Bayesian Logistic Regression

Jan Drugowitsch

Apr 2008, last update: May 2010

The Model

The data y is, dependent on the D-dimensional input $\boldsymbol{x}$, assumed to be of either class $y = -1$ or $y = 1$. The log-likelihood ratio $\ln(p(y = 1|\boldsymbol{x}, \boldsymbol{w})/p(y = -1|\boldsymbol{x}, \boldsymbol{w}))$ is assumed to be linear in $\boldsymbol{x}$, such that the conditional likelihood for $y = 1$ is given by the sigmoid

$$p(y = 1|\boldsymbol{x}, \boldsymbol{w}) = \frac{1}{1 + \exp(-\boldsymbol{w}^T\boldsymbol{x})} = \sigma(\boldsymbol{w}^T\boldsymbol{x}). \tag{1}$$

Equally, $p(y = -1|\boldsymbol{x}, \boldsymbol{w}) = 1 - p(y = 1|\boldsymbol{x}, \boldsymbol{w}) = 1/(1 + \exp(\boldsymbol{w}^T\boldsymbol{x}))$, such that

$$p(y|\boldsymbol{x}, \boldsymbol{w}) = \sigma(y\boldsymbol{w}^T\boldsymbol{x}). \tag{2}$$

Given some data $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{Y}\}$, where $\boldsymbol{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ and $\boldsymbol{Y} = \{y_1, \dots, y_N\}$ are N input/output pairs, the aim is to find the posterior $p(\boldsymbol{w}|\mathcal{D})$, given some prior $p(\boldsymbol{w})$. Unfortunately, the sigmoid data likelihood does not admit a conjugate-exponential prior. Therefore, approximations need to be applied to find an analytic expression for the posterior.

The approximation that will be used is quadratic in $\boldsymbol{w}$ in the exponential, such that the conjugate Gaussian prior

$$p(\boldsymbol{w}|\alpha) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \alpha^{-1}\boldsymbol{I}) = \left(\frac{\alpha}{2\pi}\right)^{D/2} \exp\left(-\frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w}\right) \tag{3}$$

can be used. This prior is parametrised by the hyper-parameter $\boldsymbol{\alpha}$ that is modelled by a conjugate Gamma distribution

$$p(\alpha) = \text{Gam}(\alpha|a_0, b_0) = \frac{1}{\Gamma(a_0)} b_0^{a_0} \alpha^{a_0 - 1} \exp(-b_0\alpha). \tag{4}$$

Variational Bayesian Inference

Variational Bayesian inference is based on maximising a lower bound on the marginal data log-likelihood

$$\ln p(\boldsymbol{Y}|\boldsymbol{X}) = \ln \iint p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{w}) p(\boldsymbol{w}|\alpha) p(\alpha) \mathrm{d}\boldsymbol{w}\mathrm{d}\alpha. \tag{5}$$

This lower bound is given by

$$\ln p(\boldsymbol{Y}|\boldsymbol{X}) \geq \mathcal{L}(q) = \iint q(\boldsymbol{w}, \alpha) \ln \frac{p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{w}) p(\boldsymbol{w}|\alpha) p(\alpha)}{q(\boldsymbol{w}, \alpha)} \mathrm{d}\boldsymbol{w}\mathrm{d}\alpha, \tag{6}$$

where the variational distribution $q(\boldsymbol{w}, \alpha)$, approximating the posterior $p(\boldsymbol{w}, \alpha|\mathcal{D})$, is assumed to factor into $q(\boldsymbol{w}, \alpha) = q(\boldsymbol{w})q(\alpha)$. This approximation leads to analytic posterior expressions if the model structure is conjugate-exponential.

The data likelihood

$$p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{w}) = \prod_n p(y_n|\boldsymbol{x}_n,\boldsymbol{w}) = \prod_n \sigma(y_n \boldsymbol{w}^T \boldsymbol{x}_n) \tag{7}$$

does not admit a conjugate prior in the exponential family and will be approximated by the use of

$$\sigma(z) \geq \sigma(\xi) \exp\left((z-\xi)/2 - \lambda(\xi)(z^2-\xi^2)\right), \quad \lambda(\xi) = \frac{1}{2\xi}\left(\sigma(\xi) - \frac{1}{2}\right), \tag{8}$$

