2001 Workshop on Information-Based Induction Sciences (IBIS2001)
Tokyo, Japan, July 30 - August 1, 2001.

Subspace Information Criterion for Sparse Regressors

Koji Tsuda * Masashi Sugiyama † Klaus-Robert Müller ‡

Abstract: Non-quadratic regularizers, in particular the ℓ_1 norm regularizer can yield sparse solutions that generalize well. In this work we propose the Generalized Subspace Information Criterion (GSIC) that allows to predict the generalization error for this useful family of regularizers. We show that under some technical assumptions GSIC is an asymptotically unbiased estimator of the generalization error. GSIC is demonstrated to have a good performance in experiments with the ℓ_1 norm regularizer as we compare with the Network Information Criterion and cross-validation in relatively large sample cases. However in the small sample cases, GSIC tends to fail to capture the optimal model due to its large variance. Therefore, also a biased version of GSIC is introduced, which achieves reliable model selection in the relevant and challenging scenario of high dimensional data and few samples.

1 Introduction

To avoid overfitting in supervised learning, the parameters $\boldsymbol{\theta}$ of the learning machine are often determined such that a weighted sum of the training error and the regularization term R

$$\mathrm{Error}(\boldsymbol{\theta}) = \mathrm{TrainingError}(\boldsymbol{\theta}) + \lambda R(\boldsymbol{\theta}) \qquad (1)$$

is minimized, where λ is called the *regularization constant*. In this paper we will consider the problem of determining the regularization constant. The ideal criterion would be the generalization error itself, or approximations thereof, e.g. in a worst or average case setting. The former considers the worst generalization error achieved on all possible training sets (e.g. methods based on VC theory [1]). The latter considers ensemble averages over all possible training sets, for example the Network Information Criterion (NIC) [2] or the Subspace Information Criterion (SIC) [3, 4]. Furthermore there are very successful criteria as cross validation[1], C_L [5] or the Bayesian evidence framework [6], which approximately evaluate the ensemble error using the training data.

Among the prediction methods for the ensemble average of the generalization error, SIC has been shown to perform better than other methods, particularly in the small sample setting [3]. The technical feature of SIC is that it predicts the generalization error by utilizing a *reference estimator*, which is an unbiased estimate of the true parameter. SIC was so far only applicable to linear regression with *quadratic* regularizers, which includes e.g. *weight decay*.

Recently sparsity inducing non-quadratic regularizers have become rather popular (e.g. [7]) since with still good generalization properties sparse solutions (i.e. most of the model parameters become zero) are found in the training process. Often they are based on l_1 regularization. Since such regularization terms are non-quadratic, the original SIC criterion cannot be applied to them.

In this work we therefore propose the Generalized Subspace Information Criterion (GSIC) that allows to predict the generalization error for the family of non-quadratic regularizers. Among several other interesting theoretical properties, we will show that GSIC is an asymptotically *unbiased* estimator of the generalization error under several assumptions. In experiments with relatively large samples, GSIC achieves a good performance as we compare with NIC and cross-validation. However, in small sample cases, GSIC tends to fail to capture the optimal model due to its large variance. To alleviate this problem, we introduce a biased version of GSIC, which is derived from a reference estimator regularized by a quadratic regularizer. This biased version (GSICb) introduces yet another model selection problem: determining the regularization constant of the reference estimator. But, since a quadratic regularizer is used here, the regularization constant can be determined by efficient algorithms [8]. As a result, overall computation time is much shorter for GSICb in comparison with cross validation. In experiments using an ℓ_1 norm regularizer, GSICb shows an excellent performance, when compared to NIC and cross-validation.

2 Preliminaries

In a linear regression problem, a target function is approximated by a parametric model which is linear in

*AIST Computational Biology Research Center, 2-41-6, Aomi, Koto-ku, Tokyo, 135-0064, Japan
†Tokyo Institute of Technology, 2-12-1, O-okayama, Meguro-ku, Tokyo, 152-8552, Japan
‡GMD FIRST, Kekuléstr. 7, 12489 Berlin, Germany and University of Potsdam, Am Neuen Palais 10, 14469 Potsdam, Germany

parameters. Let us assume that the target function $f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^d$, is contained in a parametric model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{i=1}^{p} \theta_i \phi_i(\boldsymbol{x})$, where $\phi_i : \mathbb{R}^d \to \mathbb{R}$ is a given (nonlinear) basis function and $\boldsymbol{\theta} \in \mathbb{R}^p$ is the parameter vector. Then, we can describe $f(\boldsymbol{x})$ as $f(\boldsymbol{x}) = \sum_{i=1}^{p} \theta_i^* \phi_i(\boldsymbol{x})$, where θ_i^* is the true parameter. The training examples consist of input points $\boldsymbol{x}_i \in \mathbb{R}^d$ and the corresponding output $y_i \in \mathbb{R}$, which are degraded by additive noise ϵ_i: $y_i = f(\boldsymbol{x}_i) + \epsilon_i$. We assume that all random variables $\{\epsilon_i\}_{i=1}^n$ are independent and subject to the same symmetric distribution with mean zero and variance σ^2. In this paper, we focus on the case where the parameter $\boldsymbol{\theta}$ is determined by finding $\boldsymbol{\theta}$ that minimizes a weighted sum of squared errors and a (twice differentiable) *regularization term* $R(\boldsymbol{\theta})$

