Review:

Support Vector Machine and Generalization

Takio Kurita

Neurosceince Research Institute,
National Institute of Advanced Indastrial Science and Technology
Umezono 1-1-1, Tsukuba, Ibaraki 305-8568, Japan
E-mail: takio-kurita@aist.go.jp

[Received August 11, 2003; accepted December 1, 2003]

The support vector machine (SVM) has been extended to build up nonlinear classifiers using the kernel trick. As a learning model, it has the best recognition performance among the many methods currently known because it is devised to obtain high performance for unlearned data. This paper reviews how to enhance generalization in learning classifiers centering on the SVM.

Keywords: Perceptron, support vector machine, logistic regression, generalization, shrinkage method, ridge regression, feature selection, model selection, weight decay

1. Preface

The Support Vector Machine (SVM) has been extended to build up nonlinear classifiers using the kernel trick [1–3]. As a learning model, it has the best recognition performance among the many methods currently known because it is devised to obtain high performance for unlearned data. The SVM uses linear threshold elements to build up two-classes classifier. It learns linear threshold element parameters based on "margin maximization" from training samples. This paper reviews how to enhance generalization in learning classifiers. The SVM is introduced, then multiple regression analysis (MRA) and logistic regression analysis (LRA) are explained as the statistical methods for building up a classifier with a structure similar to that for the SVM. The same method as used for the SVM can be introduced in both MRA and LRA to enhance performance for unlearned samples. This paper reviews how to enhance generalization in classifier learning and compares the SVM with these methods at the criterion function level[4].

2. Support Vector Machine

The SVM originated from the Optical Separating Hyperplane (OSH) developed by Vapnik et al. in the 1960s was extended to a nonlinear classifer combined with learning with kernels in the 1990s [1]. This extended version of the SVM that enables nonlinear classification is a learning model with the best pattern recognition performance among presently known methods. The SVM builds up a classifier that basically identifies two classes. It requires additional techniques such as a combination of multiple SVMs to build up a multiclass classifier. This section outlines how to build up a classifier from training samples using the SVM. Generalization performance must usually be enhanced for a learned classifier to demonstrate high recognition performance for unlearned data not contained in training samples. The SVM uses the criterion of "gmargin maximization" to do so.

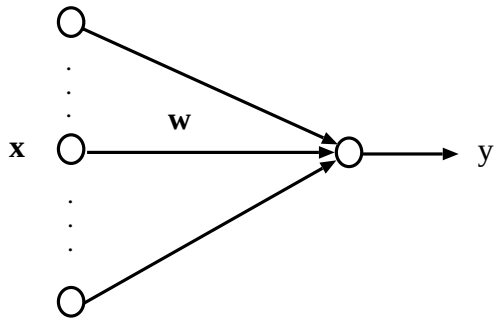


Fig. 1. Linear threshold element.

2.1. Learning by Support Vector Machine

To implement a classifier for pattern recognition, features from the object to be recognized must be extracted. In most cases, multiple features are measured and used simultaneously represented as feature vector $\boldsymbol{x}^T = (x_1, \ldots, x_M)$, where $\boldsymbol{x}^T$ represents the transpose of vector $\boldsymbol{x}$ and M is the number of features.

The SVM uses the simplest linear threshold element as a neuron model to build up two-classes classifier. The linear threshold element is a model with simplified neurons (**Fig. 1**) that calculates binary output for the input feature

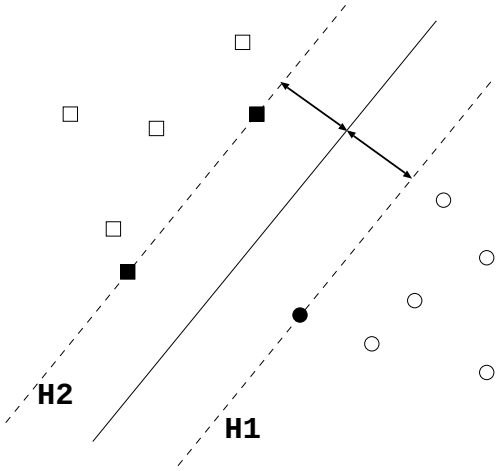


Fig. 2. Separated super plane and margin of the linear threshold element (○ and □ indicate class 1 and class −1 samples. ● and ■ indicate support vectors.)

vector using the linear discriminant function as follows:

$$y = \text{sign}(\boldsymbol{w}^T\boldsymbol{x} - h) \quad \ldots\ldots\ldots\ldots \quad (1)$$

where $\boldsymbol{w}$ is a set of parameters corresponding to a synaptic weight and h is a threshold. Function $\text{sign}(u)$ is a sign function of 1 when $u > 0$ or -1 when $u \leq 0$. This model outputs 1 if the inner product of the input vector and synaptic weights exceeds the threshold or -1 if not. This is geometrically equivalent to having the input feature space divided into two by the linear discriminant function.

Assume two classes to be C_1 and C_2 and digitize their labels as 1 and -1. Also assume that N feature vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ and correct answer class labels $t_1, \ldots, t_N$ are given to their samples as a training sample set. The training set is assumed to be classified without error by properly adjusting the parameters of the linear threshold element. Such a training set is called a "linearly separable" set.