which is a tight lower bound on the sigmoid, parametrised by ξ (Jaakkola and Jordan, 1998). Applying this bound, the data log-likelihood is lower-bounded by

$$\begin{aligned} \ln p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{w}) &\geq \ln h(\boldsymbol{w},\boldsymbol{\xi}) && (9)\\ &= \boldsymbol{w}^T \sum_n \frac{y_n}{2}\boldsymbol{x}_n - \boldsymbol{w}^T\left(\sum_n \lambda(\xi_n)\boldsymbol{x}_n\boldsymbol{x}_n^T\right)\boldsymbol{w} \\ &\quad + \sum_n \left(\ln\sigma(\xi_n) - \frac{\xi_n}{2} + \lambda(\xi_n)\xi_n^2\right), && (10) \end{aligned}$$

with one local variation parameter ξ_n per datum. This results in the new variational bound

$$\tilde{\mathcal{L}}(q,\boldsymbol{\xi}) = \iint q(\boldsymbol{w},\alpha) \ln \frac{h(\boldsymbol{w},\boldsymbol{\xi})p(\boldsymbol{w}|\alpha)p(\alpha)}{q(\boldsymbol{w},\alpha)} \mathrm{d}\boldsymbol{w}\mathrm{d}\alpha, \tag{11}$$

which is a lower bound on the original variational bound, that is $\tilde{\mathcal{L}}(q,\boldsymbol{\xi}) \leq \mathcal{L}(q)$.

The variational posteriors are evaluated by standard variational methods for factorised distributions. The variational posterior for $\boldsymbol{w}$ is given by

$$\begin{aligned} \ln q^*(\boldsymbol{w}) &= \ln h(\boldsymbol{w},\boldsymbol{\xi}) + \mathbb{E}_\alpha(\ln p(\boldsymbol{w}|\alpha)) + \text{const.} && (12)\\ &= \boldsymbol{w}^T \sum_n \frac{y_n}{2}\boldsymbol{x}_n - \frac{1}{2}\boldsymbol{w}^T\left(\mathbb{E}_\alpha(\alpha)\boldsymbol{I} + 2\sum_n \lambda(\xi_n)\boldsymbol{x}_n\boldsymbol{x}_n^T\right)\boldsymbol{w} + \text{const.} && (13)\\ &= \ln \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_N,\boldsymbol{V}_N), && (14) \end{aligned}$$

with

$$\begin{aligned} \boldsymbol{V}_N^{-1} &= \mathbb{E}_\alpha(\alpha)\boldsymbol{I} + 2\sum_n \lambda(\xi_n)\boldsymbol{x}_n\boldsymbol{x}_n^T, && (15)\\ \boldsymbol{w}_N &= \boldsymbol{V}_N \sum_n \frac{y_n}{2}\boldsymbol{x}_n. && (16) \end{aligned}$$

The variational posterior for α results in

$$\begin{aligned} \ln q^*(\alpha) &= \mathbb{E}_{\boldsymbol{w}}(\ln p(\boldsymbol{w}|\alpha)) + \ln p(\alpha) + \text{const.} && (17)\\ &= \left(a_0 - 1 + \frac{D}{2}\right)\ln\alpha - \left(b_0 + \frac{1}{2}\mathbb{E}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{w})\right)\alpha + \text{const.} && (18)\\ &= \ln \text{Gam}(\alpha|a_N,b_N), && (19) \end{aligned}$$

with

$$\begin{aligned} a_N &= a_0 + \frac{D}{2}, && (20)\\ b_N &= b_0 + \frac{1}{2}\mathbb{E}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{w}). && (21) \end{aligned}$$

The expectations are evaluated with respect to the variational posteriors and result in

$$\begin{aligned} \mathbb{E}_\alpha(\alpha) &= \frac{a_N}{b_N}, && (22) \\ \mathbb{E}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{w}) &= \boldsymbol{w}_N^T\boldsymbol{w}_N + \mathrm{Tr}(\boldsymbol{V}_N). && (23) \end{aligned}$$

