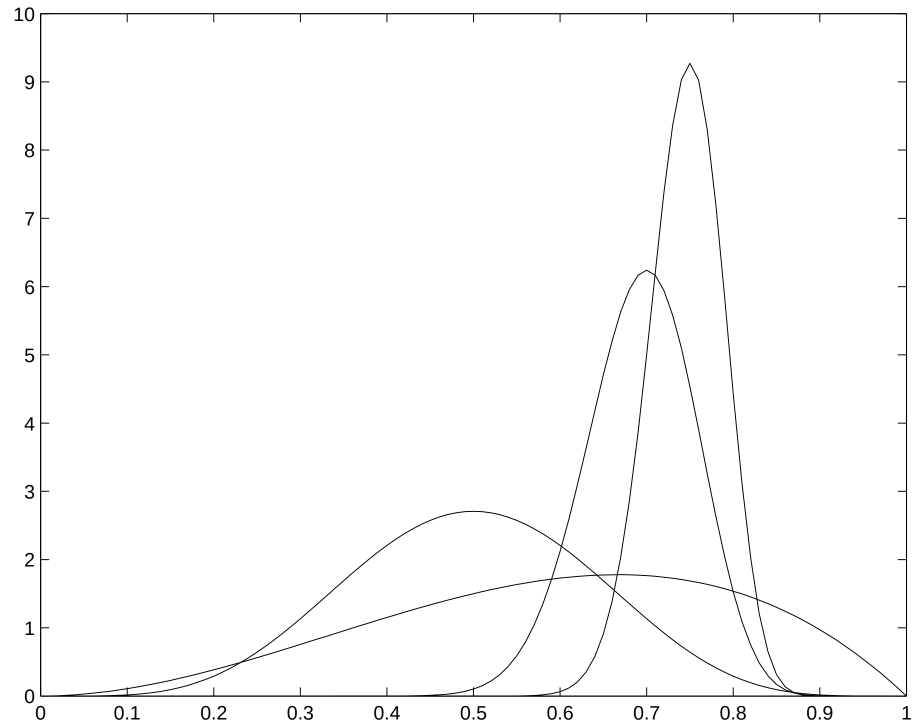
Computational tools I: Laplace approximation

Idea: For large n, the posterior will be concentrated around the MAP $\sim$ ML value $\widehat{\theta}$ and (for continuous θ) can be approximated by a Gaussian. This stems from the behaviour of the likelihood for large n.



Posterior density of θ for the biased coin for $n = 3, 10, 50, 100$. The true value under which the data were generated was $\theta = 0.7$.

$$\ln p(D|\theta) = \sum_{i=1}^{n} \ln p(x_i|\theta) = \sum_{i=1}^{n} \ln p(x_i|\widehat{\theta}) + \frac{c_2}{2} n \left(\theta - \widehat{\theta}\right)^2 + \frac{c_3}{3!} n \left(\theta - \widehat{\theta}\right)^3 + \ldots$$

with

$$c_k = \frac{1}{n} \sum_{i=1}^{n} \partial_\theta^k \ln p(x_i|\theta)_{|\widehat{\theta}} \approx E_x[\partial_\theta^k \ln p(x|\theta)_{|\widehat{\theta}}] = O(1)$$

Hence, in the posterior, the dominating term

$$p(\theta|D) \propto \exp\left[-\frac{-|c_2|}{2} n \left(\theta - \widehat{\theta}\right)^2\right] \left(1 + \frac{c_3}{3!} n \left(\theta - \widehat{\theta}\right)^3 + \ldots\right)$$

is a Gaussian and the corrections are small: With high posterior probability, we have $|\theta - \widehat{\theta}| \sim \frac{1}{\sqrt{n}}$ and $n \left|\theta - \widehat{\theta}\right|^3 \sim \frac{1}{\sqrt{n}}$.