$$L_r = \frac{1}{n} \sum_{i=1}^{n} (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2 + \lambda R(\boldsymbol{\theta}), \qquad (2)$$

where λ is called the *regularization constant*. Let us define $\hat{\boldsymbol{\theta}}$ as the solution of the optimization problem: $\hat{\boldsymbol{\theta}} = \mathrm{argmin}_{\boldsymbol{\theta}} L_r(\boldsymbol{\theta})$. The generalization error of $\hat{\boldsymbol{\theta}}$ is

$$E_{\boldsymbol{x}}[(f(\boldsymbol{x}) - f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))^2] = \int (f(\boldsymbol{x}) - f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))^2 q(\boldsymbol{x}) d\boldsymbol{x}, \quad (3)$$

where $q(\boldsymbol{x})$ denotes the distribution of input $\boldsymbol{x}$. Let us assume that the solution of (2) is unique. Then, the solution $\hat{\boldsymbol{\theta}}$ is considered as an implicit function of training examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$: $\hat{\boldsymbol{\theta}}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, y_1, \cdots, y_n)$. In model selection, the optimal λ should be determined so that the generalization error is minimized. However, since $\hat{\boldsymbol{\theta}}$ depends on random variables y_i, the generalization error (3) is also a random variable. In order to compare two random variables, we focus on the mean only. The mean generalization error is called *ensemble average*, which is described as

$$J_G = E_{\boldsymbol{\epsilon}} E_{\boldsymbol{x}}[(f(\boldsymbol{x}) - f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))^2], \qquad (4)$$

where $\hat{\boldsymbol{\theta}}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, y_1, \cdots, y_n)$ is abbreviated as $\hat{\boldsymbol{\theta}}$ and $E_{\boldsymbol{\epsilon}} := E_{\epsilon_1} \cdots E_{\epsilon_n}$.

For the sake of a better geometrical understanding, we define the inner product in parameter space as $\langle \boldsymbol{\theta}, \boldsymbol{\theta}' \rangle = \boldsymbol{\theta}^T P \boldsymbol{\theta}'$, where P is the matrix whose (i,j) element is given as $P_{ij} = E_{\boldsymbol{x}}[\phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x})]$. Then we can rewrite the ensemble average of the generalization error using the norm of parameter space as

$$J_G = E_{\boldsymbol{\epsilon}} \|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|^2. \qquad (5)$$

The matrix P can be exactly calculated if we know the input distribution $q(\boldsymbol{x})$. If $q(\boldsymbol{x})$ is unknown, P can be estimated, e.g. by using the unlabeled samples $\{\boldsymbol{x}'_k\}_{k=1}^m$ or one can assume that $q(\boldsymbol{x})$ is the uniform distribution over some domain.

3 Generalization Error Prediction

Basic Idea Fig. 1 illustrates the idea. With respect to different training sets, the parameter $\hat{\boldsymbol{\theta}}$ takes various values and forms a distribution, where $\boldsymbol{\theta}^m$ is the mean of $\hat{\boldsymbol{\theta}}$, i.e., $\boldsymbol{\theta}^m = E_{\boldsymbol{\epsilon}}[\hat{\boldsymbol{\theta}}]$. The generalization error J_G is the average distance between $\hat{\boldsymbol{\theta}}$ and the underlying true solution $\boldsymbol{\theta}^*$. Because there is no information about $\boldsymbol{\theta}^*$, we introduce another parameter $\hat{\boldsymbol{\theta}^u}$ such that $\hat{\boldsymbol{\theta}^u}$ is an unbiased estimate of $\boldsymbol{\theta}^*$: $E_{\boldsymbol{\epsilon}}[\hat{\boldsymbol{\theta}^u}] = \boldsymbol{\theta}^*$. A typical choice of $\hat{\boldsymbol{\theta}^u}$ is the least mean squares estimator (i.e. without the regularizer) [3]. Then the distance between $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\theta}^u}$ (the broken line in Fig. 1) gives a rough estimate of the generalization error. We will derive an unbiased estimator of the generalization error by adding modification terms to this distance. Note that this technique to use an unbiased estimator was first introduced in SIC [3].