Even if the training sample set is linearly separable, however, parameters for classifying it without error are not determined uniquely. The SVM finds the decision hyperplane with the largest margin. The margin is defined as any positive distance from the decision hyperplane to training samples. If the training sample set is linearly separable, parameters exist that satisfy the following conditions:

$$t_i(\boldsymbol{w}^T\boldsymbol{x}_i - h) \geq 1, \quad i = 1, \ldots, N \quad \ldots\ldots \quad (2)$$

This shows that training samples are completely separated on two hyperplanes of H1: $\boldsymbol{w}^T\boldsymbol{x} - h = 1$ and H2: $\boldsymbol{w}^T\boldsymbol{x} - h = -1$, and that there is no sample between these two hyperplanes. The distance (margin) is $\frac{1}{||\boldsymbol{w}||}$ between the decision hyperplane and these hyperplanes. A problem that finds parameters $\boldsymbol{w}$ and h by maximizing the margin is equivalent to a problem that finds parameters minimizing the objective function

$$L(\boldsymbol{w}) = \frac{1}{2}||\boldsymbol{w}||^2 \quad \ldots\ldots\ldots\ldots \quad (3)$$

under the following constraints

$$t_i(\boldsymbol{w}^T\boldsymbol{x}_i - h) \geq 1, \quad (i = 1, \ldots, N). \quad \ldots\ldots \quad (4)$$

This optimization problem is known as quadratic programming in mathematical programming and a variety of numerical calculations have been proposed. An unconstrained problem is obtained by Lagrange underermined multipliers $\alpha_i (\geq 0)$, $i = 1, \ldots, N$. The objective function is rewritten as follows:

$$L(\boldsymbol{w}, h, \boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{N} \alpha_i \{t_i(\boldsymbol{w}^T\boldsymbol{x}_i - h) - 1\} \quad (5)$$

At the stationary point, the following relationship is established from the partial derivatives of the objective function for parameters $\boldsymbol{w}$ and h:

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x}_i \quad \ldots\ldots\ldots\ldots \quad (6)$$

$$0 = \sum_{i=1}^{N} \alpha_i t_i \quad \ldots\ldots\ldots\ldots \quad (7)$$

When they are substituted into the objective function above, a dual problem is obtained that maximizes the object function

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{x}_i^T \boldsymbol{x}_j \quad \ldots\ldots \quad (8)$$

under the following constraints

$$\sum_{i=1}^{N} \alpha_i t_i = 0 \quad \ldots\ldots\ldots\ldots \quad (9)$$

$$0 \leq \alpha_i, \quad i = 1, \ldots, N. \quad \ldots\ldots \quad (10)$$

This is an optimization problem with Langrange multipliers α_i (≥ 0), $i = 1, \ldots, N$ as unknown parameters. In solutions, training sample $\boldsymbol{x}_i$, for which α_i^* is not 0, i.e., $\alpha_i^* > 0$, is on one of the hyperplanes $\boldsymbol{w}^T\boldsymbol{x} - h = 1$ and $\boldsymbol{w}^T\boldsymbol{x} - h = -1$. From this, training sample $\boldsymbol{x}_i$, for which α_i^* is not 0, is called the "support vector." This is how the SVM got its name. The number of support vectors is generally fewer than the number of training samples, meaning that a small number of support vectors is automatically selected from a large number of training samples by maximizing the margin.

From optimal solution $\alpha_i^* (i \geq 0)$ of the dual problem and the conditional expression at the stationary point, optimal parameters $\boldsymbol{w}^*$ are given as

$$\boldsymbol{w}^* = \sum_{i \in S} \alpha_i^* t_i \boldsymbol{x}_i \quad \ldots\ldots\ldots\ldots \quad (11)$$

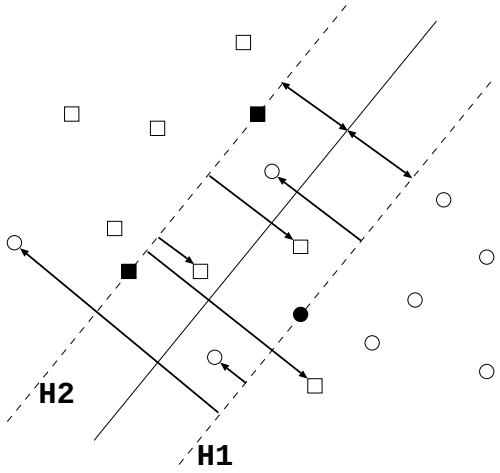


Fig. 3. Soft margin (◯ and □ indicate class 1 and class −1 samples. ● and ■ indicate support vectors.)

where S is a set of subscripts corresponding to support vectors. Optimal threshold h^* is obtained from the relationship holding that they are on one of two hyperplanes $\boldsymbol{w}^T\boldsymbol{x}-h=1$ and $\boldsymbol{w}^T\boldsymbol{x}-h=-1$, i.e., it can be found by the following expression from any support vector, $\boldsymbol{x}_s, s\in S$:

$$h^* = \boldsymbol{w}^{*T}\boldsymbol{x}_s - t_s \quad \ldots\ldots\ldots\ldots\ldots (12)$$

The optimal classifier is also expressed as

$$\begin{aligned} y &= \mathrm{sign}(\boldsymbol{w}^{*T}\boldsymbol{x}-h^*) \\ &= \mathrm{sign}(\sum_{i\in S}\alpha_i^* t_i \boldsymbol{x}_i^T\boldsymbol{x}-h^*) \quad \ldots\ldots (13)\end{aligned}$$

by using optimal solution $\alpha_i^*(i\geq 0)$ of the dual problem. In other words, the large number of training samples of $\alpha_i^*=0$ is ignored and only the small number of training samples with $\alpha_i^*>0$ close to the discriminant hyperplane are used to build up the classifier. The key here is that only a small number of training samples near the discriminant hyperplane is automatically selected from the criterion of the "margin maximization." This leads to a certain degree of good generalization performance that can be maintained for unlearned data. The SVM uses the criterion of margin maximization to select training samples, causing models that suppress the degree of freedom to be selected.

