Neural Networks, Model Averaging and Model Choice

B. D. Ripley

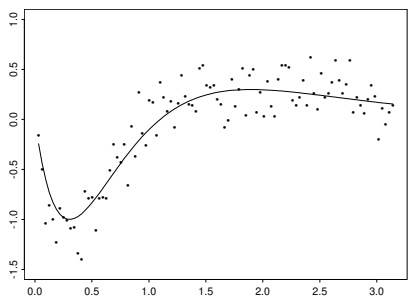
Professor of Applied Statistics
University of Oxford

Programme

- Complex models such as neural networks
- Regularization
- Performance criteria
- AIC, NIC, p_{eff}, …
- Cross-validation
- Bayes factors
- Combining models

Demonstration example

Taken from Wahba & Wold (1975) where it illustrates smoothing splines.



100 data points from a curve-fitting problem, with the true curve.

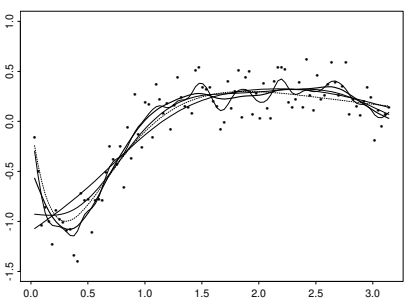
Throughout we assume that the true noise variance, $(0.2)^2$, is known.

Smoothing Splines

Numerical analysts know that very flexible families of functions will overfit, so they add a penalty, for example

$$\sum_{i=1}^{n}[y_i - f(x_i)]^2 + \lambda \int f''(u)^2\, du$$

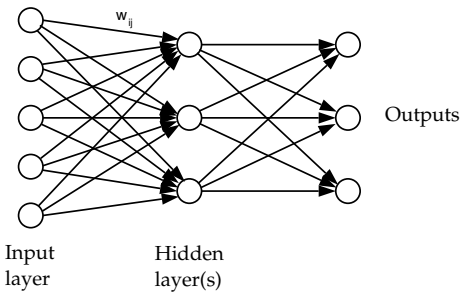
This is minimized over *all* functions f by a (cubic) smoothing spline. It remains to choose λ, which controls the degree of smoothness.



Smoothing splines with various degrees of regularization. The bold line was chosen as the best fit.

Neural Networks

Only consider the simplest possible cases.



A generic feed-forward neural network.

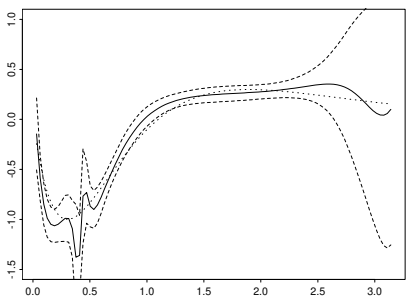
In mathematics:

$$y_k = \alpha_k + \sum_j w_{jk}\,\phi_h\Big(\alpha_j + \sum_i w_{ij}\,x_i\Big)$$

where ϕ_h is the logistic function $\phi_h(x) = e^x/(1+e^x)$
the 'biases' α_i can be replaced by weights to $+1$
the number of *hidden units* can vary.

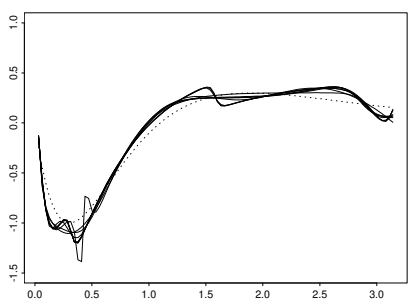
Fitting Regression Neural Networks

Treat as a non-linear regression, and fit by least-squares, varying the parameters (the weights and biases). Model choice abouts to choosing the number of hidden units.



Curve fitted by 1-8-1 neural network, with ± 2 standard error bands and true curve (dotted).

One problem is multiple local minima:



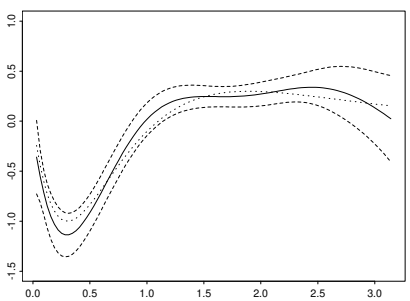
1-8-1 neural networks with different starting weights.