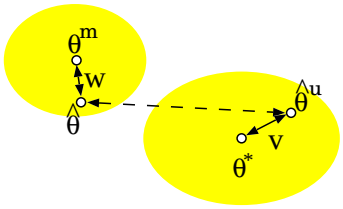

Figure 1: Basic idea for evaluating the generalization error.

Generalized SIC We will derive an unbiased estimator of J_G. J_G can be decomposed into the *bias* and *variance* as

$$E_{\boldsymbol{\epsilon}} \|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|^2 = \|\boldsymbol{\theta}^m - \boldsymbol{\theta}^*\|^2 + E_{\boldsymbol{\epsilon}} \langle \boldsymbol{w}, \boldsymbol{w} \rangle, \qquad (6)$$

where $\boldsymbol{w} := \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^m$. The bias term can be expressed by using $\|\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}\|^2$ as

$$\|\boldsymbol{\theta}^m - \boldsymbol{\theta}^*\|^2 = \|\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}\|^2 - \|\boldsymbol{w} - \boldsymbol{v}\|^2 - 2\langle \boldsymbol{w} - \boldsymbol{v}, \boldsymbol{\theta}^m - \boldsymbol{\theta}^* \rangle, \qquad (7)$$

where $\boldsymbol{v} := \hat{\boldsymbol{\theta}^u} - \boldsymbol{\theta}^*$. The second and third terms in (7) can not be directly evaluated, so we average out these terms. Then the second term yields

$$-E_{\boldsymbol{\epsilon}} \|\boldsymbol{w} - \boldsymbol{v}\|^2 = -E_{\boldsymbol{\epsilon}} \langle \boldsymbol{w}, \boldsymbol{w} \rangle + 2E_{\boldsymbol{\epsilon}} \langle \boldsymbol{w}, \boldsymbol{v} \rangle - E_{\boldsymbol{\epsilon}} \langle \boldsymbol{v}, \boldsymbol{v} \rangle,$$

and the third term vanishes. This approximation gives the following unbiased estimator of J_G called the *Generalized Subspace Information Criterion*:

$$\mathrm{GSIC} = \|\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}\|^2 + 2E_{\boldsymbol{\epsilon}} \langle \boldsymbol{w}, \boldsymbol{v} \rangle - E_{\boldsymbol{\epsilon}} \langle \boldsymbol{v}, \boldsymbol{v} \rangle. \qquad (8)$$

GSIC for Quadratic Regularizers We will derive GSIC for linear regression with a quadratic regularizer $R(\theta) = \hat{\boldsymbol{\theta}}^T R \hat{\boldsymbol{\theta}}$. Let K be the $n \times p$ matrix whose (i,j) element is $\phi_j(x_i)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)^T$. When $(\frac{1}{n} K^T K + \lambda R)$ is invertible, $\hat{\boldsymbol{\theta}}$ is given as

$$\hat{\boldsymbol{\theta}} = \frac{1}{n} (\frac{1}{n} K^T K + \lambda R)^{-1} K^T y. \qquad (9)$$

When K^TK is invertible, an unbiased estimate $\hat{\boldsymbol{\theta}^u}$ is given as [3]

$$\hat{\boldsymbol{\theta}^u} = (K^TK)^{-1}K^Ty. \quad (10)$$

This finally yields the original SIC for quadratic regularizers [3]:

$$\mathrm{SIC} = (\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u})^T P(\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}) + 2\sigma^2\mathrm{tr}(PW) - \sigma^2\mathrm{tr}(PV), \quad (11)$$

where W and V are the $p \times p$ matrices defined as

$$W = \frac{1}{n}(\frac{1}{n}K^TK + \lambda R)^{-1}, \quad V = (K^TK)^{-1}. \quad (12)$$

SIC gives an unbiased estimate of J_G [3]. Usually, the variance σ^2 is not known, so we use its unbiased estimate instead. When $n > p$, its unbiased estimate is given as $\hat{\sigma}^2 = \{\boldsymbol{y}^T\boldsymbol{y} - (K\hat{\boldsymbol{\theta}^u})^T\boldsymbol{y}\}/(n-p)$ [9]. Even when we replace σ^2 by $\hat{\sigma}^2$, the unbiasedness property is conserved[3].

GSIC for Non-Quadratic Regularizers When we are concerned with non-quadratic regularizers, $\hat{\boldsymbol{\theta}}$ can not be obtained analytically like in (9). For this reason, it is difficult to evaluate the second term $E_{\boldsymbol{\epsilon}}\langle \boldsymbol{w}, \boldsymbol{v}\rangle$ in (8). So we approximate $E_{\boldsymbol{\epsilon}}\langle \boldsymbol{w}, \boldsymbol{v}\rangle$ under the assumption that the Hessian $H = [\frac{\partial^2 L_r}{\partial\theta_i\partial\theta_j}]$ of the loss function L_r is invertible for any $\hat{\boldsymbol{\theta}}$. Then, $E_{\boldsymbol{\epsilon}}\langle \boldsymbol{w}, \boldsymbol{v}\rangle$ is approximated as

