(\theta) \approx \pi(\theta^*)\exp\left[\frac{1}{2}\delta\theta^T[\mathrm{Hess}(\ln p_\theta(\mathcal{D}))]\delta\theta\right]. \tag{15}
$$

In other words, the posterior is almost Gaussian near the maximum likelihood estimate, which justifies optimizing the variational bound only on Gaussian β, namely.

$$
\beta = \mathcal{N}(\bar\theta, \Sigma). \tag{16}
$$

It is therefore reasonable to compute the gradient descent of $L_\beta(\mathcal{D})$ for Gaussian β.

[2] We are using the natural logarithm.

[3] It is not always possible.

1.2 Gradient descent for Gaussians

We now give the gradient of $\mathbb{E}_{\theta\sim\mathcal{N}(\bar\theta,\Sigma)} f(\theta)$ with respect to $\bar\theta$ and Σ.

$\frac{\partial}{\partial\bar\theta}\mathbb{E}_{\theta\sim\mathcal{N}(\bar\theta,\Sigma)} f(\theta)$ can be computed in two different ways:

(we will not write the contribution of $\ln\alpha$, because it is similar to the contribution of $p_\theta(\mathcal{D})$).

Lemma 3 (Gradient with respect to the mean)**.** *We have:*

$$\frac{\partial}{\partial\bar\theta}\mathrm{Ent}(\beta) = 0, \tag{17}$$

and

$$\begin{aligned}\frac{\partial}{\partial\bar\theta}\mathbb{E}_{\theta\sim\mathcal{N}(\bar\theta,\Sigma)} f(\theta) =&\mathbb{E}_{\theta\sim\mathcal{N}(\bar\theta,\Sigma)}\left(\frac{\partial f(\theta)}{\partial\theta}|_{\theta=\bar\theta}\right) && (18)\\ =&\Sigma^{-1}\mathbb{E}\left[(\theta-\bar\theta)f(\theta)\right] && (19)\\ \approx&\frac{1}{k}\sum_{\theta_i\sim\mathcal{N}(\bar\theta,\Sigma)}\frac{\partial f}{\partial\theta_i}, && (20)\end{aligned}$$

Proof. For the first form of $\frac{\partial}{\partial\bar\theta}\mathbb{E}_{\theta\sim\mathcal{N}(\bar\theta,\Sigma)} f(\theta)$:

$$\begin{aligned}\frac{\partial}{\partial\bar\theta}\mathbb{E}_{\theta\sim\mathcal{N}(\bar\theta,\Sigma)} f(\theta) =&\frac{\partial}{\partial\bar\theta}\int\beta(\theta)f(\theta) && (21)\\ =&\int\beta(\theta)\frac{\partial\ln\beta(\theta)}{\partial\bar\theta}f(\theta) && (22)\\ =&\Sigma^{-1}\mathbb{E}(\theta-\bar\theta)f(\theta). && (23)\end{aligned}$$

For its second form:

$$\begin{aligned}\frac{\partial}{\partial\bar\theta}\mathbb{E}_{\theta\sim\mathcal{N}(\bar\theta,\Sigma)} f(\theta) =&\frac{\partial}{\partial\bar\theta}\mathbb{E}_{\xi\sim\mathcal{N}(0,\Sigma)} f(\bar\theta+\xi) && (24)\\ =&\mathbb{E}_{\xi\sim\mathcal{N}(0,\Sigma)}\frac{\partial}{\partial\bar\theta}f(\bar\theta+\xi) && (25)\\ =&\mathbb{E}_{\theta\sim\mathcal{N}(\bar\theta,\Sigma)}\frac{\partial f}{\partial\theta} && (26)\\ \text{(Monte-Carlo)}\approx&\frac{1}{k}\sum_{\theta_i\sim\mathcal{N}(\bar\theta,\Sigma)}\frac{\partial f}{\partial\theta_i}, && (27)\end{aligned}$$

□

The second method involves the computation of derivatives that might be difficult, whereas the first one will need more samples to work.