We clearly have over-fitting. Many ways to avoid this. Perhaps the best is to use *regularization* as used for smoothing splines. Somewhat easier for neural networks to use *weight decay*:

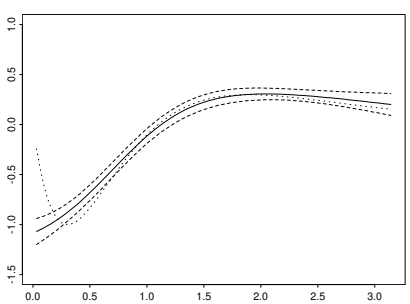
$$\sum_{i=1}^{n}[y_i - f(x_i; \mathbf{w})]^2 + \lambda \sum_{ij} w_{ij}^2$$

with the inputs and outputs suitably scaled. Generalizes *ridge regression*.

Effect of Weight Decay

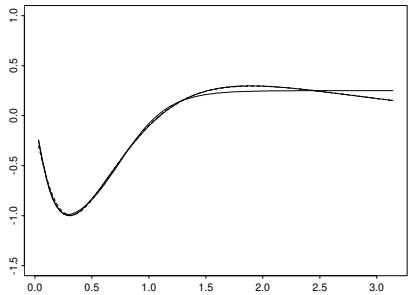


Curve fitted by 1-8-1 neural network, with ± 2 standard error bands and true curve $\lambda = 10^{-3}$ (above) and $\lambda = 0.1$ (below).



Bias/variance Compromises

We now have two ways to control the fit, the number of hidden units and λ, the degree of regularization. The true curve is not exactly representable by our model, so we have some bias.



Fitting to the true curve with 2 and 3 hidden units, and 8 units with $\lambda = 10^{-3}$.

Choosing fewer hidden units leads to more bias, as does adding a regularizer. But it also reduces their variability, so reduces the mean-square error. Can manipulate λ and the number of hidden units as a function of n to get the right compromise for very large n. Doesn't help in real problems!

AIC, NIC, p_{eff}

Akaike's (1973, 4) 'An Information Criterion'.
ML measure of fit is the deviance

$$2 \sum \log p(X_i, Y_i) / p(X_i, Y_i; \widehat{\theta})$$

on the training set (the supplied data). Consider a test set, that is an independent sample from the same process of the same size. We would expect the deviance for the test set to be larger.

AIC corrects this bias:

$$E\left[\text{deviance}_{\text{test}} - \text{deviance}_{\text{train}}\right] \sim 2p$$

for p parameters and for large n *under the model*. Despite frequent statements in the Bayesian literature, this does take the variability of $\widehat{\theta}$ into account (that's where the second p comes from). Choose model(s) which have small values of

$$\text{deviance}_{\text{train}} + 2p$$

Since AIC is based on maximum likelihood estimations, it has nothing to say in our example if regularization is used.

If we drop the requirements that the model is true, and that fitting is by ML, we find

$$E\left[\text{deviance}_{\text{test}} - \text{deviance}_{\text{train}}\right] \sim 2p_{\text{eff}}$$

where $p_{\text{eff}} = \text{tr}[KJ^{-1}]$ and

$$J = -E\frac{\partial^2 \log p(X_i, Y_i; \theta_0)}{\partial \theta \, \partial \theta^T}, \; K = \text{Var}\frac{\partial \log p(X_i, Y_i; \theta_0)}{\partial \theta}$$

and θ_0 is the 'least false' value of θ, that is minimizes

$$-E \log p(X, Y; \theta)$$

This is due to Moody (1991, 1992) and Murata *et al.* (1991,3,4).

We then have to estimate J and K. Obvious estimators are the sample averages (with divisors $n - p$ for J and n for K), replacing θ_0 by $\widehat{\theta}$. Moody's p_{eff} is equivalent (up to divisors) for the least-squares measure of fit.

In our demonstration example $p = 25$, p_{eff} is uncomputable for no weight decay (J is too close to singular), 18 for $\lambda = 10^{-3}$ and 6 for $\lambda = 0.1$. This gives estimates of the test-set sum of squares as 4.67 for $\lambda = 10^{-3}$ and 5.90 for $\lambda = 0.1$.

Note that AIC, NIC only correct the bias as an estimate of the expected test-set deviance. We have to consider the fluctuations. These can be of order $\sqrt{n}$, although the deviance is of order n. Thus if two test-set deviances are the same (they will be if both models are true), the fluctuations can swamp the correction term. If the models are *nested*, the fluctuations are still of the same order as the correction term. Thus for large n, AIC-like procedures never choose too small a model, but may choose too large a one.