$$E_{\boldsymbol{\epsilon}}\langle \boldsymbol{w}, \boldsymbol{v}\rangle \approx \sigma^2\mathrm{tr}(PW^0), \quad (13)$$

where

$$W^0 = \frac{1}{n}\left(\frac{1}{n}K^TK + \frac{1}{2}\lambda\nabla\nabla R(\hat{\boldsymbol{\theta}})\right)^{-1}, \quad (14)$$

and $\nabla\nabla R(\hat{\boldsymbol{\theta}})$ is the $p \times p$ matrix whose (i,j) element is $\frac{\partial R(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j}\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$. The derivation of this approximation is described in appendix A. It gives GSIC for non-quadratic regularizers, which we propose in this paper:

$$\mathrm{GSIC} = (\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u})^T P(\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}) + 2\sigma^2\mathrm{tr}(PW^0) - \sigma^2\mathrm{tr}(PV). \quad (15)$$

When the regularization term is quadratic, GSIC agrees with the original SIC (11). When σ^2 is not known, it is replaced by $\hat{\sigma}^2$ as in SIC. With regard to the relationship between GSIC and the generalization error J_G, we have the following theorem.

Theorem 1 *Assuming that $\hat{\boldsymbol{\theta}}$ can be represented as a b-th ($b < \infty$) order polynomial of $\boldsymbol{y}$ and the moments of ϵ_i up to b-th order are bounded, then GSIC for non-quadratic regularizers is an asymptotic unbiased estimate of J_G: $E_{\boldsymbol{\epsilon}}[\mathrm{GSIC}] = J_G + O(n^{-2})$.*

A proof of the above theorem is provided in [10]. In order to discuss theoretical properties of GSIC with mathematical rigor, it would be necessary to obtain a version of the theorem without the assumption that $\hat{\boldsymbol{\theta}}$ is a finite order polynomial of $\boldsymbol{y}$. This, however, would go beyond the scope of this paper, which considers mainly the practical performance of GSIC.

4 Biased GSIC

In practical situations, it is common that as many basis functions as training examples are used, e.g. the Gaussian functions centered on all input points. In such cases, the unbiased solution $\hat{\boldsymbol{\theta}^u}$ tends to have a large variance, which also makes the variance of GSIC large. Therefore model selection can become unstable.

For reducing the variance, it is effective to replace $\hat{\boldsymbol{\theta}^u}$ by $\hat{\boldsymbol{\theta}^\alpha}$ obtained by weight decay regularization as $\hat{\boldsymbol{\theta}^\alpha} = (K^TK + \alpha I)^{-1}K^T\boldsymbol{y}$, where I is the $p \times p$ identity matrix. Although the mean of $\hat{\boldsymbol{\theta}^\alpha}$ has a small bias away from the true parameter $\boldsymbol{\theta}^*$, the variance of $\hat{\boldsymbol{\theta}^\alpha}$ becomes much smaller than that of $\hat{\boldsymbol{\theta}^u}$. We observe that by using the regularized $\hat{\boldsymbol{\theta}^\alpha}$ instead of $\hat{\boldsymbol{\theta}^u}$, GSIC becomes slightly biased but its variance is drastically reduced. We call this technique *biased GSIC*(GSICb). However, now another regularization constant α has to be determined. By adjusting α such that $\hat{\boldsymbol{\theta}^\alpha}$ is an accurate estimator of $\boldsymbol{\theta}^*$, the error of GSIC is expected to be improved. Fortunately, it is by far easier to determine α for weight decay regularization than to determine λ in the sparse regressor since in the weight decay case, the leave-one-out error can be efficiently computed in closed-form and other sophisticated methods are available[8]. Note that, by the regularized reference estimator, a good estimate of the noise variance σ^2 is obtained as $\hat{\sigma}^2 = \boldsymbol{y}^TZ^2\boldsymbol{y}/\mathrm{tr}(Z)$, where $Z = I - K(K^TK + \alpha I)^{-1}K^T$ (see e.g.[11]).

5 GSIC for Sparse Regression

It is well-known that the ℓ_1 norm regularization leads to a sparse solution, where most of the parameters θ_i's are zero [12, 13]. A sparse regressor is practically useful because it automatically selects necessary basis functions and moreover a sparse solution saves the computational cost. The loss function for the sparse regressor is given as

$$L_r = \frac{1}{n}\sum_{i=1}^{n}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2 + \lambda\sum_{i=1}^{p}|\theta_i|. \quad (16)$$

Minimizing L_r with respect to $\boldsymbol{\theta}$ is done by a convex quadratic programming [14].

