ON COMPARISON OF ADAPTIVE REGULARIZATION METHODS

Sigurdur Sigurdsson, Jan Larsen and Lars Kai Hansen
Department of Mathematical Modeling, Building 321
Technical University of Denmark, DK-2800 Lyngby, Denmark
Phone: +45 4525 3920,3923,3889
Email: siggi,jl,lkhansen@imm.dtu.dk, Web: eivind.imm.dtu.dk

INTRODUCTION

Modeling with flexible models, such as neural networks, requires careful control of the model complexity and generalization ability of the resulting model which finds expression in the ubiquitous bias-variance dilemma [4].

Regularization is a tool for optimizing the model structure reducing variance at the expense of introducing extra bias. The overall objective of adaptive regularization is to tune the amount of regularization ensuring minimal generalization error. Regularization is an supplement to direct model selection techniques like step-wise selection and one would prefer a hybrid scheme; however, a very flexible regularization may substitute the need for selection procedures.

This paper investigates recently suggested adaptive regularization schemes. Some methods focus directly on minimizing an estimate of the generalization error (either algebraic or empirical) [1], [3], [5], [6], [7], [12], [13], whereas others starts from different criteria, e.g., the Bayesian evidence [2, Ch. 10], [7], [15], [16]. The evidence expresses basically the probability of the model, which is conceptually different from generalization error; however, asymptotically for large training data sets they will converge[1] [15].

The papers is organized as follows: first the basic model definition, training and generalization is presented. Next, different adaptive regularization schemes are reviewed and extended. Finally, the experimental section presents a comparative study concerning linear models and feed-forward neural networks models for regression/time-series problems.

TRAINING AND GENERALIZATION

Suppose that our model, $\mathcal{M}$ (e.g., neural network), is described by the function $\boldsymbol{f}(\boldsymbol{x};\boldsymbol{w})$ where $\boldsymbol{x}$ is the input vector and $\boldsymbol{w}$ is the vector of parameters

[1] Up to a scaling factor and an additive constant.

(or weights) with dimensionality m. The objective is to use the model for approximating the true conditional input-output distribution $p(\boldsymbol{y}|\boldsymbol{x})$, or some moments thereof. For regression and signal processing problems we normally model the conditional expectation $E\{\boldsymbol{y}|\boldsymbol{x}\}$. Define the training set $\mathcal{T} = \{\boldsymbol{x}(k); \boldsymbol{y}(k)\}_{k=1}^{N_\mathcal{T}}$ of $N_\mathcal{T}$ input-output examples sampled from the unknown but fixed joint input-output probability density $p(\boldsymbol{x}, \boldsymbol{y})$. The model is trained by minimizing a cost function, $C_\mathcal{T}(\boldsymbol{w})$, which is usually the sum of a loss function (or training error), $S_\mathcal{T}(\boldsymbol{w})$, and a regularization term $R(\boldsymbol{w}, \boldsymbol{\kappa})$ parameterized by a set of regularization parameters $\boldsymbol{\kappa}$,

$$C_\mathcal{T}(\boldsymbol{w}) = S_\mathcal{T}(\boldsymbol{w}) + R(\boldsymbol{w}, \boldsymbol{\kappa}) = \frac{1}{N_\mathcal{T}} \sum_{k \in \mathcal{T}} \ell\left(\boldsymbol{y}(k), \widehat{\boldsymbol{y}}(k; \boldsymbol{w})\right) + R(\boldsymbol{w}, \boldsymbol{\kappa}) \qquad (1)$$

where $\ell(\cdot)$ measures the cost of estimating the output $\boldsymbol{y}(k)$ with the model prediction $\widehat{\boldsymbol{y}}(k; \boldsymbol{w}) = \boldsymbol{f}(\boldsymbol{x}(k); \boldsymbol{w})$, e.g., log-likelihood loss or the simple squared error loss function $\ell = \|\boldsymbol{y} - \widehat{\boldsymbol{y}}\|^2$. Often we will consider linear regularization, i.e., $R(\boldsymbol{w}, \boldsymbol{\kappa}) = \boldsymbol{\kappa}^\top \boldsymbol{r}(\boldsymbol{w}) = \sum_{i=1}^{q} \kappa_i r_i(\boldsymbol{w})$ where $r_i(\boldsymbol{w})$ are associated regularization functions. Many suggested regularizers are linear; this includes the popular weight decay regularization and regularizers imposing smooth functions such as the Tikhonov regularizer [2]. Training provides the estimated weight vector $\widehat{\boldsymbol{w}} = \operatorname{argmin}_{\boldsymbol{w}} C_\mathcal{T}(\boldsymbol{w})$. The *Generalization error* is defined as the expected loss on a future independent sample $(\boldsymbol{x}, \boldsymbol{y})$,

$$G(\widehat{\boldsymbol{w}}) = E_{\boldsymbol{x},\boldsymbol{y}}\{\ell(\boldsymbol{y}, \widehat{\boldsymbol{y}}(\widehat{\boldsymbol{w}}))\} = \int \ell(\boldsymbol{y}, \widehat{\boldsymbol{y}})\, p(\boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{x} d\boldsymbol{y}, \qquad (2)$$

and the *average generalization error* Γ is defined by averaging $G(\widehat{\boldsymbol{w}})$ over all possible training sets:[2] $\Gamma = E_\mathcal{T}\{G(\widehat{\boldsymbol{w}})\} = \int G(\widehat{\boldsymbol{w}})\, p(\mathcal{T})\, d\mathcal{T}$.