The variational bound itself is given by

$$\begin{aligned} \tilde{\mathcal{L}}(q, \boldsymbol{\xi}) &= \mathbb{E}_{\boldsymbol{w}}(\ln h(\boldsymbol{w}, \boldsymbol{\xi})) + \mathbb{E}_{\boldsymbol{w},\alpha}(\ln p(\boldsymbol{w}|\alpha)) + \mathbb{E}_\alpha(\ln p(\alpha)) \\ &\quad -\mathbb{E}_{\boldsymbol{w}}(\ln q(\boldsymbol{w})) - \mathbb{E}_\alpha(\ln q(\alpha)), && (24) \\ \mathbb{E}_{\boldsymbol{w}}(\ln h(\boldsymbol{w}, \boldsymbol{\xi})) &= \frac{1}{2}\boldsymbol{w}_N^T\boldsymbol{V}_N^{-1}\boldsymbol{w}_N - \frac{D}{2} + \frac{1}{2}\frac{a_N}{b_N}\left(\boldsymbol{w}_N^T\boldsymbol{w}_N + \mathrm{Tr}(\boldsymbol{V}_N)\right) \\ &\quad + \sum_n \left(\ln\sigma(\xi_n) - \frac{\xi_n}{2} + \lambda(\xi_n)\xi_n^2\right), && (25) \\ \mathbb{E}_{\boldsymbol{w},\alpha}(\ln p(\boldsymbol{w}|\alpha)) &= -\frac{D}{2}\ln 2\pi + \frac{D}{2}(\psi(a_N) - \ln b_N) - \frac{1}{2}\frac{a_N}{b_N}\left(\boldsymbol{w}_N^T\boldsymbol{w}_N + \mathrm{Tr}(\boldsymbol{V}_N)\right), && (26) \\ \mathbb{E}_\alpha(\ln p(\alpha)) &= -\ln\Gamma(a_0) + a_0\ln b_0 + (a_0 - 1)(\psi(a_N) - \ln b_N) - b_0\frac{a_N}{b_N}, && (27) \\ \mathbb{E}_{\boldsymbol{w}}(\ln q(\boldsymbol{w})) &= -\frac{1}{2}\ln|\boldsymbol{V}_N| - \frac{D}{2}(1 + \ln 2\pi), && (28) \\ \mathbb{E}_\alpha(\ln q(\alpha)) &= -\ln\Gamma(a_N) + (a_N - 1)\psi(a_N) + \ln b_N - a_N, && (29) \end{aligned}$$

where $\psi(\cdot)$ is the digamma function. In combination, this gives

$$\begin{aligned} \tilde{\mathcal{L}}(q, \boldsymbol{\xi}) &= \frac{1}{2}\boldsymbol{w}_N^T\boldsymbol{V}_N^{-1}\boldsymbol{w}_N + \frac{1}{2}\ln|\boldsymbol{V}_N| + \sum_n\left(\ln\sigma(\xi_n) - \frac{\xi_n}{2} + \lambda(\xi_n)\xi_n^2\right) \\ &\quad -\ln\Gamma(a_0) + a_0\ln b_0 - b_0\frac{a_N}{b_N} - a_N\ln b_N + \ln\Gamma(a_N) + a_N. && (30) \end{aligned}$$

This bound is to be maximised in order to find the variational posteriors for $\boldsymbol{w}$ and α. The expressions that maximise this bound with respect to $q(\boldsymbol{w})$ and $q(\boldsymbol{\alpha})$, while keeping all other parameters fixed, are given by $q^*(\boldsymbol{w})$ and $q^*(\alpha)$ respectively. To find the local variational parameters ξ_n that maximise $\tilde{\mathcal{L}}(q, \boldsymbol{\xi})$, its derivative with respect to ξ_n is set to zero (see (Bishop, 2006)), resulting in

$$(\xi_n^{\text{new}})^2 = \boldsymbol{x}_n^T\left(\boldsymbol{V}_N + \boldsymbol{w}_N\boldsymbol{w}_N^T\right)\boldsymbol{x}_n. \quad (31)$$

The variational bound is maximised by iterating over the update equations for $\boldsymbol{w}_N$, $\boldsymbol{V}_N$, a_N, b_N and $\boldsymbol{\xi}$, until $\tilde{\mathcal{L}}(q, \boldsymbol{\xi})$ reaches a plateau. A lower bound on the marginal data log-likelihood $p(\mathcal{D})$ is given by the bound itself, as $\ln p(\mathcal{D}) \geq \mathcal{L}(q) \geq \tilde{\mathcal{L}}(q, \boldsymbol{\xi})$.