As for sparse regressors $\nabla\nabla R$ is not well defined, GSIC cannot applied directly. In order to apply GSIC to the sparse regressor, we need to approximate the regularization term $R(\boldsymbol{\theta}) = \sum_{i=1}^{p}|\theta_i|$ by a continuous function as

$$R'(\boldsymbol{\theta}) = \sum_{i=1}^{p}\theta_i\tanh(\gamma\theta_i), \quad (17)$$

where the slope is e.g. $\gamma = 10$. Then, $\nabla\nabla R$ is a diagonal matrix whose i-th element is

$$\nabla\nabla R_{ii} = 2(\gamma\mathrm{sech}^2(\gamma\hat{\theta}_i) - \gamma^2\hat{\theta}_i\mathrm{sech}^2(\gamma\hat{\theta}_i)\tanh(\gamma\hat{\theta}_i)).$$

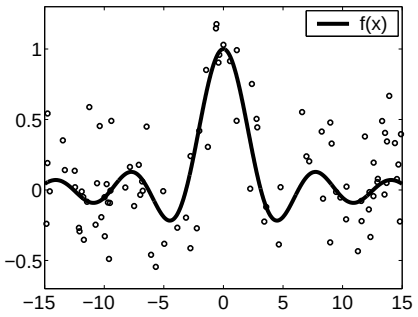

Figure 2: Learning target function and 100 training examples with $\sigma = 0.3$.

Because of this approximation, we are investigating the generalization error of the approximated regressor, not that of the sparse regressor itself. It is a further interesting topic to consider how to choose the approximator for minimizing the difference of the generalization errors, but outside the scope of this contribution.

6 Experiments

An Illustrative Example Let the regression function be $f_{\boldsymbol{\theta}}(x) = \sum_{i=1}^{50} \theta_i \exp\left(-\frac{\|x-s_i\|^2}{\eta^2}\right)$, where $\eta = 1$ and 50 template samples s_i's are equally spaced in $[-15, 15]$. We obtain the true parameter $\boldsymbol{\theta}^*$ by the least mean squares estimate with $\{(s_i, g(s_i))\}_{i=1}^{50}$, where $g(x) = |x|^{-1}\sin|x|$. For training, n input points $\{x_i\}_{i=1}^n$ are chosen randomly from the uniform distribution on $[-15, 15]$. The output values are obtained as $y_i = f(x_i) + \epsilon_i$, where ϵ_i's are independently subject to a normal distribution with mean zero and standard deviation σ. The target function and training examples are displayed in Fig. 2. The regularization constant is selected from $\lambda = 1.0 \times 10^{-4}, 1.0 \times 10^{-3.5}, \ldots, 1.0 \times 10^{-1}$ by 10-fold cross validation (CV), NIC, GSIC and GSICb. Also, 100 additional unlabeled samples $\{x'_i\}_{i=1}^{100}$ are given from the uniform distribution on $[-15, 15]$. In GSIC and GSICb, the distribution $q(x)$ of these additional input points is estimated by the empirical distribution of the unlabeled samples: $q(x) = \frac{1}{100}\sum_{i=1}^{100}\delta(x - x'_i)$, where $\delta(x) = 1$ when $x = 0$ and otherwise $\delta(x) = 0$. The true generalization error is measured by Error $= \int_{-15}^{15}(f_{\boldsymbol{\theta}}(x) - f(x))^2\,dx$.

The performance of CV, NIC, and GSIC is measured by the generalization error at selected λ (Fig. 3). The experiment consists of 100 trials with different noise. When $n = 200$, all criteria work well with no significant difference. As n decreases to 60, CV still works well, but NIC and GSIC tend to give a large generalization error.

In order to investigate the cause of errors by NIC and GSIC in detail, actual values of CV, NIC, and GSIC are displayed in Fig. 4 for $(n, \sigma) = (60, 0.3)$ and $(200, 0.3)$. Note that the values of GSIC in the figure are biased, because we ignored the terms $\|\hat{\boldsymbol{\theta}}^u\|^2$ and $\hat{\sigma}^2\mathrm{tr}(PV)$, which are irrelevant to model selection. Thus we can see the essential contributions to the variance of the estimate.