ADAPTIVE REGULARIZATION

Validation Error Approach

Adapting regularization so as to minimize an empirical estimate of the generalization error, viz. the K-fold cross-validation [18], leads to an adaptive regularization scheme originally suggested in [12], which was further improved in [1], [3], [5], [13]. Suppose that all available data $\mathcal{D} = \{\boldsymbol{x}(k); \boldsymbol{y}(k)\}_{k=1}^{N}$ of N input-output examples, split into K randomly chosen disjoint sets of approximately equal size, i.e., $\mathcal{D} = \cup_{j=1}^{K} \mathcal{V}_j$ and $\forall\, i \neq j : \mathcal{V}_i \cap \mathcal{V}_j = \emptyset$. Training and validation is replicated K times, and in the j'th run training is done on the set $\mathcal{T}_j = \mathcal{D} \backslash \mathcal{V}_j$ and validation is performed on $\mathcal{V}_j$. The K-fold cross-validation estimate is then given by the average validation error estimates,

$$\widehat{\Gamma}_{\text{cv}} = \frac{1}{K} \sum_{j=1}^{K} S_{\mathcal{V}_j}(\widehat{\boldsymbol{w}}_j), \quad S_{\mathcal{V}_j}(\widehat{\boldsymbol{w}}_j) = \frac{1}{N_{vj}} \sum_{k \in \mathcal{V}_j} \ell\left(\boldsymbol{y}(k), \widehat{\boldsymbol{y}}(k); \widehat{\boldsymbol{w}}_j\right) \qquad (3)$$

[2] For more details on empirical generalization error, generalization error distribution and average generalization error, see e.g., [11].

where $\widehat{\boldsymbol{w}}_j$ are the weights estimated from training set $\mathcal{T}_j$. N_{vj} is number of validation examples. $\widehat{\Gamma}_{\mathrm{cv}}$ is an estimate of the average generalization error over all possible training sets of size N_{tj}, see [13].

The optimal regularization can be found by using gradient descent[3],

$$\boldsymbol{\kappa}_{(n+1)} = \boldsymbol{\kappa}_{(n)} - \eta \frac{\partial \widehat{\Gamma}_{\mathrm{cv}}}{\partial \boldsymbol{\kappa}}(\widehat{\boldsymbol{w}}(\boldsymbol{\kappa}_{(n)})) \tag{4}$$

where $\eta > 0$ is a step-size (learning rate) and $\boldsymbol{\kappa}_{(n)}$ is the estimate of the regularization parameters in iteration n. After convergence[4] it is recommended to retrain on all available data using the optimized regularization parameters.

In case of linear regularization, the gradient of the cross-validation error can be written as [12], [13]

$$\frac{\partial \widehat{\Gamma}_{\mathrm{cv}}}{\partial \boldsymbol{\kappa}}(\boldsymbol{\kappa}) = \frac{1}{K}\sum_{j=1}^{K} \frac{\partial S_{\mathcal{V}_j}}{\partial \boldsymbol{\kappa}}(\widehat{\boldsymbol{w}}_j),\ \frac{\partial S_{\mathcal{V}_j}}{\partial \boldsymbol{\kappa}}(\widehat{\boldsymbol{w}}_j) = -\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{w}^\top}(\widehat{\boldsymbol{w}}_j)\cdot \boldsymbol{J}_j^{-1}(\widehat{\boldsymbol{w}}_j)\cdot \frac{\partial S_{\mathcal{V}_j}}{\partial \boldsymbol{w}}(\widehat{\boldsymbol{w}}_j). \tag{5}$$

where $\boldsymbol{J}_j(\boldsymbol{w}) = \partial^2 C_{\mathcal{T}_j}(\boldsymbol{w})/\partial \boldsymbol{w}\partial \boldsymbol{w}^\top$ is the Hessian of the cost function. As an example, consider the case of weight decay regularization with separate weight decays for two group of weights, e.g., the input-to-hidden and hidden-to output weights of a neural network:

$$R(\boldsymbol{w}, \boldsymbol{\kappa}) = \kappa^I \cdot |\boldsymbol{w}^I|^2 + \kappa^H \cdot |\boldsymbol{w}^H|^2 \tag{6}$$

where $\boldsymbol{\kappa} = [\kappa^I, \kappa^H]$, $\boldsymbol{w} = [\boldsymbol{w}^I, \boldsymbol{w}^H]$ with $\boldsymbol{w}^I$, $\boldsymbol{w}^H$ denoting the input-to-hidden and hidden-to output weights, respectively. The gradient of the validation error then yields,