2.2. Soft Margin

The above discussion concerns the case in which training samples are linearly separable, but this rarely happens in actual pattern recognition, meaning that improvements are required before the SVM can be applied to actual problems. One approach is to release restrictions on separability and allow more or fewer recognition errors to be accepted. This is called a "soft margin."

The soft margin permits some samples to move to the opposite side beyond hyperplanes H1 or H2 (**Fig. 3**) while maximizing margin $\frac{1}{||\boldsymbol{w}||}$. Assuming that the distance into the opposite side is represented as $\frac{\xi_i}{||\boldsymbol{w}||}$ using parameter $\xi_i(\geq 0)$, the sum is

$$\sum_{i=1}^{N}\frac{\xi_i}{||\boldsymbol{w}||}. \quad \ldots\ldots\ldots\ldots\ldots (14)$$

This sum must be as small as possible. The problem that finds the optimal descriminant hyperplane from these requirements is definded as finding a parameter minimizing the objective function

$$L(\boldsymbol{w},\boldsymbol{\xi}) = \frac{1}{2}||\boldsymbol{w}||^2 + \gamma\sum_{i=1}^{N}\xi_i \quad \ldots\ldots\ldots (15)$$

under the following constraints

$$\xi_i \geq 0, \quad t_i(\boldsymbol{w}^T\boldsymbol{x}_i - h)\geq 1-\xi_i, \quad (i=1,\ldots,N) \quad (16)$$

Newly introduced parameter γ is a constant that controls the balance between the size of the margin in the first term and the degree of overflow in the second term. By introducing Lagrange multiplier α_i and ν_i for two constraints, the objective function is rewritten as

$$\begin{aligned} L(\boldsymbol{w},h,\boldsymbol{\alpha},\boldsymbol{\nu}) &= \frac{1}{2}||\boldsymbol{w}||^2 + \gamma\sum_{i=1}^{N}\xi_i \\ &\quad -\sum_{i=1}^{N}\alpha_i\{t_i(\boldsymbol{w}^T\boldsymbol{x}_i - h)-(1-\xi_i)\} \\ &\quad -\sum_{i=1}^{N}\nu_i\xi_i \quad \ldots\ldots\ldots (17)\end{aligned}$$

Assuming partial derivatives of this objective function for parameters $\boldsymbol{w}, h$, and ν_i to be 0, the following relationships can be established at the stationary point:

$$\begin{aligned} \boldsymbol{w} &= \sum_{i=1}^{N}\alpha_i t_i \boldsymbol{x}_i \quad \ldots\ldots\ldots\ldots (18) \\ 0 &= \sum_{i=1}^{N}\alpha_i t_i \quad \ldots\ldots\ldots\ldots (19) \\ \alpha_i &= \gamma - \nu_i \quad \ldots\ldots\ldots\ldots (20)\end{aligned}$$

If they are substituted into the objective function, a dual problem is obtained that maximizes the object function

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j t_i t_j \boldsymbol{x}_i^T\boldsymbol{x}_j \quad \ldots\ldots (21)$$

under the following constraints

$$\begin{aligned} \sum_{i=1}^{N}\alpha_i t_i &= 0 \quad \ldots\ldots\ldots\ldots (22) \\ 0 &\leq \alpha_i \leq \gamma, \quad i=1,\ldots,N. \quad \ldots\ldots (23)\end{aligned}$$

If training samples are linearly separable, samples are classified by optimal solution α_i^* into support vectors on

hyperplanes H1 and H2 and other samples, but for the soft margin, some samples overflow to the opposite side of hyperplanes H1 or H2. If $\alpha_i^* = 0$, the corresponding sample is classified correctly by the learned classifier. For $0 < \alpha_i^* < \gamma$, the sample becomes the support vector existing on hyplerplanes H1 or H2 and is also correctly classified. For $\alpha_i^* = \gamma$, the sample becomes the support vector, but $\xi_i \neq 0$ occurs and the sample is misclassified by the learned classifier.

3. Linear Classifier and Generalization

This section discusses how the SVM is related to conventional statistical pattern recognition.