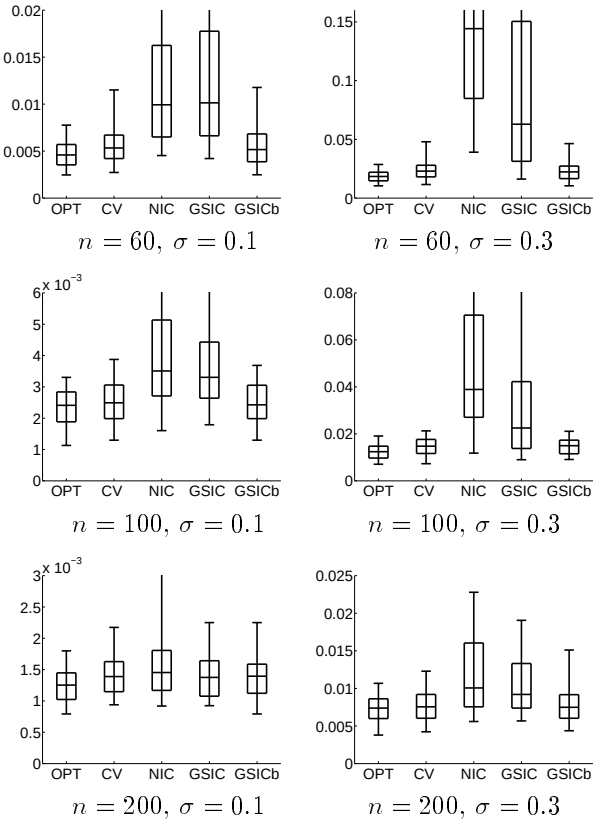

Figure 3: Generalization errors at selected λ for the respective model selection criterion shown with standard box plot (100 trials). The box plot notation specifies marks at 95, 75, 50, 25, and 5 percentiles of values. 'OPT' denotes the generalization error with the optimal λ.

When $(n, \sigma) = (200, 0.3)$, the shape of the curves by CV, NIC, and GSIC is very close to the true curve, which explains why the model selection was carried out successfully. Although CV still gives an accurate curve when $(n, \sigma) = (60, 0.3)$, the curves of NIC and GSIC are no longer accurate. These graphs also show that the inaccuracy of the curves by NIC and GSIC has different characteristics. The NIC curve is tilted towards the left, which shows that NIC tends to choose smaller regularization constants. This figure tells us that the unbiasedness of NIC is essentially lost because of the small sample effect. In GSIC, huge variance dominates the graph, so the shape of the average curve is unreliable. The variance of GSIC is large especially when the regularization constant λ is small. So, for explaining the failure in NIC, the bias plays a main role whereas in GSIC, the variance is of primal importance.

In order to reduce the variance of GSIC, we use the biased version GSICb. In the experiments, we use a leave-one-out cross-validation to approximate the

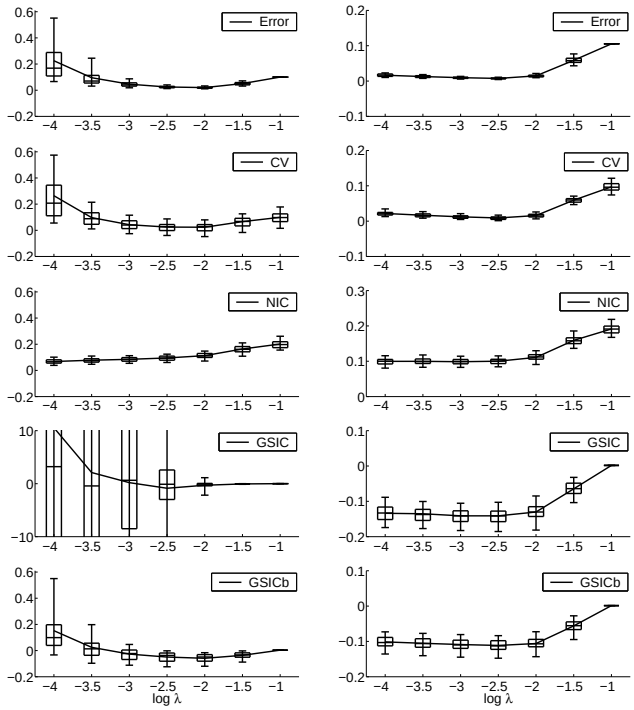

Figure 4: Values of each criterion by 100 trials shown with standard box plot. The horizontal axis denotes $\log \lambda$. The solid line denotes the mean values.

true generalization error and thus to determine α (see Sec. 4) from candidate values $\alpha = 1.0 \times 10^{-4}, 1.0 \times 10^{-3.5}, \ldots, 1.0 \times 10^{1}$. Fig. 3 shows that GSICb works as well as other methods when $n = 200$. With the decrease of n to 60, GSICb tends to work much better than NIC and non-regularized GSIC, and its performance is comparable to CV. Fig. 4 shows that the shape of the GSICb curve shadows the true curve nicely when $(n, \sigma) = (200, 0.3)$. Note that terms which are irrelevant to model selection are ignored also in GSICb because of the similar reason to above. When $(n, \sigma) = (60, 0.3)$, the variance of the GSICb curve is far reduced compared to that of the non-regularized GSIC curve, and its shape coincides very well with the true curve. This implies that the introduction of regularization parameter α for obtaining a reference estimator drastically reduces the variance with an irrelevant effect on the bias. Therefore, GSICb works well even for small samples.

The computation times of GSICb and CV are plotted in Fig. 5. The plot of GSICb shows the overall computation time which includes the model selection procedure for choosing α. The plot of CV shows the computation time of actually performing the 10-fold cross validation procedure by repeating to solve the quadratic programming problem. In this experiment GSICb is much faster than CV, and the advantage increases as n becomes larger.