Derivation of A/NIC

Let $i(x, \theta)$ be the Hessian of the likelihood for just one sample, and $d(p, p_\theta) = \mathsf{E}_p \log \frac{p(x)}{p_\theta(x)}$. We approximate

$$D = 2\mathsf{E}[\log p(X) - \log p(X, \widehat{\theta})]$$

via the Taylor expansion

$$\begin{aligned} 2\log p(x, \widehat{\theta}) &= 2\log p(x, \theta_0) + 2\tfrac{\partial \log p(x, \widehat{\theta})}{\partial \theta} \\ &\quad + (\widehat{\theta} - \theta_0)^T i(x, \widetilde{\theta})(\widehat{\theta} - \theta_0) \\ &= 2\log p(x, \theta_0) + 2\tfrac{\partial \log p(x, \widehat{\theta})}{\partial \theta} \\ &\quad + \mathrm{tr}\big[i(x, \widetilde{\theta})(\widehat{\theta} - \theta_0)(\widehat{\theta} - \theta_0)^T\big] \end{aligned}$$

so since $\widehat{\theta} \sim N(\theta_0, J^{-1}KJ^{-1}/n)$,

$$\begin{aligned} D &\approx 2\,d(p, p_{\theta_0}) - 2\,\mathsf{E}\tfrac{\partial \log p(X, \widehat{\theta})}{\partial \theta} \\ &\quad - \mathsf{E}\,\mathrm{tr}[i(X, \theta_0)(\widehat{\theta} - \theta_0)(\widehat{\theta} - \theta_0)^T] \\ &\approx 2\,d(p, p_{\theta_0}) + \mathrm{tr}[J\,\mathrm{Var}(\widehat{\theta})] \\ &= 2\,d(p, p_{\theta_0}) + \tfrac{1}{n}\mathrm{tr}[JJ^{-1}KJ^{-1}] \\ &= 2\,d(p, p_{\theta_0}) + \tfrac{1}{n}\mathrm{tr}[KJ^{-1}] \end{aligned}$$

We will call the trace p_{eff}. It is p if the model is correct.

On the other hand, for the training set

$$\begin{aligned} &2\sum \log \tfrac{p(X_i)}{p(X_i, \theta_0)} \\ &\approx 2\sum \log \tfrac{p(X_i)}{p(X_i, \widehat{\theta})} - \sum \mathrm{tr}\big[i(X_i, \theta_0)(\widehat{\theta} - \theta_0)(\widehat{\theta} - \theta_0)^T\big] \\ &\approx 2\sum \log \tfrac{p(X_i)}{p(X_i, \widehat{\theta})} + p_{\text{eff}} \end{aligned}$$

so we have

$$n \times D \approx \mathsf{E}\,\text{deviance} + 2\,p_{\text{eff}}$$

and the LHS is the expectation of the test-set deviance.

Argument, but not the conclusion, is due to M. Stone (1977).

Cross-validation

If we had an independent test set, the obvious way to choose the model would be via test-set performance (not necessarily the deviance). Use cross-validation instead.

Split the training set into V parts. Use $V-1$ of these to train, and test the performance on the Vth part. Do this for all V parts. This creates a pseudo test set of size n. The performance measure is (almost) unbiased, since the parts are (almost) independent, but it is for performance at size $(1 - 1/V)n$.

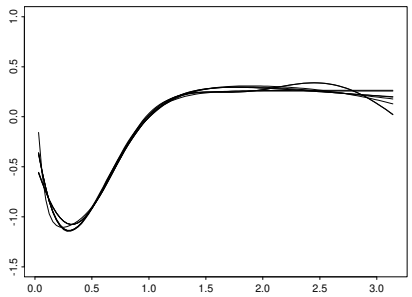
How do we choose V? Traditionally *leave-one-out*, so $V = n$. This can be very variable. Often better to choose $V \approx \sqrt{n}$ or 10, and much cheaper computationally.

Can correct for the effective sample size $(1 - 1/V)n$ by the *jackknife*, but effect is very similar to LOO CV.

Asymptotically LOO CV is the same as NIC, and often accurately so in small samples too. NIC is much less work!

Results for Example

Size	λ	SSq	SSq_{CV}	p_{eff}	NIC	Bayes
2	0	3.47	4.29	7	4.03	—
2	10^{-3}	3.57	4.78	5	3.98	3.30
3	0	4.12	4.81	10	4.92	—
3	10^{-3}	3.40	4.54	6	3.93	3.30
3	10^{-2}	3.61	4.91	6	4.11	3.72
3	10^{-1}	5.40	6.20	6	5.94	6.08
5	0	3.14	4.57	16	4.42	—
5	10^{-3}	3.21	4.51	17	4.59	3.38
5	10^{-2}	3.61	4.91	6	4.11	4.11
5	10^{-1}	5.38	6.18	6	5.92	5.98
8	0	2.89	5.35	25	4.89	—
8	10^{-3}	3.21	4.54	18	4.67	4.74
8	10^{-2}	3.58	4.94	7	4.16	4.63
8	10^{-1}	5.36	6.14	7	5.90	5.83



The fitted curves for all the well-supported models.