3.1. Classifier Using a Linear Threshold Element

The SVM is a classifier using a linear threshold element. The perceptron proposed by Rosenblatt is also a classifier that learns from training samples using linear threshold element [5]. It is called a simple perceptron to distinguish it from a multilayer perceptron. Like the SVM, the simple perceptron calculates output y for input $\boldsymbol{x} = (x_1, \ldots, x_M)^T$ as follows:

$$\begin{aligned} y &= f(\eta) \\ \eta &= \boldsymbol{w}^T\boldsymbol{x} - h = \tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{x}} \quad \ldots\ldots\ldots \quad (24) \end{aligned}$$

where w_i is a synaptic weight linked from the ith input to output and h is a threshold. For simplification, it is represented as $\tilde{\boldsymbol{w}} = (h, w_1, \ldots, w_M)^T$. A vector with a constant term added to the input feature vector is represented as $\tilde{\boldsymbol{x}} = (-1, x_1, \ldots, x_M)^T$. The original model proposed by Rosenblatt used the following threshold function as activation function f of the output unit:

$$f(\eta) = \mathrm{sign}(\eta) = \begin{cases} 1 & \text{if } \eta \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad \ldots\ldots (25)$$

The linear (27) or logistic (28) function is often used as an activation function.

$$f(\eta) = \eta \quad \ldots\ldots\ldots\ldots\ldots\ldots (26)$$

$$f(\eta) = \frac{\exp(\eta)}{1+\exp(\eta)} \quad \ldots\ldots\ldots\ldots (27)$$

In multivariate data analysis, a simple perceptron with a linear function corresponds to a linear multiple regression model, while a simple perceptron with a logistic function corresponds to a logistic regression model.

3.2. Learning of Simple Perceptron

Learning algorithms have been proposed to estimate synaptic weights (parameters) of the simple perceptron. The original method proposed by Rosenblatt et al. attempted to classify training samples by the preceptron and, if it failed, synaptic weights were modified to correct errors. This learning algorithm may not converge to a solution, however, even by infinitely repeating procedures, if training samples are not linearly separable. There is no guarantee that parameters obtained when learning is aborted before completion are optimal.

3.3. Linear Multiple Regression Analysis

If the linear function is used as the activation function of the output unit and mean squared errors between the teacher signal and the output of the classifier is used to estimate synaptic weights, the optimal solution is found by matrix calculation.

Assume a set of N training samples to be $\{(\boldsymbol{x}_i, t_i) | i = 1, \ldots, N\}$, where $\boldsymbol{x}_i$ an input vector and t_i is the desired output (teacher signal) to the input vector. Also assume a matrix of $N \times (M+1)$ dimensions that arranges input vectors of training samples to be $X = (\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_N)^T$ and the vector of N dimensions that arranges teacher signals to be $\boldsymbol{t} = (t_1, \ldots, t_N)^T$. The sum of squared errors is expressed as follows:

$$\varepsilon_{emp}^2 = \sum_{i=1}^{N} (t_i - y_i)^2 = ||\boldsymbol{t} - X\tilde{\boldsymbol{w}}||^2 \quad \ldots\ldots (28)$$

By taking the derivatives with respective to parameter $\tilde{\boldsymbol{w}}$ to 0, we have the following equation:

$$\frac{\partial \varepsilon_{emp}^2}{\partial \tilde{\boldsymbol{w}}} = X^T(\boldsymbol{t} - X\tilde{\boldsymbol{w}}) = 0 \quad \ldots\ldots\ldots (29)$$

Thus, if (X^TX) is nonsingular, optimal parameter $\tilde{\boldsymbol{w}}^*$ is given as follows:

$$\tilde{\boldsymbol{w}}^* = (X^TX)^{-1}X^T\boldsymbol{t} \quad \ldots\ldots\ldots\ldots (30)$$

3.4. Methods to Improve Generalization for Multiple Regression Analysis

Multiple regression analysis, a basic form of multivariate data analysis, is used in a wide variety of fields to build up estimation models from a set of training samples for the purpose of estimating the value of the object variable from unlearned explanation variables not included in the training sample set. If the constructed estimation model fails to output good estimates for unlearned samples, it is meaningless to build it up. The estimation performance for unlearned samples, called generalization, is an important element for building up estimation models. Typical ways of improving generalization include variable selection, shrinkage, and regularization. The sections that follow introduce variable selection and ridge regression, which is an example of shrinkage.

(1) Variable Selection

Input feature vector $\boldsymbol{x}$ may contain features both useful and not useful for estimation. In an extreme case, if

the input feature vector contains features not at all related to estimation, they either do not operate effectively for estimating from unlearned samples, or, worse, they may interfere with estimation. If the number of input features is larger than the number of training samples, parameters of the estimation model may not be able to be determined uniquely, in which case, it becomes necessary to select a subet of features effective in estimating from among given features and to build up the estimation model from this subset, i.e., variable selection.

Estimation performance for all possible subsets of features must be evaluated to select the best subset of features. Increasing the number of features, however, increases the number of subsets exponentially. If there are many features, it is thus not realistic to evaluate all possible subsets. One way to search for a subset of relatively good features is called forward or backward stepwise selection. Forward stepwise selection starts with a model with only one feature and adds features one by one to find the best subset. Backward stepwise selection, in contrast, removes features one by one from a model including all features. The subset of features may also be selected by using genetic algorithms.

In selecting variables, a criterion to evaluate estimation performance of the model must be specified when learning has been completed for the subset of features. The sum of the squared error criterion of training samples, described previously, decreases with the increasing number of features, preventing the selection of a subset of features based on this criterion.

The generalization performance of a model is defined as estimation performance for unknown samples. If many samples other than training samples can be collected relatively easily, it is possible to evaluate the generalization performance of the model by using these samples. Specifically, it is possible to prepare a set of samples for evaluating generalization performance and to select a subset of features so that maximizes estimation performance for these samples. This is easiest and most direct, and should be attempted if many samples can be easily collected in addition to training samples.