In summary, this illustrative one dimensional experiment shows that non-regularized GSIC performs well when n is large, but it can become unstable for small sample cases. Although it is heuristically derived, GSICb works comparably well as CV in the cases studied and it is computationally much more efficient than CV. For the experimental results with multidimensional data, please refer to [10].

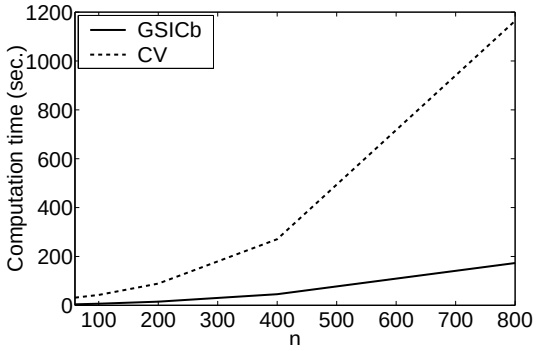

Figure 5: Computation time. The horizontal axis denotes the number of training examples and the vertical axis denotes the computation time in seconds.

7 Concluding Remarks

In this paper, we proposed GSIC and GSICb, two generalization error prediction methods for non-quadratic regularizers. They extend SIC, whose range of application is limited to quadratic regularizers. Theoretically, the bias of GSIC was shown to vanish asymptotically. In experiments, GSIC worked well with larger samples in its original form, and its regularized variant GSICb worked excellently even for small sample sizes. Future work will focus on theoretical aspects of choosing reference estimators for GSICb. Furthermore we plan to apply GSIC(b) to classification and unsupervised learning.

Acknowledgements K.R.M gratefully acknowledges partial financial support from DFG under contracts JA 379/91, MU 987/11 and the EU in the Neurocolt 2 and the BLISS project (IST-1999-14190). We thank Shunichi Amari, Hidemitsu Ogawa, Takashi Onoda, Gunnar Rätsch and Sebastian Mika for valuable discussions. MS would like to thank GMD for warm hospitality.

A Derivation of GSIC

In this section, we will show the derivation of (13). Because we assumed that the solution of the learning problem is unique, $\hat{\boldsymbol{\theta}}$ is considered as a function of $\boldsymbol{y}$. Let $\boldsymbol{z} := (f(x_1), \ldots, f(x_n))^T$ and $\boldsymbol{\epsilon} := (\epsilon_1, \ldots, \epsilon_n)^T$. Also, let the derivatives of $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ be denoted as

$$\nabla \hat{\theta}_i(\boldsymbol{y}) := \left(\frac{\partial \hat{\theta}_i}{\partial y_1}(\boldsymbol{y}), \ldots, \frac{\partial \hat{\theta}_i}{\partial y_n}(\boldsymbol{y}) \right)^T, \qquad (18)$$

and $\nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y}) := (\nabla\hat{\theta}_1(\boldsymbol{y}), \cdots, \nabla\hat{\theta}_p(\boldsymbol{y}))^T$. Then, $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ can be expressed via Taylor expansion as follows:

$$\hat{\theta}_i(\boldsymbol{y}) = \hat{\theta}_i(\boldsymbol{z}+\boldsymbol{\epsilon}) = \hat{\theta}_i(\boldsymbol{z}) + \nabla\hat{\theta}_i(\boldsymbol{z})^T\boldsymbol{\epsilon} + S_i, \quad (19)$$

where S_i is the residual. Then, w_i (i-th element of $\boldsymbol{w}$) is described as

$$w_i = \hat{\theta}_i(\boldsymbol{z}+\boldsymbol{\epsilon}) - E_{\boldsymbol{\epsilon}}[\hat{\theta}_i(\boldsymbol{z}+\boldsymbol{\epsilon})] = \nabla\hat{\theta}_i(\boldsymbol{z})^T\boldsymbol{\epsilon} + S_i - E_{\boldsymbol{\epsilon}}[S_i].$$

Expressing an unbiased estimator $\hat{\boldsymbol{\theta}^u}(\boldsymbol{y})$ by (10), $\nabla\hat{\boldsymbol{\theta}^u}(\boldsymbol{y})$ is given as

$$\nabla\hat{\boldsymbol{\theta}^u} = (K^TK)^{-1}K^T, \quad (20)$$

and hence v_i (i-th element of $\boldsymbol{v}$) is described as

$$v_i = \nabla\hat{\theta}_i^{uT}\boldsymbol{\epsilon}. \quad (21)$$

Now $E_{\boldsymbol{\epsilon}}\langle\boldsymbol{w},\boldsymbol{v}\rangle$ is written as

$$E_{\boldsymbol{\epsilon}}\langle\boldsymbol{w},\boldsymbol{v}\rangle = \sum_{i=1}^{p}\sum_{j=1}^{p} P_{ij}E_{\boldsymbol{\epsilon}}[w_iv_j], \quad (22)$$

where $E_{\boldsymbol{\epsilon}}[w_iv_j]$ is expressed as

$$E_{\boldsymbol{\epsilon}}[w_iv_j] = \sigma^2\nabla\hat{\theta}_i(\boldsymbol{z})^T\nabla\hat{\theta}_j^u + E_{\boldsymbol{\epsilon}}[S_i(\nabla\hat{\theta}_j^{uT}\boldsymbol{\epsilon})]. \quad (23)$$