Bayes asymptotics

For finite dimensional parametric models with continuous priors we have

$$p(\theta|D) \approx \mathcal{N}\left(\widehat{\boldsymbol{\theta}}, \mathbf{I}^{-1}(\widehat{\boldsymbol{\theta}})\right)$$

for $n \to \infty$, where $\widehat{\boldsymbol{\theta}}$ is the ML estimator and $\mathbf{I}_{ij}(\boldsymbol{\theta}) = -\partial_i \partial_j \sum_{k=1}^{n} \ln p(x_k|\boldsymbol{\theta})$. This should be compared to the asymptotic errors of ML estimation !

Laplace approximation

Compute integrals by Taylor expansion to 2nd order at maximum $\hat{\mathbf{z}}$.

$$\int e^{-h(\mathbf{z})}\, d\boldsymbol{\theta} \approx e^{-h(\hat{\mathbf{z}})} \int \exp\left[-\frac{1}{2}(\mathbf{z}-\hat{\mathbf{z}})^T \mathbf{A}(\mathbf{z}-\hat{\mathbf{z}})\right]\, d\mathbf{z}$$

$$= e^{-h(\hat{\mathbf{z}})} \frac{(2\pi)^{K/2}}{|\mathbf{A}|^{1/2}}$$

with $\mathbf{A} = \nabla^2 h(\hat{\mathbf{z}})$.

<u>Approximating the evidence</u>

$$-\ln p(D) = -\ln \int p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta} \approx -\ln p(D|\hat{\boldsymbol{\theta}}) - \ln p(\hat{\boldsymbol{\theta}}) - \frac{K}{2}\ln(2\pi) + \frac{1}{2}\ln|\mathbf{A}|$$

with $\mathbf{A} = -\nabla^2 \ln p(\hat{\boldsymbol{\theta}}|D)$ and $\hat{\boldsymbol{\theta}}$ is the MAP estimator.

<u>Further approximation:</u> Bayes Information Criterion(BIC) :

Use $|A| = O(N^K)$ and $\hat{\boldsymbol{\theta}} \approx \boldsymbol{\theta}_{ML}$

$$-\ln p(D) \approx -\ln p(D|\boldsymbol{\theta}_{ML}) + \frac{K}{2}\ln n$$

Posterior expectations

Approximate

$$\langle g(\boldsymbol{\theta}) \rangle \doteq E[g(\boldsymbol{\theta})|D] = \frac{\int e^{-h^*(\boldsymbol{\theta})} \, d\boldsymbol{\theta}}{\int e^{-h(\boldsymbol{\theta})} \, d\boldsymbol{\theta}}$$

with

$$\begin{aligned} -h^*(\boldsymbol{\theta}) &= \ln p(\boldsymbol{\theta}) + \ln p(D|\boldsymbol{\theta}) + \ln g(\boldsymbol{\theta}) \\ -h(\boldsymbol{\theta}) &= \ln p(\boldsymbol{\theta}) + \ln p(D|\boldsymbol{\theta}) \end{aligned}$$

and let $\widehat{\boldsymbol{\theta}}^*$, $\widehat{\boldsymbol{\theta}}$ the maximisers of h^* and h. Then

$$\langle g(\boldsymbol{\theta}) \rangle \approx \sqrt{\frac{\left|\nabla^2 h(\widehat{\boldsymbol{\theta}})\right|}{\left|\nabla^2 h^*(\widehat{\boldsymbol{\theta}}^*)\right|}} \exp\left[-h^*(\widehat{\boldsymbol{\theta}}^*) + h(\widehat{\boldsymbol{\theta}})\right]$$

Application: Bayesian Neural Networks

Consider neural network input-ouput

$$f_{\mathbf{w}}(\mathbf{x}) = \sum_j W_j \sigma(\mathbf{w}_j^T \mathbf{x})$$

where e.g. $\sigma(z) = \tanh(z)$.