If it is difficult to collect many samples and the number of samples is small, it is difficult to prepare samples other than training samples. In such case, generalization performance must be evaluated from these training samples alone. A rather large amount of computation is required, but generalization performance can be evaluated relatively simply if the required computing power is available, in what is called resampling. The simplest form of resampling is to "leave-one-out." If N samples are given, leave-one-out method divides them into $N-1$ training samples and one evaluation sample. Learning results using $N-1$ training samples are used to evaluate the one evaluation sample. The one evaluation sample may be selected N possible ways. The average of evaluation results for all these ways is used as the criterion of estimation performance. Jackknife [7, 8] and bootstrap [9–11] are more sophistcated forms of resampling. Resampling, which maximizes computer power to evaluate generalization performance, is particularly promising, given rapid advances in downsizing, price reduction, and computer availability.

Instead of the square error criterion for training samples, another evaluation criterion has been proposed that is calculated from training samples alone to evaluate generalization performance. Information criteria include the information theoretical criterion (AIC) [12] by Akaike and the minimum description length (MDL) [13, 14] introduced by Rissanen. Learning may be done only once and evaluation done relatively simply. Since learning parameters using multiple regression analysis is regarded as maximum likelihood estimation, generalizatin performance of models is compared by calculating information criteria such as AIC and MDL from logarithmic likelihood calculated using learned parameters. AIC is derived from analytical evaluation done by Akaike on differences between maximum logarithmic likelihood and expected average logarithmic likelihood. Assuming that the degree of freedom of a model is J, AIC is defined as follows:

$$AIC = -2(\text{maximum log likelihood}) + 2J \quad . \quad . \quad (31)$$

MDL, introduced by Rissanen as the principle for the minimum description length in encoding, is defined as follows:

$$MDL = -(\text{maximum log likelihood}) + \frac{N}{2}\log J \quad (32)$$

If teacher signals and estimated output on training samples differ greatly, a large difference appears in the first term. If no such large difference occurs, the second term becomes dominant and a model with a low degree of freedom is selected. To select a model with high generalization performance, learn parameters of the model with a subset of features and calculate its logarithmic likelihood. Then, select the model with the smallest AIC or MDL.

(2) Ridge regression

Variable selection attempts to build up a model with good generalization performance for unlearned samples by selecting a subset of explanation variables, but this variable selection is discrete, i.e., it either selects or does not select variables. A model may be restricted more continuously by shrinkage, exemplified by ridge regression. The criterion of multiple regression analysis is modified so that a penalty term (33) is added to the square error criterion (34). This penalty term prevents the number of

parameters from becoming too large.

$$\sum_{j=1}^{M} w_j^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots (33)$$

$$\varepsilon_{emp}^2 = \sum_{i=1}^{N}(t_i - y_i)^2 = \sum_{i=1}^{N}(t_i - (\sum_{j=1}^{M} w_j x_{ij} - h))^2 \ . \ (34)$$

Specifically, by summing them up, consider the following as the modified criterion for ridge regression and find parameters to minimize it:

$$Q(\boldsymbol{w}, h) = \sum_{i=1}^{N}(t_i - (\sum_{j=1}^{M} w_j x_{ij} - h))^2 + \lambda \sum_{j=1}^{M} w_j^2 \ . \ (35)$$

where λ is a constant for determining the balance between a square error and a penalty. Thus, if $\lambda = 0$, ridge regression becomes the same as ordinary multiple regression analysis. By taking the partial derivative of $Q(\boldsymbol{w}, h)$ for h and setting it to 0, the following expression is established:

$$\frac{\partial Q(\boldsymbol{w}, h)}{\partial h} = 2N(\bar{t} - \sum_{j=1}^{M} w_j \bar{x_j} + h) = 0 \ \ldots\ldots (36)$$

From this, the condition related to h is obtained as follows:

$$h = -\bar{t} + \sum_{j=1}^{M} w_j \bar{x_j} \ \ldots\ldots\ldots\ldots (37)$$

where $\bar{t} = \frac{1}{N}\sum_{i=1}^{N} t_i$ and $\bar{x_j} = \frac{1}{N}\sum_{i=1}^{N} x_{ij}$. Substituting these into expression $Q(\boldsymbol{w}, h)$ yields the following:

$$Q(\boldsymbol{w}) = (\tilde{\boldsymbol{t}} - \tilde{X}\boldsymbol{w})^T(\tilde{\boldsymbol{t}} - \tilde{X}\boldsymbol{w}) + \lambda \boldsymbol{w}^T\boldsymbol{w} \ \ldots\ldots (38)$$

where $\tilde{\boldsymbol{t}}$ and $\tilde{X}$ are a vector with $(t_i - \bar{t})$ as an element and a matrix with $(x_{ij} - \bar{x_j})$ as an element. Taking partial derivatives again yields the following:

$$\frac{\partial Q(\boldsymbol{w})}{\partial \boldsymbol{w}} = -2\tilde{X}\tilde{\boldsymbol{t}} + 2(\tilde{X}^T\tilde{X}\boldsymbol{w} + \lambda I)\boldsymbol{w} = 0 \ \ldots (39)$$

Optimal parameter $\boldsymbol{w}^*$ is given as follows:

$$\boldsymbol{w}^* = (\tilde{X}^T\tilde{X} + \lambda I)^{-1}\tilde{X}^T\tilde{\boldsymbol{t}} \ \ldots\ldots\ldots (40)$$

This corresponds to the fact that λ is added to the diagonal element of matrix $\tilde{X}^T\tilde{X}$ to calculate an inverse matrix. This prevents matrix $\tilde{X}^T\tilde{X}$ from becoming singular and also effectively stabilizes numerical calculation of the inverse matrix.