Here, we approximate $E_{\boldsymbol{\epsilon}}[w_iv_j]$ by

$$E_{\boldsymbol{\epsilon}}[w_iv_j] \approx \sigma^2\nabla\hat{\theta}_i(\boldsymbol{y})^T\nabla\hat{\theta}_j^u, \quad (24)$$

i.e., the second term of (23) is ignored and $\boldsymbol{z}$ in the first term is replaced by $\boldsymbol{y}$. The analysis of the error due to the approximation using Eq.(24) will be given in [10] under several assumptions. Then we obtain $E_{\boldsymbol{\epsilon}}\langle\boldsymbol{w},\boldsymbol{v}\rangle \approx \sigma^2\mathrm{tr}(PW^0)$ where

$$W^0 = \nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y})\nabla\hat{\boldsymbol{\theta}^u}^T. \quad (25)$$

The derivatives $\nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ can be obtained from the saddle point equation,

$$\frac{\partial L_r}{\partial\theta_i} = 0, \ (i = 1, \ldots, p). \quad (26)$$

Differentiating the above equation with respect to y_k, we have $H\nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y}) + M = \boldsymbol{0}$, where H is a $p \times p$ matrix whose (i,j) element is $H_{ij} = \frac{\partial^2 L_r}{\partial\theta_i\partial\theta_j}$, and M is a $p \times n$ matrix whose (i,j) element is $M_{ij} = \frac{\partial^2 L_r}{\partial\theta_i\partial y_j}$. When H is invertible, $\nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ is described as

$$\nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y}) = -H^{-1}M. \quad (27)$$

Substituting (2) to (27), we have

$$\nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y}) = \frac{1}{n}\left(\frac{1}{n}K^TK + \frac{1}{2}\lambda\nabla\nabla R(\hat{\boldsymbol{\theta}}(\boldsymbol{y}))\right)^{-1}K^T. \quad (28)$$

Consequently, (14) is derived by substituting (20) and (28) into (25).

References

[1] V.N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[2] N. Murata, S. Yoshizawa, and S. Amari, "Network information criterion — determining the number of hidden units for an artificial neural network model," *IEEE Transactions on Neural Networks*, vol. 5, pp. 865–872, 1994.

[3] M. Sugiyama and H. Ogawa, "Subspace information criterion for model selection," *Neural Computation*, vol. 13, no. 8, 2001.

[4] M. Sugiyama and H. Ogawa, "Theoretical and experimental evaluation of subspace information criterion," *Machine Learning, Special Issue on New Methods for Model Selection and Model Combination*, 2001, to appear.

[5] C.L. Mallows, "Some comments on C_P," *Technometrics*, vol. 15, no. 4, pp. 661–675, 1973.

[6] D.J.C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992.

[7] K.P. Bennett and O.L. Mangasarian, "Robust linear programming discrimination of two linearly inseparable sets," *Optimization Methods and Software*, vol. 1, pp. 23–34, 1992.

[8] M.J.L. Orr, "Introduction to radial basis function networks," Tech. Rep., Centre for Cognitive Science, University of Edinburgh, 1996.

[9] V.V. Fedorov, *Theory of Optimal Experiments*, Academic Press, New York, 1972.

[10] K. Tsuda, M. Sugiyama, and K.-R. Müller, "Subspace information criterion for non-quadratic regularizers," Tech. Rep. 120, GMD, 2000, http://wsv.gmd.de/aiw/report.htm.

[11] G. Wahba, *Spline Model for Observational Data*, vol. 59 of *Series in Applied Mathematics*, SIAM: Society for Industrial and Applied Mathematics, Philadelphia and Pennsylvania, 1990.

[12] P.M. Williams, "Bayesian regularisation and pruning using a Laplace prior," *Neural Computation*, vol. 7, no. 1, pp. 117–143, 1995.

[13] O.L. Mangasarian, "Mathematical programming in data mining," *Data Mining and Knowledge Discovery*, vol. 42, no. 1, pp. 183–201, 1997.

[14] S. Mika, G. Rätsch, and K.-R. Müller, "A mathematical programming approach to the kernel fisher algorithm," to appear in Neural Information Processing Systems 13, 2001.