Predictive Density

In order to get the predictive density, the posterior $p(\boldsymbol{w}|\mathcal{D})$ is approximated by the variational posterior $q(\boldsymbol{w})$, and the sigmoid is lower-bounded by above bound, such that

$$\begin{aligned} p(y = 1|\boldsymbol{x}, \mathcal{D}) &= \int p(y = 1|\boldsymbol{x}, \boldsymbol{w})p(\boldsymbol{w}|\mathcal{D})\mathrm{d}\boldsymbol{w} && (32) \\ &\approx \int p(y = 1|\boldsymbol{x}, \boldsymbol{w})q(\boldsymbol{w})\mathrm{d}\boldsymbol{w}, && (33) \\ &\geq \int \sigma(\xi)\exp\left(\frac{\boldsymbol{w}^T\boldsymbol{x} - \xi}{2} - \lambda(\xi)\boldsymbol{w}^T\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{w} + \lambda(\xi)\xi^2\right)q(\boldsymbol{w})\mathrm{d}\boldsymbol{w}. && (34) \end{aligned}$$

The integral is solved by noting that the lower bound is exponentially quadratic in $\boldsymbol{w}$, such that the Gaussian can be completed, to give

$$\ln p(y=1|\boldsymbol{x},\mathcal{D}) \approx \frac{1}{2}\ln\frac{|\tilde{\boldsymbol{V}}|}{|\boldsymbol{V}_N|} - \frac{1}{2}\boldsymbol{w}_N^T\boldsymbol{V}_N^{-1}\boldsymbol{w}_N + \frac{1}{2}\tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{V}}^{-1}\tilde{\boldsymbol{w}} + \ln\sigma(\xi) - \frac{\xi}{2} + \lambda(\xi)\xi^2, \quad (35)$$

with

$$\tilde{\boldsymbol{V}}^{-1} = \boldsymbol{V}_N^{-1} + 2\lambda(\xi)\boldsymbol{x}\boldsymbol{x}^T, \quad (36)$$

$$\tilde{\boldsymbol{w}} = \tilde{\boldsymbol{V}}\left(\boldsymbol{V}_N^{-1}\boldsymbol{w}_N + \frac{\boldsymbol{x}}{2}\right). \quad (37)$$

The bound parameter ξ that maximises $\ln p(y=1|\boldsymbol{x},\mathcal{D})$ is given by

$$(\xi^{\text{new}})^2 = \boldsymbol{x}^T\left(\tilde{\boldsymbol{V}} + \tilde{\boldsymbol{w}}\tilde{\boldsymbol{w}}^T\right)\boldsymbol{x}. \quad (38)$$

Thus, the predictive density is found by iterating over the updates for $\tilde{\boldsymbol{w}}$, $\tilde{\boldsymbol{V}}$ and ξ until $\ln p(y=1|\boldsymbol{x},\mathcal{D})$ reaches a plateau. The hyper-prior $p(\alpha)$ does not need to be considered as it does not appear in the variational posterior $q(\boldsymbol{w})$.

Using Automatic Relevance Determination

To use Automatic Relevance Determination (ARD), each element of the prior of $\boldsymbol{w}$ is assigned a separate prior,

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0},\boldsymbol{A}^{-1}) = \frac{|\boldsymbol{A}|^{1/2}}{\sqrt{2\pi}^D}\exp\left(-\frac{1}{2}\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w}\right), \quad (39)$$

where $\boldsymbol{A}$ is the diagonal matrix with the vector $\boldsymbol{\alpha} = (\alpha_1,\dots,\alpha_D)^T$ along its diagonal. The conjugate hyper-prior $p(\boldsymbol{\alpha})$ is given by

$$p(\boldsymbol{\alpha}) = \prod_i \text{Gam}(\alpha_i|a_0,b_0). \quad (40)$$

Note that α_i determines the precision (inverse variance) of the ith element of $\boldsymbol{w}$. A low precision makes the prior uninformative, whereas a high precision tells us that the associated element in $\boldsymbol{w}$ is most likely zero and the associated input element is therefore irrelevant for the prediction of y. Thus, such a prior structure automatically determines the relevance of each element of the input to predict the class of the output.