$$\frac{\partial S_{\mathcal{V}_j}}{\partial \kappa^I}(\widehat{\boldsymbol{w}}_j) = -2(\widehat{\boldsymbol{w}}_j^I)^\top \cdot \boldsymbol{g}_j^I,\ \ \frac{\partial S_{\mathcal{V}_j}}{\partial \kappa^H}(\widehat{\boldsymbol{w}}_j) = -2(\widehat{\boldsymbol{w}}_j^H)^\top \cdot \boldsymbol{g}_j^H \tag{7}$$

where $\boldsymbol{g}_j$ is the vector $\boldsymbol{g}_j = [\boldsymbol{g}_j^I, \boldsymbol{g}_j^H] = \boldsymbol{J}_j^{-1}(\widehat{\boldsymbol{w}}_j) \cdot \partial S_{\mathcal{V}_j}(\widehat{\boldsymbol{w}}_j)/\partial \boldsymbol{w}$.

Algebraic Generalization Error Approach

The literature suggests many algebraic estimators of the generalization error, including: FPER [10], GEN [8], GPE [14] and NIC [17]. The various estimators differ mainly in assumptions regarding model bias and dependence among data examples. In particular, they are all $o(1/N_{\mathcal{T}})$ estimators where $N_{\mathcal{T}}$ is the number of training examples. In many practical modeling scenarios the large training set assumption may be violated, however, the adaptive regularization based on this algebraic estimate might still be useful, as demonstrated in the experimental section. The major advantage of algebraic estimators is that all available data can be used to train the model,

[3] Optimization can be improved by using second order information [5], [3].

[4] E.g., small norm of gradient or small change in validation error.

i.e., $\mathcal{T} = \mathcal{D}$. This is not the case when using the empirical validation error approach discussed above.

In [7] properties of adaptive regularization is studied in the simple case of estimating the mean of a random variable using an algebraic estimate of the average generalization error, and [6] proposed an adaptive regularization scheme for neural networks based on an algebraic estimate. In the following we present an extended version of this scheme where regularization parameters are adapted by an iterative gradient descent scheme aiming at minimizing the GEN/NIC [8], [9], [17], estimate of the generalization error. We use GEN/NIC in this work as an representative for the family of algebraic estimators. GEN/NIC has the advantage that model biased is not assumed negligible. The presented procedure can in principle be invoked for any of the mentioned estimators.

The $o(1/N_{\mathcal{T}})$ GEN/NIC estimate of the average generalization error is

$$\Gamma_{\mathrm{GEN}} = E_{\mathcal{T}}\{S_{\mathcal{T}}(\widehat{\boldsymbol{w}})\} + \frac{m_{\mathrm{eff}}}{N_{\mathcal{T}}} - \frac{A}{N_{\mathcal{T}}} \cdot \frac{\partial R}{\partial \boldsymbol{w}^\top}(\boldsymbol{w}^*)\boldsymbol{J}^{-1}(\boldsymbol{w}^*)\frac{\partial R}{\partial \boldsymbol{w}}(\boldsymbol{w}^*) \tag{8}$$

where $\boldsymbol{w}^*$ are the optimal model weights, i.e., $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} G(\boldsymbol{w})$. m_{eff} is the effective number of parameters[5] (weights) in the model [8], [9]

$$\begin{aligned} m_{\mathrm{eff}} &= \mathrm{tr}\left[\boldsymbol{J}^{-1}(\boldsymbol{w}^*)\left(\boldsymbol{K}(0) + \sum_{n=1}^{\bar{M}} \frac{N_{\mathcal{T}} - n}{N_{\mathcal{T}}}(\boldsymbol{K}(n) + \boldsymbol{K}^\top(n))\right)\right] \\ &= \mathrm{tr}\left[\boldsymbol{J}^{-1}(\boldsymbol{w}^*)\boldsymbol{L}\right] \end{aligned} \tag{9}$$

where $\bar{M} = \min(M, N_{\mathcal{T}} - 1)$, M is the time dependence length (for i.i.d. examples $M = 0$), $A = \bar{M} + 1 - \bar{M}(\bar{M}+1)/2N_{\mathcal{T}}$, and $\boldsymbol{K}(n) = E\{\partial\ell(k)/\partial\boldsymbol{w} \cdot \partial\ell(k+n)/\partial\boldsymbol{w}^\top\}$ with $\ell(k) \equiv \ell(\boldsymbol{y}(k), \widehat{\boldsymbol{y}}(k;\boldsymbol{w}^*))$, as $E\{\cdot\}$ denotes expectation w.r.t. joint input-output distribution. $\boldsymbol{J}(\boldsymbol{w})$ is the Hessian matrix of the expected cost function $E_{\mathcal{T}}\{C_{\mathcal{T}}(\boldsymbol{w})\}$, i.e., $\boldsymbol{J}(\boldsymbol{w}) = \boldsymbol{H}(\boldsymbol{w}) + \partial^2 R/\partial\boldsymbol{w}\partial\boldsymbol{w}^\top$. If data are independent $\boldsymbol{K}(n) \equiv \boldsymbol{0}$ for $n > 0$, and if the cost is the log-likelihood loss then $\boldsymbol{K}(0)$ becomes the Hessian matrix of the unregularized cost, i.e., $\boldsymbol{K}(0) = \boldsymbol{H}(\boldsymbol{w}^*) = \partial^2 G(\boldsymbol{w}^*)/\partial\boldsymbol{w}\partial\boldsymbol{w}^\top$.