3.5. Logistic Regression

If the logistic function is used as an activation function and its parameters are estimated by maximum likelihood estimation, this becomes equivalent to logistic regression. Fisher's scoring algorithm is well known as a parameter estimation algorithm for logistic regression. Assume a set of training samples to be $\{(\boldsymbol{x}_i, u_i) | i = 1, \ldots, N\}$. Teacher signal u_i is assumed as binary 0 or 1.

Considering output y when input $\boldsymbol{x}$ is given as the estimated value of the probability that teacher signal u is 1 under input $\boldsymbol{x}$, the likelihood of the network for the training sample set is given as follows:

$$L = \prod_{i=1}^{N} y_i^{u_i}(1 - y_i)^{(1-u_i)} \quad \ldots\ldots\ldots (41)$$

The logarithm (log-likelihood) is therefore as follows:

$$\begin{aligned} l &= \sum_{i=1}^{N}\{u_i \log y_i + (1 - u_i)\log(1 - y_i)\} \\ &= \sum_{i=1}^{N}\{u_i\eta_i - \log\{1 + \exp(\eta_i)\}\} \quad \ldots\ldots (42) \end{aligned}$$

Parameters with maimum log-likelihood are called maximum likelihood estimates. Consider that optimal parameters are found by using steepest descent. Partial derivatives of log-likelihood for parameters are as follows:

$$\frac{\partial l}{\partial w_j} = \sum_{i=1}^{N}(u_i - y_i)x_{ij} = \sum_{i=1}^{N}\delta_i x_{ij} \quad \ldots\ldots (43)$$

where $\delta_i = (u_i - y_i)$. The partial derivative of log-likelihood for parameter h is as follows:

$$\frac{\partial l}{\partial h} = \sum_{i=1}^{N}(u_i - y_i)(-1) = \sum_{i=1}^{N}\delta_i(-1) \ \ldots\ldots (44)$$

The parameter update expression is thus as follows:

$$w_j \Leftarrow w_j + \alpha(\sum_{i=1}^{N}\delta_i x_{ij}) \quad \ldots\ldots\ldots (45)$$

$$h \Leftarrow h + \alpha(\sum_{i=1}^{N}\delta_i(-1)) \ \ldots\ldots\ldots (46)$$

The Fisher information matrix plays an important role in maximum likelihood estimation. In general, when data y follows distribution of density function $f(y, \theta_1, \ldots, \theta_M)$ with parameters $\theta_1, \ldots, \theta_M$, the following is referred to as Fisher information and matrix $F = [F_{ij}]$ as the Fisher information matrix:

$$F_{ij} = -E\left(\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log f(y, \theta_1, \ldots, \theta_M)\right) \quad \ldots (47)$$

To calculate Fisher information, second derivatives of log-likelihood must be calculated as follows:

$$\frac{\partial^2 l}{\partial\tilde{w}_k\partial\tilde{w}_j} = -\sum_{i=1}^{N}\omega_i\tilde{x}_{ik}\tilde{x}_{ij} \quad \ldots\ldots\ldots (48)$$

Note $\omega_i = y_i(1 - y_i)$. The first and second derivatives are represented collectively as follows:

$$\begin{aligned} \nabla l &= \sum_{i=1}^{N}\delta_i\tilde{\boldsymbol{x}}_i = X^T\boldsymbol{\delta}, \quad \ldots\ldots\ldots (49) \\ \nabla^2 l &= -\sum_{i=1}^{N}\omega_i\tilde{\boldsymbol{x}}_i\tilde{\boldsymbol{x}}_i^T = -X^TWX \end{aligned}$$

Note that $X^T = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_N]$, $W = diag(\omega_1, \ldots, \omega_N)$ and $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_N)^T$. Their use causes the Fisher information matrix for parameter w, that is, the minus expected value of the Hessian matrix, to be as follows:

$$F = -E(\nabla^2 l) = X^T W X \quad \ldots\ldots\ldots\ldots (50)$$

This is a correlation matrix weighted with ω_i of input vector $\{\tilde{\boldsymbol{x}}_i\}$. Weight ω_i at that time decreases when output is certain (close to 1 or 0). It increases when output is uncertain (close to 0.5). The Fisher information matrix is thus regarded as the correlation matrix of an uncertain input vector.

Finding parameters that maximize the log-likelihood requires nonlinear optimization, i.e., Fisher's scoring algorithm is used in logistic regression [15]. This is a form of Newton's method and uses the Fisher matrix instead of the Hessian matrix. For logistic regression, the Fisher matrix differs from the Hessian matrix only in the sign, meaning that the Fisher scoring algorithm becomes equivalent to Newton's method.