Using the same variational Bayes inference as before, the variational posteriors are given by

$$q^*(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_N,\boldsymbol{V}_N), \qquad q^*(\boldsymbol{\alpha}) = \prod_i \text{Gam}(\alpha_i|a_N,b_{Ni}), \quad (41)$$

with

$$\boldsymbol{V}_N^{-1} = \mathbb{E}_\alpha(\boldsymbol{A}) + 2\sum_n \lambda(\xi_n)\boldsymbol{x}_n\boldsymbol{x}_n^T, \quad (42)$$

$$\boldsymbol{w}_N = \boldsymbol{V}_N\sum_n \frac{y_n}{2}\boldsymbol{x}_n, \quad (43)$$

$$a_N = a_0 + \frac{1}{2}, \quad (44)$$

$$b_{Ni} = b_0 + \frac{1}{2}\mathbb{E}_{\boldsymbol{w}}(w_i^2), \quad (45)$$

where w_i is the ith element of $\boldsymbol{w}$, and $\boldsymbol{A}_N = \mathbb{E}_\alpha(\boldsymbol{A})$ is a diagonal matrix with its ith diagonal element given by $\mathbb{E}_\alpha(\alpha_i) = a_N/b_{Ni}$. $\mathbb{E}_{\boldsymbol{w}}(w_i^2)$ evaluates to $\mathbb{E}_{\boldsymbol{w}}(w_i^2) = \mathbb{E}_{\boldsymbol{w}}(w_i)^2 + \text{var}_{\boldsymbol{w}}(w_i) = (\boldsymbol{w}_N)_i^2 + (\boldsymbol{V}_N)_{ii}$.

The new expectations to evaluate the variation bound are

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{w}}(\ln h(\boldsymbol{w}, \boldsymbol{\xi})) &= \frac{1}{2}\boldsymbol{w}_N^T \boldsymbol{V}_N^{-1} \boldsymbol{w}_N - \frac{D}{2} + \frac{1}{2}\left(\text{Tr}(\boldsymbol{A}_N \boldsymbol{V}_N) + \boldsymbol{w}_N^T \boldsymbol{A}_N \boldsymbol{w}_N\right) \\
&\quad + \sum_n \left(\ln \sigma(\xi_n) - \frac{\xi_n}{2} + \lambda(\xi_n)\xi_n^2\right), \qquad (46) \\
\mathbb{E}_{\boldsymbol{w},\boldsymbol{\alpha}}(\ln p(\boldsymbol{w}|\boldsymbol{\alpha})) &= \frac{1}{2}\sum_i \left(\psi(a_N) - \ln b_{Ni}\right) \\
&\quad - \frac{D}{2}\ln 2\pi - \frac{1}{2}\left(\text{Tr}(\boldsymbol{A}_N \boldsymbol{V}_N) + \boldsymbol{w}_N^T \boldsymbol{A}_N \boldsymbol{w}_N\right), \qquad (47) \\
\mathbb{E}_{\boldsymbol{\alpha}}(\ln p(\boldsymbol{\alpha})) &= \sum_i \left(-\ln \Gamma(a_0) + a_0 \ln b_0 + (a_0 - 1)(\psi(a_N) - \ln b_{Ni}) - b_0 \frac{a_N}{b_{Ni}}\right) \qquad (48) \\
\mathbb{E}_{\boldsymbol{\alpha}}(\ln q(\boldsymbol{\alpha})) &= \sum_i \left(-\ln \Gamma(a_N) + (a_N - 1)\psi(a_N) + \ln b_{Ni} - a_N\right), \qquad (49)
\end{aligned}$$

resulting in

$$\begin{aligned}
\tilde{\mathcal{L}}(q, \boldsymbol{\xi}) &= \frac{1}{2}\boldsymbol{w}_N^T \boldsymbol{V}_N^{-1} \boldsymbol{w}_N + \frac{1}{2}\ln |\boldsymbol{V}_N| + \sum_n \left(\ln \sigma(\xi_n) - \frac{\xi_n}{2} + \lambda(\xi_n)\xi_n^2\right) \\
&\quad + \sum_i \left(-\ln \Gamma(a_0) + a_0 \ln b_0 - b_0 \frac{a_N}{b_{Ni}} - a_N \ln b_{Ni} + \ln \Gamma(a_N) + a_N\right) \qquad (50)
\end{aligned}$$

As the variational posterior expression $q^*(\boldsymbol{w})$ is independent of the hyper-parameters, the predictive density is evaluated as before.