For practical implementation the quantities in Eq. (8) are estimated from data, as shown by,

$$\widehat{\Gamma}_{\mathrm{GEN}} = S_{\mathcal{T}}(\widehat{\boldsymbol{w}}) + \frac{\widehat{m}_{\mathrm{eff}}}{N_{\mathcal{T}}} - \frac{A}{N_{\mathcal{T}}} \cdot \frac{\partial R}{\partial \boldsymbol{w}^\top}(\widehat{\boldsymbol{w}})\boldsymbol{J}_{\mathcal{T}}^{-1}(\widehat{\boldsymbol{w}})\frac{\partial R}{\partial \boldsymbol{w}}(\widehat{\boldsymbol{w}}) \tag{10}$$

where $\widehat{m}_{\mathrm{eff}}$ is calculated via Eq. (9) by substituting $\boldsymbol{J}_{\mathcal{T}}^{-1}(\widehat{\boldsymbol{w}})$ for $\boldsymbol{J}^{-1}(\boldsymbol{w}^*)$,

$$\boldsymbol{J}_{\mathcal{T}}(\boldsymbol{w}) = \boldsymbol{H}_{\mathcal{T}}(\boldsymbol{w}) + \partial^2 R(\boldsymbol{w},\boldsymbol{\kappa})/\partial\boldsymbol{w}\partial\boldsymbol{w}^\top, \quad \boldsymbol{H}_{\mathcal{T}}(\boldsymbol{w}) = \frac{\partial^2 S_{\mathcal{T}}}{\partial\boldsymbol{w}\partial\boldsymbol{w}^\top}(\boldsymbol{w}). \tag{11}$$

Further,

$$\boldsymbol{K}_{\mathcal{T}}(n) = \frac{1}{N_{\mathcal{T}}} \sum_{k=1}^{N_{\mathcal{T}}-n} \frac{\partial\ell(\boldsymbol{y}(k), \widehat{\boldsymbol{y}}(k;\widehat{\boldsymbol{w}}))}{\partial\boldsymbol{w}} \cdot \frac{\partial\ell(\boldsymbol{y}(k+n), \widehat{\boldsymbol{y}}(k+n;\widehat{\boldsymbol{w}}))}{\partial\boldsymbol{w}^\top} \tag{12}$$

[5] For some cost functions, e.g., mean square error, m_{eff} is scaled by noise variance.

is substituted for $\boldsymbol{K}(n)$. To proceed as in [6], a simple gradient descent optimization as in Eq. (4) can be used[6]. The gradient of $\widehat{\Gamma}_{\mathrm{GEN}}$, noting that all quantities are evaluated at $\widehat{\boldsymbol{w}} = \widehat{\boldsymbol{w}}(\kappa)$, is according to Eq. (10) and (5) in case of linear regularization, given by

$$\frac{\partial \widehat{\Gamma}_{\mathrm{GEN}}}{\partial \boldsymbol{\kappa}} = \frac{\partial S_{\mathcal{T}}}{\partial \boldsymbol{\kappa}} + \frac{1}{N_{\mathcal{T}}} \frac{\partial \widehat{m}_{\mathrm{eff}}}{\partial \boldsymbol{\kappa}} - \frac{A}{N_{\mathcal{T}}} \cdot \frac{\partial}{\partial \boldsymbol{\kappa}} \left(\boldsymbol{\kappa}^{\top} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{w}^{\top}} \boldsymbol{J}_{\mathcal{T}}^{-1} \frac{\partial \boldsymbol{r}^{\top}}{\partial \boldsymbol{w}} \boldsymbol{\kappa} \right). \tag{13}$$