'Plug-in' *vs* 'Predictive'

So far we have show $f(x; \widehat{\mathbf{w}})$. It would be better to take the uncertainty in $\widehat{\mathbf{w}}$ into account. Work with distributions, not just means. Let $\mathcal{T}$ denote the training set.

In a Bayesian approach need a prior $p(\theta)$. With regularizer $C(f)$ corresponds to $p(\theta) \propto \exp -\lambda C(f)$. Often have a family of priors as here, so take a hyperprior over λ.

We don't know $p(y \mid x)$, but should use our best estimate $p(y \mid x, \mathcal{T})$. This is the *predictive* approach. Often easier to use the 'plug-in' estimator $p(y \mid x, \widehat{\theta})$ for the MLE, posterior mode, regularized fit, ..., rather than

$$p(y \mid x, \mathcal{T}) = \int p(y \mid x; \theta) p(\theta \mid \mathcal{T}, x)\, d\theta$$

$$p(\theta \mid \mathcal{T}, x) \propto \prod_{i=1}^{n} p(y_i \mid x_i; \theta) p(x_i; \theta)\, p(\theta) p(x; \theta)$$

Note that if there are multiple local minima, the predictive approach automatically takes a weighted average of their predictions.

Predictive Inference for NNs

My attempt to approximate the predictive approach. Let $E(\theta)$ be the log-posterior, the sum of the log-likelihood and log-prior (regularization penalty, e.g. from weight decay). Consider a classification problem.

(1) Sample from prior over models (# hidden units). Assume λ has a proper prior; may need to sample it too.

(2) For each sampled model search for local minima from many (100?) random starting points.

(3) At each local minimum compute the Hessian H of the log-posterior probability, and approximate the posterior $p(\theta \mid \mathcal{T})$ by $N(\widehat{\theta}, H^{-1})$.

(4) Weight the local minima by integration over the peak, assuming that the Gaussian is valid far enough out. This gives weights within a model proportional to

$$|H(\hat{\theta})|^{-1/2} \exp -E(\hat{\theta})\ (2\pi)^{\#\,\mathrm{wts}/2}$$

Remember symmetries in weight space over hidden units.

(5) Use the weights at (**??**) to choose (10,000?) random samples from the posterior over θ and models, then compute and average $p(k \mid x, \theta)$. Can do this more accurately using importance sampling.

For a single model this amounts to approximating the posterior for θ by a mixture of Gaussians. Normally the effect of the variation about each peak is small ($p(k \mid x, \theta)$ is not often very non-linear), but the effect of including multiple peaks is often large.

A regression problem is a bit easier, as step (5) can be approximated analytically. In our demo example (with $\lambda > 10^{-4}$) we found only a single local minimum. The mean of the predictive distribution was close to the 'plug-in' prediction, and the standard errors quite similar to the approximations (based on linearization) shown earlier, *but* importance sampling had to be used.

Bayes factors

The predictive approach assesses models via $P(\text{model} \,|\, \mathcal{T})$. Now

$$P(\text{model} \,|\, \mathcal{T}) \propto P(\text{model}) P(\mathcal{T} \,|\, \text{model})$$

so we need prior opinions about models too. Strictly we average predictive distributions with weights $P(\text{model} \,|\, \mathcal{T})$. But if we choose, a crucial factor is

$$B_{st} = \frac{P(\mathcal{T} \,|\, \text{model}_s)}{P(\mathcal{T} \,|\, \text{model}_t)}$$

called the Bayes factor, since

$$\frac{P(\text{model}_s \,|\, \mathcal{T})}{P(\text{model}_t \,|\, \mathcal{T})} = B_{st} \frac{P(\text{model}_s)}{P(\text{model}_t)}$$

Often

$$\begin{aligned} \log P(\mathcal{T} \,|\, \text{model}) \;\approx\; & \log \ell(\widehat{\theta}; \mathcal{T}) + \log p(\widehat{\theta}) \\ & + \tfrac{p}{2} \log 2\pi - \tfrac{1}{2} \log |H| \end{aligned}$$

for H the Hessian of the sum of the first two terms, so can be approximated. Log Bayes factors typically grow at rate $(p_s - p_t) \log n$, so penalize *all* extra parameters more severely than NIC.