Implementation

Bayesian logistic regression can be implemented in various variants, some of which are discussed below. The first variant adds each $\boldsymbol{x}_n$ incrementally, optimising each ξ_n in turn. All other variants add all $\boldsymbol{x}_n$ at once, optimising all ξ_n in combination.

Incremental Posterior Update

The script bayes_logit_fit_iter .m updates the posterior parameters incrementally by adding the observations $\boldsymbol{x}_n, y_n$ one by one, while optimising ξ_n for each of those observations separately. Let $\boldsymbol{V}_j$ and $\boldsymbol{w}_j$ denote the parameters of $q^*(\boldsymbol{w})$ after j observations have been made. $\boldsymbol{V}_j$ follows the incremental update

$$\boldsymbol{V}_j^{-1} = \boldsymbol{V}_{j-1}^{-1} + 2\lambda(\xi_j)\boldsymbol{x}_j \boldsymbol{x}_j^T, \qquad (51)$$

starting with $\boldsymbol{V}_j^{-1} = \mathbb{E}_\alpha(\alpha)\boldsymbol{I}$. The incremental update of $\boldsymbol{w}_j$ is slightly more complex, but from observing that

$$\boldsymbol{V}_j^{-1}\boldsymbol{w}_j = \sum_n^j \frac{y_n}{2}\boldsymbol{x}_n = \frac{y_j}{2}\boldsymbol{x}_j + \sum_n^{j-1} \frac{y_n}{2}\boldsymbol{x}_n = \boldsymbol{V}_{j-1}^{-1}\boldsymbol{w}_{j-1} + \frac{y_j}{2}\boldsymbol{x}_j, \qquad (52)$$

it is easy to see that

$$\boldsymbol{w}_j = \boldsymbol{V}_j \left(\boldsymbol{V}_{j-1}^{-1}\boldsymbol{w}_{j-1} + \frac{y_j}{2}\boldsymbol{x}_j\right). \qquad (53)$$

The script avoids taking the inverse of $\boldsymbol{V}$ by updating $\boldsymbol{V}^{-1}$ and $\boldsymbol{V}$ in parallel, where the latter is based on an application of the Sherman-Morrison formula on the $\boldsymbol{V}^{-1}$ update, resulting in

$$\boldsymbol{V}_j = \left(\boldsymbol{V}_{j-1}^{-1} + 2\lambda(\xi_j)\boldsymbol{x}_j\boldsymbol{x}_j^T\right)^{-1} = \boldsymbol{V}_{j-1} - \frac{2\lambda(\xi_j)\boldsymbol{V}_{j-1}\boldsymbol{x}_j\boldsymbol{x}_j^T\boldsymbol{V}_{j-1}}{1+2\lambda(\xi_j)\boldsymbol{x}_j^T\boldsymbol{V}_{j-1}\boldsymbol{x}_j}. \tag{54}$$

$\ln|\boldsymbol{V}_j|$ can be updated in a similar way, based on the Matrix determinant lemma,

$$|\boldsymbol{V}_j^{-1}| = |\boldsymbol{V}_{j-1}^{-1} + 2\lambda(\xi_j)\boldsymbol{x}_j\boldsymbol{x}_j^T| = |\boldsymbol{V}_{j-1}^{-1}|\left(1+2\lambda(\xi_j)\boldsymbol{x}_j^T\boldsymbol{V}_{j-1}\boldsymbol{x}_j\right), \tag{55}$$

such that, using $\ln|\boldsymbol{V}_j| = -\ln|\boldsymbol{V}_j^{-1}|$,

$$\ln|\boldsymbol{V}_j| = \ln|\boldsymbol{V}_{j-1}| - \ln\left(1+2\lambda(\xi_j)\boldsymbol{x}_j^T\boldsymbol{V}_{j-1}\boldsymbol{x}_j\right). \tag{56}$$

In the iterative version of the script the hyperprior α is ignored, and instead $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, D^{-1}\boldsymbol{I})$ is used. This leads to the initial parameters $\boldsymbol{w}_0 = \boldsymbol{0}$, $\boldsymbol{V}_0 = D^{-1}\boldsymbol{I}$, $\boldsymbol{V}_0^{-1} = D\boldsymbol{I}$, and $\ln|\boldsymbol{V}_0^{-1}| = -D\ln D$.