Eq. (13) can be written as

$$\begin{aligned} \frac{\partial \widehat{\Gamma}_{\mathrm{GEN}}}{\partial \kappa_i} &= \left(1 - \frac{2A}{N_{\mathcal{T}}}\right) \left\{ \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{w}^{\top}} \boldsymbol{J}_{\mathcal{T}}^{-1} \frac{\partial \boldsymbol{r}^{\top}}{\partial \boldsymbol{w}} \boldsymbol{\kappa} \right\}_i - \frac{1}{N_{\mathcal{T}}} \mathrm{tr} \left[\boldsymbol{L}_{\mathcal{T}} \boldsymbol{J}_{\mathcal{T}}^{-1} \frac{\partial^2 r_i}{\partial \boldsymbol{w} \partial \boldsymbol{w}^{\top}} \boldsymbol{J}_{\mathcal{T}}^{-1} \right] \\ &\quad + \frac{A}{N_{\mathcal{T}}} \boldsymbol{\kappa}^{\top} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{w}^{\top}} \boldsymbol{J}_{\mathcal{T}}^{-1} \frac{\partial^2 r_i}{\partial \boldsymbol{w} \partial \boldsymbol{w}^{\top}} \boldsymbol{J}_{\mathcal{T}}^{-1} \frac{\partial \boldsymbol{r}^{\top}}{\partial \boldsymbol{w}} \boldsymbol{\kappa}. \end{aligned} \tag{14}$$

Evidence Approximation Approach

The Bayesian evidence approach adapts regularization parameters so as to minimize the evidence [2, Ch. 10], [15], [16]. The evidence is the probability of data[7] given the model, $p(\mathcal{T}|\mathcal{M}) = \int p(\mathcal{T}|\boldsymbol{w}, \mathcal{M}) \cdot p(\boldsymbol{w}|\mathcal{M})\, d\boldsymbol{w}$, where $p(\mathcal{T}|\boldsymbol{w}, \mathcal{M})$ is the likelihood and $p(\boldsymbol{w}|\mathcal{M})$ is the prior. In terms of the cost function components in Eq. (1) the likelihood and prior are expressed by:

$$p(\mathcal{T}|\boldsymbol{w}, \mathcal{M}) = Z_S^{-1} \exp(-\beta N_{\mathcal{T}} S_{\mathcal{T}}(\boldsymbol{w})),\ p(\boldsymbol{w}|\mathcal{M}) = Z_R^{-1} \exp(-R(\boldsymbol{w}, \boldsymbol{\kappa})) \tag{15}$$

where β plays the role of the precision (inverse noise variance), and Z_S, Z_R are normalization constants. The evidence approximation framework consists in expanding the evidence to second order around the maximum aposteriori solution $\widehat{\boldsymbol{w}}$. According to [15] the negative log-evidence is

$$\begin{aligned} -\log p(\mathcal{T}|\mathcal{M}) &\approx \beta N_{\mathcal{T}} S_{\mathcal{T}}(\widehat{\boldsymbol{w}}) + R(\widehat{\boldsymbol{w}}, \boldsymbol{\kappa}) + \log Z_S + \log Z_R \\ &\quad + \frac{\log |\boldsymbol{J}_{\mathcal{T}}(\widehat{\boldsymbol{w}})|}{2} + \frac{m}{2} (\log \beta + \log N_{\mathcal{T}} - \log 2\pi). \end{aligned} \tag{16}$$

If the likelihood and the weight prior are assumed to be Gaussian distributed[8], which corresponds to using mean square loss and weight decay regularization as in Eq. (6), then the negative log-evidence is approximated by

$$\begin{aligned} -\log p(\mathcal{T}|\mathcal{M}) &\approx \beta N_{\mathcal{T}} S_{\mathcal{T}}(\widehat{\boldsymbol{w}}) + \alpha^I |\widehat{\boldsymbol{w}}^I|^2 + \alpha^H |\widehat{\boldsymbol{w}}^H|^2 - \frac{(N_{\mathcal{T}} - m) \log \beta}{2} \\ &\quad - \frac{m^I \log \alpha^I + m^H \log \alpha^H}{2} + \frac{\log |\boldsymbol{J}_{\mathcal{T}}(\widehat{\boldsymbol{w}})|}{2} + \frac{N_{\mathcal{T}} \log \pi + m(\log N_{\mathcal{T}} - \log 2)}{2} \end{aligned} \tag{17}$$

[6] [6] proceeds by finding the gradient of Eq. (8) and then use plug in estimates of unknown quantities. Here we proceed from the computable estimate Eq. (10). The difference between these approaches turn out to be minor.

[7] Also for this approach no validation data is required, i.e., $\mathcal{T} = \mathcal{D}$.

[8] If these assumptions are not fulfilled, the evidence framework becomes much more complicated and closed form solution can generally not be obtained. In such cases Monte Carlo techniques are required.

where $\alpha_i = \kappa_i/(\beta N_{\mathcal{T}})$ are the normalized weight decays, m^I, m^H are the number of hidden-to-input and hidden-to-output weights, respectively.