Assume that estimated values of present parameters are $\boldsymbol{w}$, which is updated as follows by correction vector $\delta\tilde{\boldsymbol{w}}$:

$$\tilde{\boldsymbol{w}}^* = \tilde{\boldsymbol{w}} + \delta\tilde{\boldsymbol{w}} \quad \ldots\ldots\ldots\ldots\ldots (51)$$

Correction vector $\delta\tilde{\boldsymbol{w}}$ is obtained by solving the following linear equation:

$$F\delta\tilde{\boldsymbol{w}} = \nabla l \quad \ldots\ldots\ldots\ldots\ldots (52)$$

Update expression (51) is multiplied by F from the left, resulting in the following:

$$F\tilde{\boldsymbol{w}}^* = F\tilde{\boldsymbol{w}} + F\delta\tilde{\boldsymbol{w}} = F\tilde{\boldsymbol{w}} + \nabla l \quad \ldots\ldots (53)$$

$F\tilde{\boldsymbol{w}}$ is given as follows:

$$F\tilde{\boldsymbol{w}} = X^T W\boldsymbol{\eta} \quad \ldots\ldots\ldots\ldots\ldots (54)$$

Note that $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_N)^T$. New estimated parameter vector $\tilde{\boldsymbol{w}}^*$ is thus found as follows:

$$\begin{aligned} \tilde{\boldsymbol{w}}^* &= F^{-1}(F\tilde{\boldsymbol{w}} + \nabla l) \\ &= (X^T W X)^{-1}(X^T W\boldsymbol{\eta} + X^T\boldsymbol{\delta}) \\ &= (X^T W X)^{-1} X^T W(\boldsymbol{\eta} + W^{-1}\boldsymbol{\delta}) \quad \ldots (55) \end{aligned}$$

However, $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_N)^T$. This expression is regarded as the normal equation of the method of least squares with weights from input data to objective variables $\boldsymbol{z} = \boldsymbol{\eta} + W^{-1}\boldsymbol{\delta}$. Thus, to find the maximum likelihood estimates, simply repeat this weighted least squares starting with initial parameters.

3.6. Methods to Improve Generalization for Logistic Regression

For logistic regression analysis, variable selection that selects a subset of features effective for estimation is useful in building up an estimation model with high generalization performance. Variable selection is applicable to logistic regression as is. Evaluation criteria for selecting variables includes (1) direct evaluation of estimation performance by using samples for other than training samples, (2) estimating generalization performance from training samples by resampling, and (3) evaluating generalization performance using an information criterion such as AIC or MDL.

For ridge regression, a penalty term was added to the square error criterion to prevent parameters from becoming too large. In logistic regression, similarly, add a penalty term on the log-likelihood criterion to prevent parameters from becoming too large. The objective function is expressed as follows:

$$Q(\tilde{\boldsymbol{w}}) = -l + \lambda \sum_{j=1}^{M} w_j^2 \quad \ldots\ldots\ldots\ldots (56)$$

To find a parameter to minimize the above expression, partial derivatives of $Q(\tilde{\boldsymbol{w}})$ for parameter vector w_j is calculated as follows:

$$\begin{aligned} \frac{\partial Q}{\partial w_j} &= -\frac{\partial l}{\partial w_j} + 2\lambda w_j \\ &= -\sum_{i=1}^{N}(u_i - y_i)x_{ij} + 2\lambda w_j \quad \ldots\ldots (57) \end{aligned}$$

The partial derivative of $Q(\tilde{\boldsymbol{w}})$ for parameter h is as follows:

$$\begin{aligned} \frac{\partial Q}{\partial h} &= -\frac{\partial l}{\partial h} \\ &= -\sum_{i=1}^{N}(u_i - y_i)(-1) \quad \ldots\ldots\ldots (58) \end{aligned}$$

Parameter update rules in weight decay are thus as follows:

$$w_j \Leftarrow w_j + \alpha\Big(\sum_{i=1}^{N}(u_i - y_i)x_{ij}\Big) - 2\alpha\lambda w_j \quad \ldots (59)$$

$$h \Leftarrow h + \alpha\Big(\sum_{i=1}^{N}(u_i - y_i)(-1)\Big) \quad \ldots\ldots\ldots (60)$$

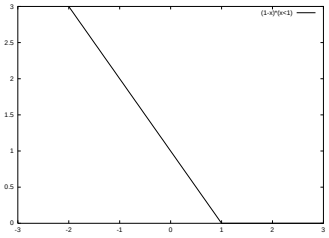
The second term of the update expression works to make absolute values of parameters as small as possible, i.e., make unnecessary parameters for estimation very small.

3.7. Support Vector Machine and Regularization

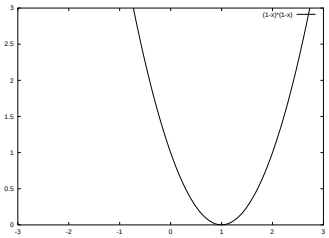
The soft margin criterion given by (15) is rewritten as follows:

$$\begin{aligned} L(\boldsymbol{w}, \boldsymbol{\xi}) &= \sum_{i=1}^{N}\xi_i + \lambda\sum_{j=1}^{M} w_j^2 \\ &= \sum_{i=1}^{N}[1 - t_i\eta_i]_+ + \lambda\sum_{j=1}^{M} w_j^2 \quad \ldots (61) \end{aligned}$$

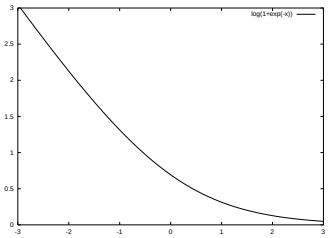
where $[x]_+$ is a function that takes only a positivevalue. **Fig. 4(a)** graphs $[1-x]_+$. The first term takes 0 when $t_i\eta_i$



(a) Support vector machine.