The script iterates over $j = 1,\dots,N$, at each step optimising ξ_j iteratively to maximise the variational bound. At each step it starts at $\xi_j = 0$, such that $\lambda(\xi_j) = 1/8$, leading to a simplified initial step,

$$\begin{aligned}
\boldsymbol{V}_j^{-1}(\xi_j) &=_{\xi_j=0} \quad \boldsymbol{V}_{j-1}^{-1} + \frac{1}{4}\boldsymbol{x}_j\boldsymbol{x}_j^T, && (57)\\
\boldsymbol{V}_j(\xi_j) &=_{\xi_j=0} \quad \boldsymbol{V}_{j-1} - \frac{\boldsymbol{V}_{j-1}\boldsymbol{x}_j\boldsymbol{x}_j^T\boldsymbol{V}_{j-1}}{4+\boldsymbol{x}_j^T\boldsymbol{V}_{j-1}\boldsymbol{x}_j}, && (58)\\
\ln|\boldsymbol{V}_j(\xi_j)| &=_{\xi_j=0} \quad \ln|\boldsymbol{V}_{j-1}| - \ln\left(1+\frac{1}{4}\boldsymbol{x}_j^T\boldsymbol{V}_{j-1}\boldsymbol{x}_j\right). && (59)
\end{aligned}$$

After this initial step, the parameters ξ_j, $\boldsymbol{V}_j(\xi_j)$, and $\boldsymbol{w}_j(\xi_j)$, are updated iteratively as described above until either $\mathcal{L}_j(\xi_j)$ changes less than 0.001% between two consecutive updates, over the number of iterations exceeds 100. The variational bound itself is, without the hyperprior, given by

$$\mathcal{L}_j(\xi_j) = \frac{1}{2}\boldsymbol{w}_j^T(\xi_j)\boldsymbol{V}_j^{-1}(\xi_j)\boldsymbol{w}_j(\xi_j) + \frac{1}{2}\ln|\boldsymbol{V}_j(\xi_j)| + \ln\sigma(\xi_j) - \frac{\xi_j}{2} + \lambda(\xi_j)\xi_j^2. \tag{60}$$

Batch Posterior Update

The scripts bayes_logit_fit .m and bayes_logit_fit_ard .m consider all inputs at once and optimise all ξ_n in combination. The difference between the two scripts is that the former operates without ARD, while the latter uses ARD. Both take inputs X and y, where X is an $N \times D$ matrix with $\boldsymbol{x}_n^T$ as its rows. y is a column vector containing the y_n's.

Let us first consider the version without ARD. In this version, all ξ_n are stored in the vector xi and are updated simultaneously. The script start by assuming $\xi_n = 0$ for all n, such that $\lambda(\xi_n) = 1/8$. Additionally, it pre-computes w_t $= \sum_n \boldsymbol{x}_n y_n/2$. The initial update of $\boldsymbol{V}_N(\boldsymbol{\xi})$, $\boldsymbol{w}_N(\boldsymbol{\xi})$, $b_N(\boldsymbol{\xi})$, and $\tilde{\mathcal{L}}(q,\boldsymbol{\xi})$ is computed outside of the loop. After that, the script iterates over first updating $\boldsymbol{\xi}$, then $b_N(\boldsymbol{\xi})$, followed by $\boldsymbol{V}_N(\boldsymbol{\xi})$ and $\boldsymbol{w}_N(\boldsymbol{\xi})$. The iteration stop if either $\tilde{\mathcal{L}}(q,\boldsymbol{\xi})$ does not change more than 0.001% between two consecutive iterations, or the number of iterations exceeds 100.

The script employs a few short-cuts and vectorisations which will be discussed here. In particular, the initial $\tilde{\mathcal{L}}(q,\boldsymbol{\xi})$ at $\boldsymbol{\xi} = \boldsymbol{0}$ is simplified by using

$$\sum_n\left(\ln\sigma(\xi_n) - \frac{\xi_n}{2} + \lambda(\xi_n)\xi_n^2\right) =_{\boldsymbol{\xi}=\boldsymbol{0}} -N\ln 2. \tag{61}$$