Minimizing the negative log-evidence by solving the equation, the derivative of Eq. (18) w.r.t. $\beta, \alpha^I, \alpha^H$ equal to zero[9] results in optimal updated choices for $\beta, \alpha^I, \alpha^H$ based on current values ($\widehat{\boldsymbol{w}} = \widehat{\boldsymbol{w}}(\beta_{(n)}, \boldsymbol{\alpha}_{(n)})$),

$$\beta_{(n+1)} = \frac{N - \widehat{m}_{\text{eff}}}{N_{\mathcal{T}} S_{\mathcal{T}}(\widehat{\boldsymbol{w}})}, \quad \alpha^I_{(n+1)} = \frac{\widehat{m}^I_{\text{eff}}}{2|\widehat{\boldsymbol{w}}^I|^2}, \quad \alpha^H_{(n+1)} = \frac{\widehat{m}^H_{\text{eff}}}{2|\widehat{\boldsymbol{w}}^H|^2} \tag{18}$$

where $\widehat{m}_{\text{eff}} = \widehat{m}^I_{\text{eff}} + \widehat{m}^H_{\text{eff}}$ and $\widehat{m}^I_{\text{eff}} = \sum_{i \in \mathcal{I}} \lambda_i/(\lambda_i + \kappa^I)$, $\widehat{m}^H_{\text{eff}} = \sum_{i \in \mathcal{H}} \lambda_i/(\lambda_i + \kappa^H)$ with $\mathcal{I}, \mathcal{H}$ defining the indices for input-to-hidden and hidden-to-output weights, respectively, and λ_i is the i'th eigenvalue of $\boldsymbol{H}_{\mathcal{T}}(\widehat{\boldsymbol{w}})$. Thus, the evidence based scheme consists in alternating between weight optimization and update of weight decay and precision parameters, like the generalization based schemes.

EXPERIMENTS

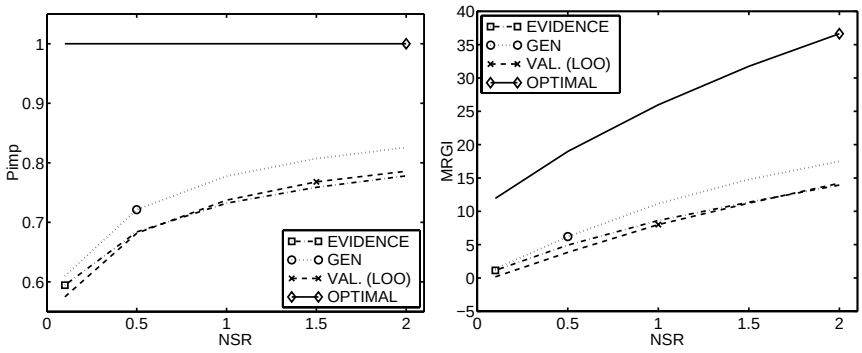

Figure 1: Results for Hessian with low eigenvalue spread. The GEN method is the most effective method for determining the regularization parameter. At a low NSR the effectiveness of all the methods is similar. The evidence method and leave-one-out (LOO) validation based method have similar performance. Optimal κ is found by exhaustive search.

Consider modeling a simple linear system $y(n) = \boldsymbol{x}^\top(n)\boldsymbol{w}^\circ + \epsilon(n)$. The input $\boldsymbol{x}(n) = [x_1(n), \cdots, x_m(n)]^\top$ is a $m = 10$ dimensional i.i.d. Gaussian distributed vector $\boldsymbol{x}(n) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{H})$, where $\boldsymbol{H}$ is the covariance matrix. The true weight vector is $\boldsymbol{w}^* = [1, 1, 1, 0, 0, 0, 0, 0, 0, 0]^\top$. The noise $\epsilon(n) \sim \mathcal{N}(0, \sigma^2_\epsilon)$ is i.i.d. and independent of $\boldsymbol{x}(n)$. The noise variance is determined by $\sigma^2_\epsilon = \text{NSR} \cdot (\boldsymbol{w}^*)^\top \boldsymbol{H} \boldsymbol{w}^\circ$, where NSR is the noise-to-signal ratio of the output. The weights are estimated using mean square error augmented by a simple weight decay, i.e., $\widehat{\boldsymbol{w}} = \boldsymbol{J}^{-1}_{\mathcal{T}} \boldsymbol{X}^\top \boldsymbol{y}/N_{\mathcal{T}}$, where $\boldsymbol{y} = [y(1), \cdots, y(N_{\mathcal{T}})]^\top$ and

[9] Some negligible terms are omitted, see [16].

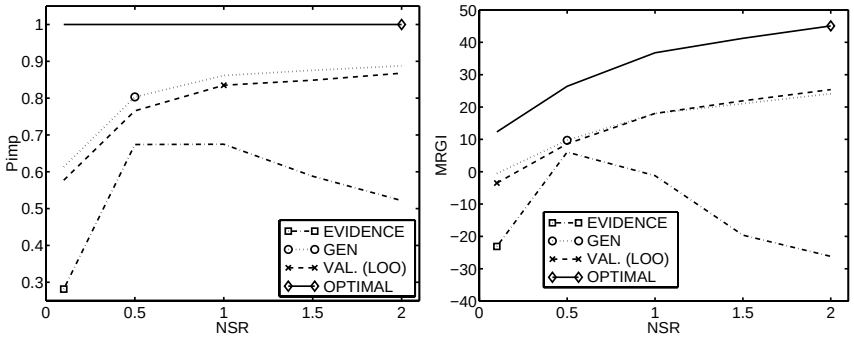

Figure 2: Results for Hessian with high eigenvalue spread. The GEN method has still the highest $P_{\rm imp}$, but the MRGI is similar to the LOO validation based method. The evidence method has clearly the worst performance. This is caused by extremely low $\widehat{m}_{\rm eff}$, which seems to influence more the evidence method then the GEN. Notice that all methods have negative MRGI at NSR=0.1.

