Machine Learning and Bayesian Inference

Problem Sheet III: Bayesian neural networks and Gaussian processes

Sean B. Holden © 2010-17

1 The Bayesian approach to neural networks

1. **Slide 19**. Show that
$$\nabla\nabla\frac{1}{2}||\mathbf{w}||^2 = \mathbf{I}.$$

2. **Slide 22**. Show that
$$Z = (2\pi)^{W/2}|\mathbf{A}|^{-1/2}\exp(-S(\mathbf{w}_{\text{MAP}})).$$

3. For the next question we're going to need something known variously as the *matrix inversion lemma*, the *Woodbury formula* and the *Sherman-Morrison formula*, depending on the precise form used. In order to derive this we'll first need to know how to derive the formulae stated on slide 48 for *inverting a block matrix*.

   (a) We want to invert the block matrix
   $$\mathbf{\Sigma} = \begin{bmatrix}\mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12}\\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22}\end{bmatrix} \tag{1}$$
   to get
   $$\mathbf{\Sigma}^{-1} = \begin{bmatrix}\mathbf{\Lambda}_{11} & \mathbf{\Lambda}_{12}\\ \mathbf{\Lambda}_{21} & \mathbf{\Lambda}_{22}\end{bmatrix}. \tag{2}$$
   Show that
   $$\begin{aligned}\mathbf{\Lambda}_{11} &= (\mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21})^{-1}\\ \mathbf{\Lambda}_{12} &= -\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}\mathbf{\Lambda}_{22}\\ \mathbf{\Lambda}_{21} &= -\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{\Lambda}_{11}\\ \mathbf{\Lambda}_{22} &= (\mathbf{\Sigma}_{22} - \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12})^{-1}\end{aligned}$$
   (Hint: write $\mathbf{\Sigma}\mathbf{\Sigma}^{-1} = \mathbf{I}$ and solve the resulting equations. Note that these are different to the ones on slide 48, but you can re-arrange one version into the other.)

   (b) Now do the same thing again, this time solving $\mathbf{\Sigma}^{-1}\mathbf{\Sigma} = \mathbf{I}$. Show that
   $$\begin{aligned}\mathbf{\Lambda}_{12} &= -\mathbf{\Lambda}_{11}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\\ \mathbf{\Lambda}_{21} &= -\mathbf{\Lambda}_{22}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}.\end{aligned}$$

   (c) The two expressions for $\mathbf{\Lambda}_{21}$ must be equal. Equate them to show that
   $$(\mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21})^{-1} = \mathbf{\Sigma}_{21}^{-1}\mathbf{\Sigma}_{22}(\mathbf{\Sigma}_{22} - \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12})^{-1}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}.$$

Now write $\boldsymbol{\Sigma}_{21}^{-1}\boldsymbol{\Sigma}_{22}$ as

$$\boldsymbol{\Sigma}_{21}^{-1}\boldsymbol{\Sigma}_{22} = \boldsymbol{\Sigma}_{21}^{-1}(\boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}) + \boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}$$

and show that

$$(\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})^{-1} = \boldsymbol{\Sigma}_{11}^{-1} + \boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}(\boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12})^{-1}\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}.$$

This is the full version of the formula. Note that it is a method for *updating an existing inverse*: provided we know the inverse of $\boldsymbol{\Sigma}_{11}$, it tells us how to *update* that inverse when $-\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}$ is added to $\boldsymbol{\Sigma}_{11}$. We have to be able to calculate a different inverse, but crucially the new inverse might be *much simpler to calculate*. We shall see the extreme version of this in the last part of the question.

(d) Use the special case where $\mathbf{y}$ and $\mathbf{z}$ are vectors and

$$\boldsymbol{\Sigma} = \begin{bmatrix} \mathbf{X} & -\mathbf{y} \\ \mathbf{z}^T & 1 \end{bmatrix}$$

to show that

$$(\mathbf{X} + \mathbf{y}\mathbf{z}^T)^{-1} = \mathbf{X}^{-1} - \frac{\mathbf{X}^{-1}\mathbf{y}\mathbf{z}^T\mathbf{X}^{-1}}{1 + \mathbf{z}^T\mathbf{X}^{-1}\mathbf{y}}.$$

This is what we'll actually need in the next question.

4. Use the standard Gaussian integral to derive the final equation for Bayesian regression

$$p(Y|\mathbf{y};\mathbf{x},\mathbf{X}) = \frac{1}{\sqrt{2\pi\sigma_Y^2}} \exp\left(-\frac{(Y - h_{\mathbf{w}_{\text{MAP}}}(\mathbf{x}))^2}{2\sigma_Y^2}\right)$$

where

$$\sigma_Y^2 = \frac{1}{\beta} + \mathbf{g}^T\mathbf{A}^{-1}\mathbf{g}$$

given on slide 24. You might want to break this into steps:

(a) Write down the integral that needs to be evaluated. How does this compare to the standard integral result presented in the lectures? Can you make an immediate simplification? (Hint: the integral is over the whole of the space $\mathbb{R}^W$ where W is the number of weights. What happens to the value of an integral over all of $\mathbb{R}$ in 1 dimension if you just shift the integrand a bit to the left? If you can't see a simplification at this point you should still be able to complete the question, but it might be more complex.)

(b) Use the integral identity from the lectures to evaluate the integral.

(c) Does the expression you now have for $p(Y|\mathbf{y};\mathbf{x},\mathbf{X})$ look familiar? You should find that it looks like a Gaussian density. Extract expressions for the mean and variance.

(d) Use the matrix inversion lemma derived above to simplify the expression for the variance to give the final result presented in the lectures.