Probabilistic model:

$$p(y|\mathbf{x}, \mathbf{w}) \propto \exp\left(-\frac{\beta}{2}(y - f_{\mathbf{w}}(\mathbf{x}))^2\right) \qquad \text{Regression}$$

$$p(y|\mathbf{x}, \mathbf{w}) = \left(\frac{1}{1 + e^{-f_{\mathbf{w}}(\mathbf{x})}}\right)^y \left(\frac{1}{1 + e^{f_{\mathbf{w}}(\mathbf{x})}}\right)^{1-y} \qquad \text{Classification}$$

Priors:

$$p(\mathbf{w}) \propto \exp\left(-\frac{1}{2}\sum_k \alpha_k ||\mathbf{w}_k||^2\right)$$

Approximate posterior (Regression)

Introduce

$$E_D = \sum_i (y_i - f_{\mathbf{w}}(\mathbf{x}_i))^2$$

$$E_W = ||\mathbf{w}||^2$$

and the minimiser as $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} (\beta E_D + \alpha E_W)$, we get the posterior approximation

$$p(\mathbf{w}|D) \propto e^{-(\beta E_D + \alpha E_W)} \approx \exp\left[-\frac{1}{2}\Delta\mathbf{w}^T \mathbf{A} \Delta\mathbf{w}\right]$$

where $\Delta\mathbf{w} = \mathbf{w} - \hat{\mathbf{w}}$ and $\mathbf{A} = \beta\nabla^2 E_D^{MP} + \alpha\mathbf{I}$

<u>Approximate Predictive distribution</u>

Linearise $f_{\mathbf{w}}(\mathbf{x}) \approx f_{\hat{\mathbf{w}}}(\mathbf{x}) + \mathbf{g}^T \Delta\mathbf{w}$

$$p(y|x, D) \approx C \int p(y|x, \mathbf{w}) \exp\left[-\frac{1}{2}\Delta\mathbf{w}^T \mathbf{A} \Delta\mathbf{w}\right] \approx \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - y_{MP})^2}{2\sigma^2}\right)$$

with $y_{MP} = f_{\hat{\mathbf{w}}}(\mathbf{x})$ and $\sigma^2 = \frac{1}{\beta} + \mathbf{g}^T \mathbf{A}^{-1} \mathbf{g}$.

Evidence approximation

$$-\ln p(D|\alpha, \beta) = \beta E_D^{MP} + \alpha E_W^{MP} + \frac{1}{2}\ln|\mathbf{A}| - \frac{W}{2}\ln\alpha - \frac{n}{2}\ln\beta + \frac{n}{2}\ln(2\pi)$$

Estimate hyperparameters:

Compute $\gamma = \sum_{k=1}^{W} \frac{\lambda_k}{\lambda_k+\alpha}$, where the λ_k are eigenvalues of $\beta\nabla^2 E_D^{MP}$. Start with some values of α and β, optimise $\hat{\mathbf{w}}$ and re-estimate

$$\alpha^{new} = \frac{\gamma}{2E_W}$$
$$\beta^{new} = \frac{n-\gamma}{2E_D}$$

optimise $\hat{\mathbf{w}}$ and repeat until convergence.

ARD: The method can be extended to separate α_ks for each input neuron. Large α_k leads to a 'shut off' for the corresponding weights.

Example

Artificial data set: *Friedman data* generated as

$$y(\mathbf{x}) = 0.1e^{4x_1} + \frac{4}{1 + e^{-20(x_2 - \frac{1}{2})}} + 3x_3 + 2x_4 + x_5 + 0 \cdot \sum_{i=6}^{10} x_i + \nu$$