Model Averaging

[M. Stone (1974), D. H. Wolpert (1992)]

Have a range of competing models. The predictive approach averages across models, not chooses one. Can we avoid the integrations?

Use cross-validation to choose a new classifier combining their outputs.

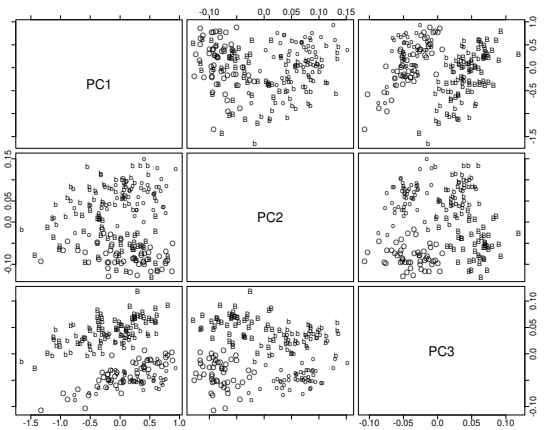
At its simplest, use cross-validation to choose a linear combination of predictions from models, but can allow combination to vary (slowly) with x. Wolpert calls such 'level 1 generalizers' 'dumb', but they are all that the Bayesian framework requires.

More formally, use leave-one-out cross-validation (or bootstrapping) to find (averaged) posterior class probabilities for each case when removed, for each model, then use these as inputs to a simple classifier.

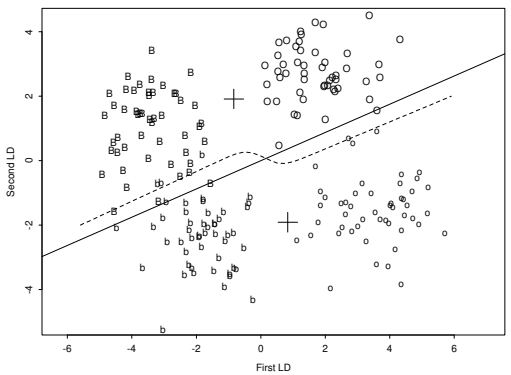
Treat each local minimum as a model in this approach; deleting just one observation *can* change completely the structure of local minima.

Sex of *L. variegatus* Crabs

Assume just the shell is given. Linear discriminant analysis ignoring forms finds a variable which is essentially $\mathtt{CL}^3\mathtt{RW}^{-2}\mathtt{CW}^{-1}$, a dimensionally neutral quantity. Six errors are made, all for the blue form.



Pairs plot of the first three principal components of the crabs data. Males are coded as capitals, females as lower case, colours as the initial letter of blue or orange.



Linear discriminants for the crabs data. Males are coded as capitals, females as lower case, colours as the initial letter of blue or orange. The crosses are the group means for a linear discriminant for sex (solid line) and the the dashed line is the decision boundary for sex based on four groups.

Performance

methods	**error rate** %	
linear discriminant	3.5	2.8
ditto, using colour form	2.0	2.8
logistic discrimination	3.0	1.6
neural network with 2 hidden	2.0	1.9

Cross-validated error rates from counts and posterior probabilities, respectively. (From only 200 observations, so standard errors of about 0.5%.)

Multiple minima

Take two hidden units. Find 5 local minima, which have smoothed cross-validated error rates of

6.1%, 1.9%, 2.3%, 1.9%, 2.3%

Predictive approach

Only local minima 3 and 5 are given appreciable weight, neither of which was that used above. In fact the lowest penalized minus log likelihood was 8.46, the two solutions given appreciable weight had 9.98 (63%) and 10.30 (35%).

Cross-validating the procedure gave error rates of 2.0% and 2.2% – little difference.

Model averaging

We can apply the predictive procedure to different λ and n_h, the number of hidden units. The table gives minus log Bayes factors (up to a constant).

	# hidden units			
λ	0	1	2	4
10^{-3}	7.5	3.5	2.8	3.8
10^{-4}	0.15	1.4	0.3	9.8
10^{-5}	0.0	0.0	0.7	—
10^{-6}	4.0	4.3	—	—

'—' indicates integration difficulties. Support for small λ reflects collinearity in the input variables.

Examples such as this show that assuming $p(\lambda \mid \mathcal{T})$ is highly peaked (*à la* MacKay) has almost no experimental support. Indeed one dataset gives one value of θ, and so can only fix λ precisely if the model effectively has many independent copies of parts of θ.