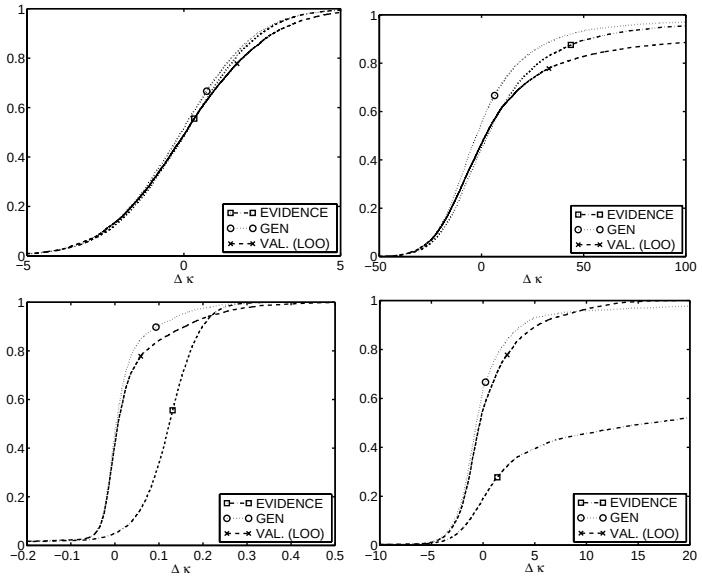

Figure 3: Cumulative probability distribution for the difference between the optimal κ and the κ's suggested by the different regularization schemes. Left panels NSR = 0.1 and right panels NSR = 2.0. Top and bottom rows are low and high eigenvalue spread for the Hessian matrix, respectively. Top row: When the NSR is low the distributions are similar, while at high NSR the evidence and LOO validation method have a larger tail then the GEN, indicating that the they are estimating κ too large. Bottom row: When the NSR is low the evidence method suggests κ's that are too large, even though it does not have a large tail. When the NSR is large the evidence method has a very large tail, again estimating κ too large. The LOO validation method and the GEN show similar distributions.

$\boldsymbol{J}_{\mathcal{T}}$ is the Hessian of the regularized cost function given by $\boldsymbol{J}_{\mathcal{T}} = \boldsymbol{H}_{\mathcal{T}} + \kappa \boldsymbol{I}$ where $\boldsymbol{H}_{\mathcal{T}} = \boldsymbol{X}\boldsymbol{X}^\top / N_{\mathcal{T}}$ and $\boldsymbol{X} = [\boldsymbol{x}(1), \cdots, \boldsymbol{x}(N_{\mathcal{T}})]^\top$. The true generalization error of the estimated linear system is easily computed as $G(\widehat{\boldsymbol{w}}) = \sigma_\epsilon^2 + (\hat{\boldsymbol{w}} - \boldsymbol{w}^\circ)^\top \boldsymbol{H} (\widehat{\boldsymbol{w}} - \boldsymbol{w}^\circ)$.

In order to evaluate the performance of the regularization methods, $Q = 5000$ independent data sets of size $N = 40$ are generated. Two measures are then used to evaluate the performance. The *probability of improvement* measures the fraction of the Q estimated models, using some regularization scheme, which generalize better than using no regularization, and is defined by $P_{\text{imp}} = Q^{-1} \sum_{i=1}^{Q} \mu(G(\widehat{\boldsymbol{w}}_{\text{unreg}}^{(i)}) - G(\widehat{\boldsymbol{w}}^{(i)}))$, where $\mu(x) = 1$ for $x > 0$, and zero otherwise. $G(\widehat{\boldsymbol{w}}_{\text{unreg}}^{(i)})$ is the generalization error of the model trained on the ith data set using no regularization. The second performance measure is the *mean relative generalization error improvement* defined as $\text{MRGI} = Q^{-1} \sum_{i=1}^{Q} 100\% \left[G(\widehat{\boldsymbol{w}}_{\text{unreg}}^{(i)}) - G(\widehat{\boldsymbol{w}}^{(i)}) \right] / G(\widehat{\boldsymbol{w}}_{\text{unreg}}^{(i)})$. Two different conditions for the Hessian are considered: small and large eigenvalue spread. Small eigenvalue spread is around 10, while more common large eigenvalue spread around 10^4 is obtained by multiplying a Vandermonde matrix $\boldsymbol{A}$ to the original input $\boldsymbol{X}$ using $\widetilde{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{X}$ as the input. The methods are also compared at different NSR's. The performance is demonstrated through Fig. 1–3.