This gradient descent already has reasonable performance when used with fixed Σ, but the update of the covariance matrix is also computable:

Lemma 4 (Gradient with respect to the covariance matrix)**.** *We have:*

$$\frac{\partial}{\partial\Sigma}\mathrm{Ent}(\beta) = \frac{1}{2}\Sigma^{-1}, \tag{28}$$

and

$$
\begin{aligned}
\frac{\partial}{\partial\Sigma}(-\mathbb{E}_{\theta\sim\beta}\ln p_\theta(\mathcal{D})) =& \frac{1}{2}\mathbb{E}_{\theta\sim\beta}\frac{\partial^2(-\ln p_\theta(\mathcal{D}))}{\partial\theta^2} && (29)\\
=& -\frac{1}{2}\mathbb{E}\left[\frac{\partial\ln p_\theta(\mathcal{D})}{\partial\theta}(\theta-\bar{\theta})^T\Sigma^{-1}\right] && (30)\\
=& -\frac{1}{2}\mathbb{E}_{\theta\sim\beta}\left[\ln p_\theta(\mathcal{D})\Sigma^{-1}(\theta-\bar{\theta})(\theta-\bar{\theta})\Sigma^{-1}-\Sigma^{-1}\right] && (31)
\end{aligned}
$$

Proof. Admitted. □

Similarly to the previous case, $\frac{\partial}{\partial\Sigma}(-\mathbb{E}_{\theta\sim\beta}\ln p_\theta(\mathcal{D}))$ can be computed in three different ways: (29) requires computing second order derivatives (although the *Gauss–Newton approximation*, which consits in replacing the second derivative by the tensor square of the gradient, can be used), (30) requires more samples than (29) to be efficient, but only first-order derivatives need to be computed, and finally, the CMA-like update (31) avoids computing derivatives altogether, but even more samples are needed.

This method has been tested in [1] for neural networks, and yielded good results. However, it would be better to use the natural gradient, which is given by:

$$\bar{\theta} \leftarrow \bar{\theta} - \eta\Sigma\frac{\partial L_\beta(\mathcal{D})}{\partial\bar{\theta}}, \tag{32}$$

and

$$\Sigma \leftarrow \Sigma - 2\eta\Sigma\frac{\partial L_\beta(\mathcal{D})}{\partial\Sigma}\Sigma. \tag{33}$$

It is interesting to notice that $\bar{\theta}$ will not necessarily converge to the maximum likelihood estimate. Indeed, Lemma 4 shows that there is an equilibrium for Σ, which is given by:

$$\frac{1}{2}\Sigma^{-1} = \frac{1}{2}\mathbb{E}_\theta\frac{\partial^2\ln p_\theta(\mathcal{D})}{\partial\theta^2}. \tag{34}$$

Consequently, if the Hessian at the ML estimate is too large, then Σ will be very small, thus yielding a large loss because of the entropy term in $L_\beta(\mathcal{D})$ (we recall equation (13): $L_\beta(\mathcal{D}) = -\mathbb{E}_{\theta\sim\beta}\ln p_\theta(\mathcal{D}) - \mathbb{E}_{\theta\sim\beta}\ln\alpha(\theta) - \mathrm{Ent}(\beta)$).

A visual interpretation for this is the following: suppose that there is only a very small area near the ML estimate θ^* which fit well the data (large Hessian), while there is another θ' which is almost as good as the ML estimate, and such that neighbouring values are also good. In that case, it is better to encode a Gaussian around θ', because less precision is needed, thus yielding a shorter codelength.

1.3 A possible application with dropout

For a neural network, the *Dropout*, introduced in [3], is roughly the following procedure: during the training, at each step, each weight has a probability $\frac{1}{2}$ to be omitted. Then, for testing, all weights are activated, and halved. This can be seen as a "modified" variational bound, with the improper prior $\alpha(\theta) =$

$\frac{1}{2}\delta_0 + \frac{1}{2}\alpha_0(\theta)$, where $\alpha_0(\theta)$ would be "uniform over $\mathbb{R}$", with a posterior of the form:

$$\beta(\theta) = \frac{1}{2}\delta_0(\theta) + \frac{1}{2}\delta_{2\bar{\theta}}(\theta). \tag{35}$$

The variational bound is then simply optimizing the likelihood of the data (this is the only term remaining in (12)), and then averaging, instead of resampling. One could argue that there is not much left of the variational bound, but the important common characteristic is that optimizing with noise creates robustness.

References

[1] Alex Graves. Practical variational inference for neural networks. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *NIPS*, pages 2348–2356, 2011.

[2] Peter D. Grünwald. *The Minimum Description Length Principle (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.

[3] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[4] Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, pages 5–13, New York, NY, USA, 1993. ACM.

[5] Ray J. Solomonoff. A formal theory of inductive inference. *Information and Control*, 7, 1964.