Also, as the scripts computes $\boldsymbol{V}_N(\boldsymbol{\xi})$ by inverting $\boldsymbol{V}_N^{-1}(\boldsymbol{\xi})$, it computes $\ln|\boldsymbol{V}_N(\boldsymbol{\xi})|$ from $\boldsymbol{V}_N^{-1}(\boldsymbol{\xi})$ for better stability, using $\ln|\boldsymbol{V}_N(\boldsymbol{\xi})| = -\ln|\boldsymbol{V}_N^{-1}(\boldsymbol{\xi})|$. In addition, the following vectorised operations are used:

$$\sum_n \frac{y_n}{2}\boldsymbol{x}_n = 0.5 * \textbf{sum}(\text{bsxfun(@times, X, y), 1)}', \tag{62}$$

$$2\sum_n \lambda(\xi_n)\boldsymbol{x}_n\boldsymbol{x}_n^T = 2 * \text{X}' * \text{bxsfun(@times, X, lam\_xi)}, \tag{63}$$

$$\boldsymbol{x}_n^T\left(\boldsymbol{V}_N + \boldsymbol{w}_N\boldsymbol{w}_N^T\right)\boldsymbol{x}_n = \left(\textbf{sum}(\text{X} .* (\text{X} * (\text{V} + \text{w} * \text{w}'))),\ 2)\right)_n . \tag{64}$$

$$\tag{65}$$

The ARD version of the code differs from the non-ARD version by the variables bn and E_a now being vectors rather than scalars. Their updates and use is adjusted accordingly, in line with what has been described above.

Predictive Density

Both scripts bayes_logit_post_iter .m and bayes_logit_post.m compute the predictive probability $p(y = 1|\boldsymbol{x}, \mathcal{D})$, the only difference being that the latter is a vectorised version of the former.

Let us firstly consider bayes_logit_post_iter .m. This script iterates over all given $\boldsymbol{x}$, as rows of the input X, and optimises $\boldsymbol{\xi}$ for each of those separately. In order to do so, it employs a simplification to logdetV_xi = $\ln|\tilde{\boldsymbol{V}}|/|\boldsymbol{V}_N|$, appearing in $\ln p(y = 1|\boldsymbol{x}, \mathcal{D})$. From the expression for $\tilde{\boldsymbol{V}}^{-1}$ and the Matrix determinant lemma it can be shown that

$$\ln|\tilde{\boldsymbol{V}}| = -\ln|\tilde{\boldsymbol{V}}^{-1}| = -\ln|\boldsymbol{V}_N^{-1}| - \ln\left(1 + 2\lambda(\xi)\boldsymbol{x}^T\boldsymbol{V}_N\boldsymbol{x}^T\right) = \ln|\boldsymbol{V}_N| - \ln\left(1 + 2\lambda(\xi)\boldsymbol{x}^T\boldsymbol{V}_N\boldsymbol{x}\right). \tag{66}$$

Thus, $\ln|\tilde{\boldsymbol{V}}|/|\boldsymbol{V}_N|$ results in

$$\ln\frac{|\tilde{\boldsymbol{V}}|}{|\boldsymbol{V}_N|} = -\ln\left(1 + 2\lambda(\xi)\boldsymbol{x}^T\boldsymbol{V}_N\boldsymbol{x}\right) \tag{67}$$

Additionally, the Sherman-Morrison formula can be applied to avoid inverting $\tilde{\boldsymbol{V}}^{-1}$ by using

$$\tilde{\boldsymbol{V}} = \boldsymbol{V}_N - \frac{2\lambda(\xi)\boldsymbol{V}_N\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{V}_N}{1 + 2\lambda(\xi)\boldsymbol{x}^T\boldsymbol{V}_N\boldsymbol{x}} \tag{68}$$

instead.

The script initially starts with $\xi = 0$ for each $\boldsymbol{x}$, using some initial simplifications based on $\lambda(\xi) = 1/8$, as already previously discussed. It then iterates over updating ξ, $\tilde{\boldsymbol{V}}$ and $\tilde{\boldsymbol{w}}$ until the variational bound either changes less than 0.001% between two consecutive iterations, or the number of iterations exceeds 100. The variational bound is computed as a simplified version of $\ln p(y = 1|\boldsymbol{x}, \mathcal{D})$, omitting all terms that are independent of ξ.

The vectorised script bayes_logit_post.m follows exactly the same principles, but optimises $\boldsymbol{\xi}$ for all $\boldsymbol{x}$ at the same time, by maximising a sum of the individual variational bounds.