The computational complexity of the adaptive regularization schemes is very different. The leave-one-out (LOO) validation based method has the most computational overhead. Reestimating the weights $N_{\mathcal{T}}$ times is obviously very time consuming. Both the LOO validation based and GEN methods use gradient descent for estimating the regularization parameters. The convergence is very dependent on the step-size η Eq. (4). In particular, when the eigenvalue spread is high a small value has to be used, thus slowing down the convergence. The evidence method is much faster as Eq. (18) are analytical equations for regularization parameter updates.

CONCLUSION

This paper compared generalization error and evidence based schemes for adaptive regularization. We suggested various algorithm extensions and performed numerical experiments with linear models. The generalization error based methods generally performs good, while the evidence method yields comparable performance at low Hessian eigenvalue spread. However, at high eigenvalue spread, which is the common case in neural net applications, the evidence method has very low generalization error improvement.

Acknowledgments. Research supported by the Danish Research Councils through the THOR Center for Neuroinformatics and the Signal and Image Processing for Telemedicine (SITE) program.

REFERENCES

[1] L.N. Andersen, J. Larsen, L.K. Hansen & M. Hintz-Madsen: "Adaptive Regularization of Neural Classifiers," in J. Principe *et al.* (eds.) **Proceedings of IEEE NNSP VII**, pp. 24–33, 1997.

[2] C.M. Bishop: **Neural Networks for Pattern Recognition**, Oxford University Press, 1995.

[3] D. Chen & M. Hagan: "Optimal Use of Regularization and Cross-Validation in Neural Network Modeling," in **Proceedings of IJCNN**, Washington DC, vol. 2, pp. 1275–1280, 1999.

[4] S. Geman, E. Bienenstock & R. Doursat: "Neural Networks and the Bias/Variance Dilemma," **Neural Computation**, vol. 4, pp. 1–58, 1992.

[5] C. Goutte & J. Larsen: "Adaptive Regularization of Neural Networks using Conjugate Gradient," in **Proceedings of ICASSP'98**, Seattle, USA, vol. 2, pp. 1201–1204, 1998.

[6] L.K. Hansen, C.E. Rasmussen, C. Svarer & J. Larsen: "Adaptive Regularization," in J. Vlontzos *et al.* (eds.) **Proceedings of IEEE NNSP IV**, pp. 78–87, 1994.

[7] L.K. Hansen, and C.E. Rasmussen: "Pruning from Adaptive Regularization," **Neural Computation**, vol. 6, pp. 1223–1232, 1994.

[8] J. Larsen: A Generalization Error Estimate for Nonlinear Systems. In S.Y. Kung *et al.* (eds.), **Proceedings of IEEE NNSP II**, pp. 29–38, 1992.

[9] J. Larsen, **Design of Neural Network Filters**, Ph.D. Thesis, Electronics Institute, Techn. Univ. of Denmark, March 1993.

[10] J. Larsen & L.K. Hansen: Generalization Performance of Regularized Neural Network Models. In J. Vlontzos *et al.* (eds.), **Proceedings of IEEE NNSP IV**, pp. 42–51, 1994.

[11] J. Larsen & L.K. Hansen: "Empirical Generalization Assessment of Neural Network Models," in F. Girosi *et al.* (eds.) **Proceedings of NNSP V**, Piscataway, New Jersey: IEEE, pp. 30–39, 1995.

[12] J. Larsen, L.K. Hansen, C. Svarer & M. Ohlsson: "Design and Regularization of Neural Networks: The Optimal Use of a Validation Set," in S. Usui *et al.* (eds.), **Proceedings of IEEE NNSP VI**, pp. 62–71, 1996.

[13] J. Larsen, C. Svarer, L. Nonboe Andersen & L.K. Hansen: "Adaptive Regularization in Neural Network Modeling," in G.B. Orr, K. Müller (eds.) **Neural Networks: Tricks of the Trade**, Lecture Notes in Computer Science 1524, Germany: Springer-Verlag, Chapter 5, pp. 113-132, 1998.

[14] J. Moody: "The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems," in J.E. Moody *et al.* (eds.) **Proceedings of NIPS 4**, Morgan Kaufmann Pub., pp. 847–854, 1992.

[15] D.J.C. MacKay: "Bayesian Model Comaprison and Backprop Nets ," in J.E. Moody *et al.* (eds.) **Proceedings of NIPS 4**, Morgan Kaufmann Pub., pp. 839–846, 1992.

[16] D.J.C. MacKay: "A Practical Bayesian Framework for Backprop Networks," **Neural Computation**, vol. 4, no. 3, pp. 448–472, 1992.

[17] N. Murata, S. Yoshizawa & S. Amari: "Network Information Criterion — Determining the Number of Hidden Units for an Artificial Neural Network Model," **IEEE Trans. on NN**, vol. 5, no. 6, pp. 865–872, 1994.

[18] M. Stone: "Cross-validatory Choice and Assessment of Statistical Predictors," **Jour. Royal Stat. Soc. B** vol. 36, no. 2, pp. 111–147, 1974.