5. This question asks you to produce a version of the graph on slide 26, using the Metropolis algorithm. Any programming language is fine, although Matlab is probably the most straightforward.

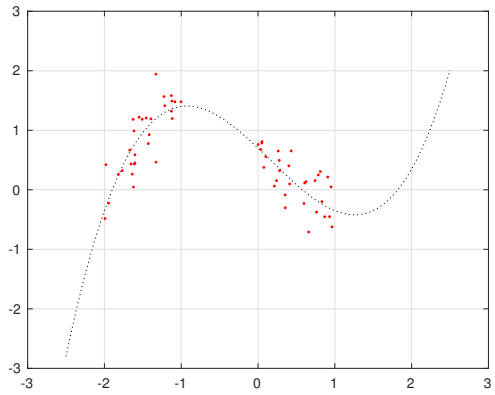
   The data is simple artificial data for a one-input regression problem. Use the target function

$$f(x) = x^3 - \frac{1}{2}x^2 - \frac{7}{2}x + 2$$

   and generate 30 examples in each of two clusters, one uniform in $[-2. - 1]$ and one uniform in $[0, 1]$. Then label these examples

$$y_i = f(x_i) + n$$

   where n is Gaussian noise of variance 0.1. You should have something like this:



   Let $\mathbf{w}$ be the weight vector and W the total number of weights in $\mathbf{w}$. You should use the prior and likelihood from the lectures, so

$$p(\mathbf{w}) = \left(\frac{2\pi}{\alpha}\right)^{-W/2} \exp\left(-\frac{\alpha}{2}||\mathbf{w}||^2\right)$$

   and

$$p(\mathbf{y}|\mathbf{w}; \mathbf{X}) = \left(\frac{2\pi}{\beta}\right)^{-m/2} \exp\left(-\frac{\beta}{2}\sum_{i=1}^{m}(y_i - h_{\mathbf{w}}(x_i))^2\right)$$

   where m is the number of examples and $h_{\mathbf{w}}(x)$ is the function computed by a suitable neural network with weights $\mathbf{w}$. Note that we are assuming that hyperparameters α and β are known; the values used to produce the lecture material were $\alpha = 1$ and $\beta = 10$.

   Complete the following steps:

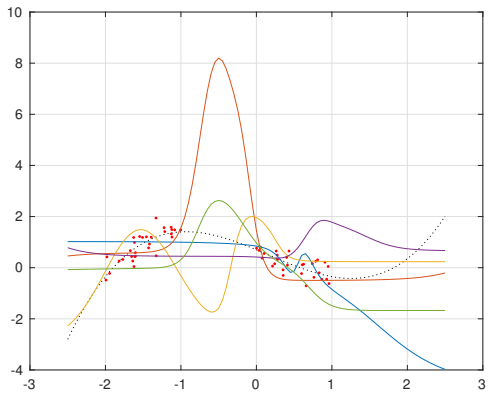
   (a) Write the code required to compute the prior and likelihood functions.

   (b) Implement a multilayer perceptron with a single hidden layer, a basic feedforward structure as illustrated in the AI I lectures, and a single output node. The network should use sigmoid activation functions for the hidden units and a linear activation function for its output. (The lecture material was produced using 5 hidden units.)

(c) Starting with a weight vector chosen at random, use the Metropolis algorithm to sample the posterior distribution $p(\mathbf{w}|\mathbf{y};\mathbf{X})$. You should generate a sequence $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N$ of N weight vectors. The lecture material used $N = 500,000$. However, note that you will probably find some degree of experimentation is required here, and it may be a good idea to start with a much smaller N while you explore parameter settings.

For example, you may find that an initial starting value for $\mathbf{w}_1$ is inappropriate, and you will find that the algorithm behaves differently for different step sizes taken when updating $\mathbf{w}_i$ to $\mathbf{w}_{i+1}$—try varying it and seeing how the proportion of steps accepted is affected. (The lecture material was produced using a step variance of 0.25.)

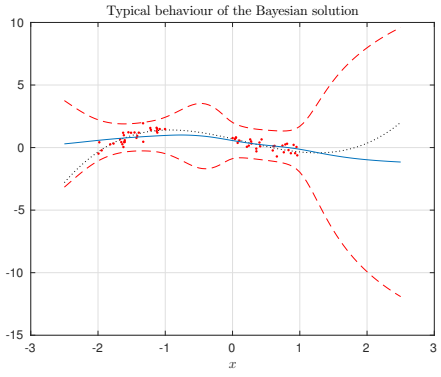
(d) Plot the function $h_{\mathbf{w}_i}(x)$ computed by the neural network for a few of the weight vectors obtained. You may see a surprising amount of variation in areas where there was no training data. (To see this it helps to take vectors from different areas in the sequence.)



(e) It takes a while for the Markov chain to settle in. Discard an initial chunk of the vectors generated. Using the remaining M, calculate the mean and variance of the corresponding functions using

$$\text{mean}(x) = \frac{1}{M}\sum_i h_{\mathbf{w}_i}(x)$$

and a similar expression to estimate the variance. Plot the mean function along with error bars at $\pm 2\sigma_Y^2$.



2 Gaussian processes

1. **Slide 44**: Show that when Gaussian noise is added as described

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I}).$$

2. **Slide 45, note 2**: what difference is made by the inclusion or otherwise of σ^2 in k?

3. **Slide 49**: provide the derivation for the final result

$$p(y'|\mathbf{y}) = \mathcal{N}(\mathbf{k}^T\mathbf{L}^{-1}\mathbf{y}, k - \mathbf{k}^T\mathbf{L}^{-1}\mathbf{k}).$$

3 Old exam questions

There are at present no old exam questions relevant to this section of the course.