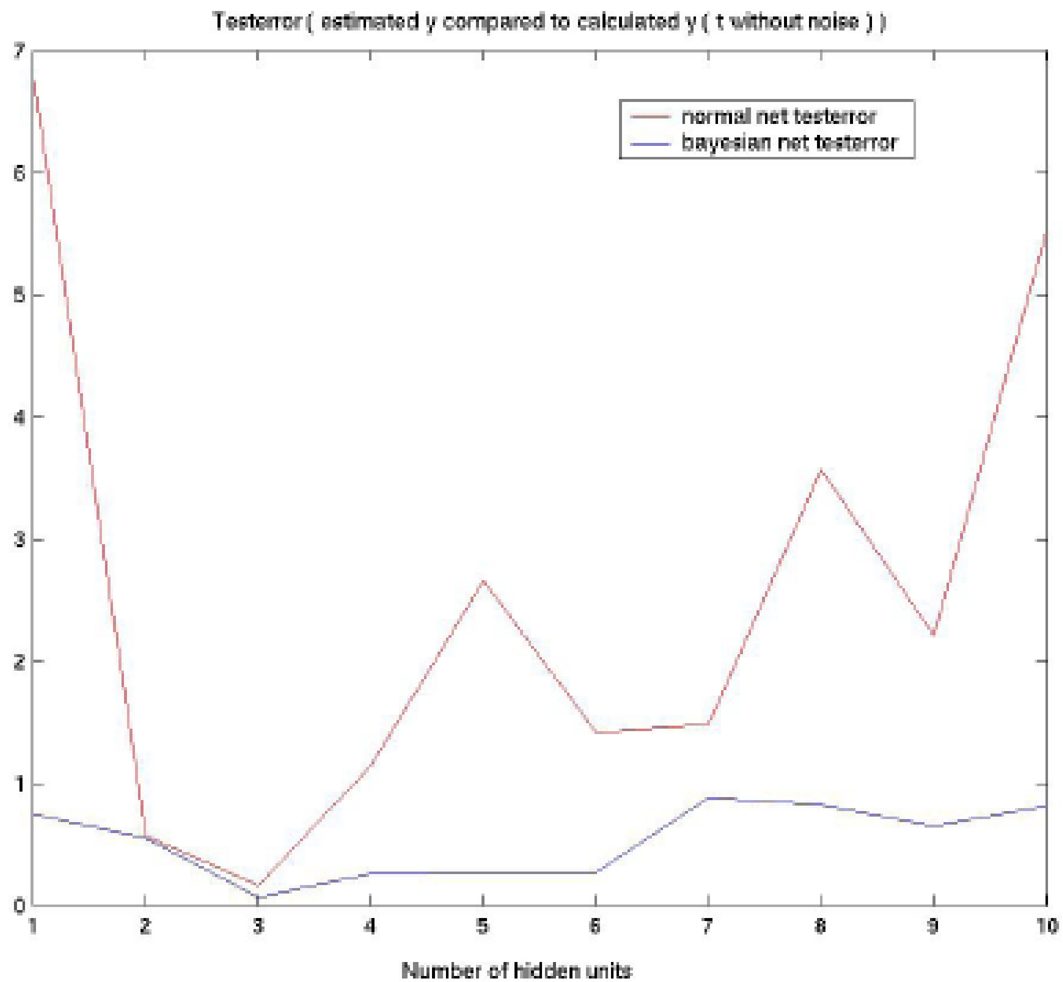


Figure 6: 200 training samples, 30 training loops, 30 evidence-iterations

ARD: α_i for network inputs x_i:

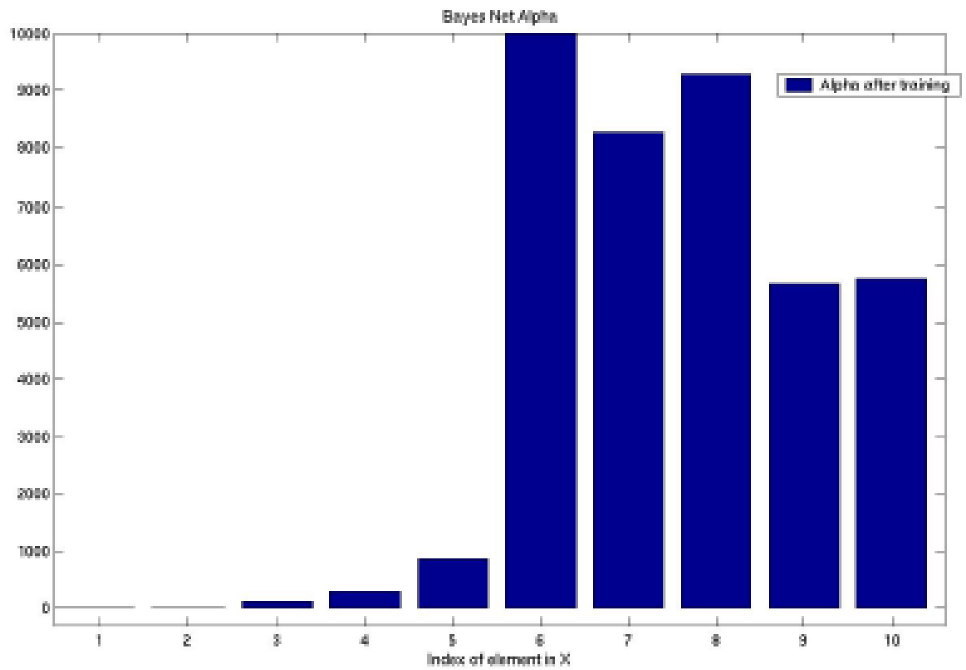


Figure 5: 5 hidden units, 200 training samples, 100 training loops, 50 evidence-iterations, zoomed into diagram

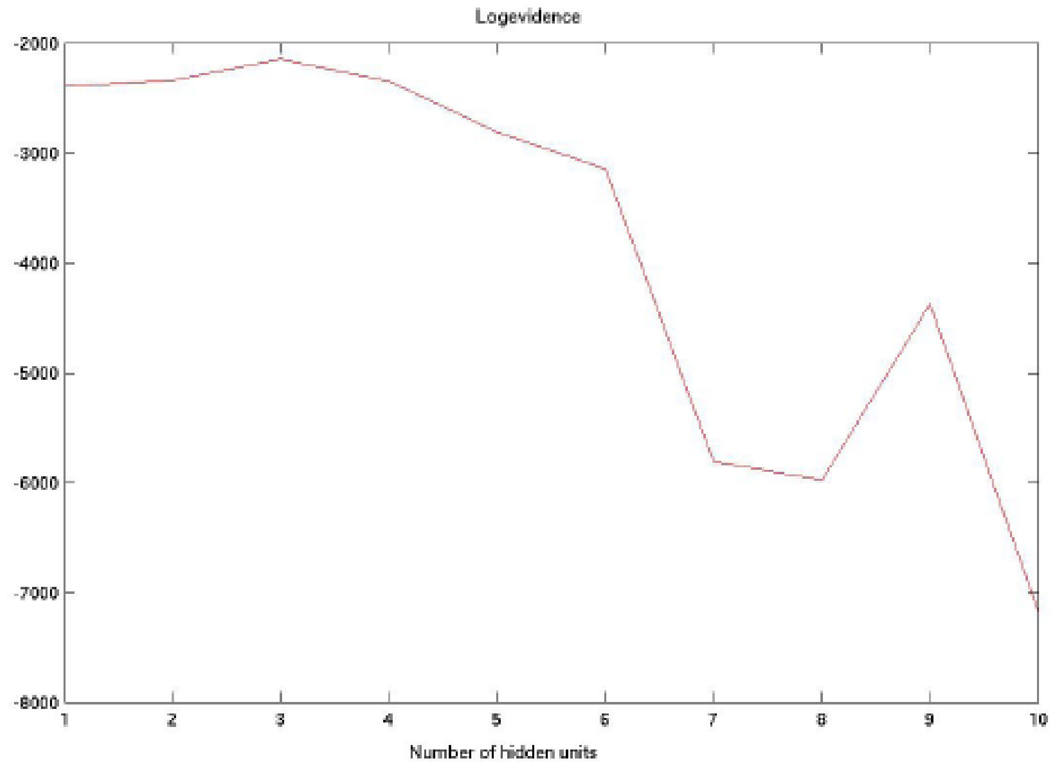


Figure 7: 200 training samples, 30 training loops, 30 evidence-iterations

Summary: Laplace approximation

- Approximates posterior (log posterior) by a Gaussian (2nd order Taylor expansion around MAP value).

- Becomes exact for large number of data for finite dimensional models with continuous parameters (under technical conditions).

- Advantages: Integration is replaced by optimisation, i.e. by finding the MAP. The Hessian which is required for the covariance can also be used for a Newton Raphson algorithm.

- Disadvantages: local approximation, takes into account only MAP and curvature. Ignores other posterior modes. Can't be used for discrete variables.