(b) Heavy regression analysis (ridge regression).



(c) Logistic regression (weight decay).

Fig. 4. Comparison of evaluation criteria.

is larger than 1 and gradually takes a larger value if $t_i\eta_i$ is less than 1. With this criterion, the first term is a function that evaluates the difference between teacher signals and the outputs of the model, while the second term is a so-called regularization term, i.e., a penalty imposed on parameters.

Similarly, the criterion of ridge regression is rewritten, using the fact that t_i has value -1 or 1, as follows:

$$Q = \sum_{i=1}^{N}(1-t_i\eta_i)^2 + \lambda\sum_{j=1}^{M} w_j^2 \quad \ldots\ldots\ldots (62)$$

The first term is also a function that evaluates the difference between teacher signals and the outputs of the model, while the second term is a penalty imposed on the parameter. **Fig. 4(b)** shows function $(1-x)^2$ in the first term. This function outputs a large value if $t_i\eta_i$ is apart from 1, regardless of whether its value is greater than or less than 1. This function imposes a larger penalty on a sample that is correctly identified so $t_i\eta_i$ is equal to or more than 1 if it is apart from 1. This means that the least squared errors criterion is not always effective for a classification problem.

Similarly, the criterion of the logistic regression with weight decay is rewritten as follows:

$$Q = \sum_{i=1}^{N}\log\{1+\exp(t_i\eta_i)\} + \lambda\sum_{j=1}^{M} w_j^2 \quad \ldots\ldots (63)$$

The first term is also a function that evaluates the difference between teacher signals and the outputs of the mode, while the second term is a penalty imposed on parameters. **Fig. 4(c)** graphs function $\log\{1+\exp(t_i\eta_i)\}$. This function is similar in shape to the first term of the SVM, but not discontinuous with $t_i\eta_i = 1$. Unlike ridge regression criterion, a penalty becomes small for samples correctly identified so that $t_i\eta_i$ is equal to or more than 1.

Comparing these three criteria shows them to be very similar, especially in how they impose a penalty on parameters in the second term. This means the same criterion is used to enhance generalization, even though evaluation differs with the way the difference between teacher signals and the outputs of the model are measured, but the criterion of the SVM is very similar to that of logistic regression with weight decay. The SVM is derived for two-classes classification problem, but logistic regression does not necessarily assume this and is easy to formalize so that it deals with multiple classes.

4. Conclusions

This paper introduced ways to improve generalization of a simple perceptron classifier. It also compared the SVM, multiple regression analysis, and logistic regression analysis from the viewpoint of its criterion. These comparisons and considerations clarify what the SVM does.

References:

[1] V.N.Vapnik, Statistical Learning Theory, John Wiley & Sons, 1998.

[2] B.Scholkopf, C.J.C.Burges, A.J.Smola, Advances in Kernel Methods - Support Vector Learning, The MIT Press, 1999.

[3] N.Cristianini, J.S-Taylor, An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, 2000.

[4] T.Hastie, R.Tibshirani, J.Friedman, The Elements of Statistical Learning - Data Mining, Inference, and Prediction, Springer-Verlag, 2001.

[5] R.O.Duda, P.E.Hart, D.G.Stork, Pattern Classification (Second Edition), John Wiley & Sons, 2001.

[6] K.R.Muller, S.Mika, G.Ratsch, K.Tsuda, B.Scholkopf, "An introduction to kernel-based learning algorithms," IEEE Trans. On Neural Networks, Vol.12, No.2, pp.181-201, 2001.

[7] Miller,R.G, "The jacknife -a review," Biometrika, Vol.61, No.1, pp.1-15, 1974.

[8] Stone,M., "Cross-validatory choice and assessment of statistic al predictions," Journal of Royal Statistical Society, Vol.B36, pp.111-147, 1974.

[9] Efron,B., "Bootstrap methods: anothoe look at the jackknife," The Annals of Statistics, Vol.7, No.1, pp.1-26, 1979.

[10] Efron,B., "Estimating the error rate of a prediction rule: imp rovements in cross-validation," Journal of Amerian Statistical Association, Vol .78, pp.316-331, 1983.

[11] Efron,B., "The bootstrap method for assessing statistical accu racy," Behaviormetrika, Vol.17, pp.1-35, 1985.

[12] Akaike,H., “A new look at the statistical model identification,” IEEE Trans. on Automatic Control, vol.AC-19, No.6, pp.716-723, 1974.
[13] Rissanen,J., “A universal prior for integers and estimation by minimum description length,” The Annals of Statistics, Vol.11, NO.2, pp.416-431, 1983.
[14] Rissanen,J., “Stochastic complexity and modeling,” The Annals of Statistics, Vol.14, No.3, pp.1080-1100, 1986.
[15] P.McCullagh, and J.A.Nelder FRS, “ Generalized Linear Models,” Chapman and Hall, 1989.

Name:
Takio Kurita

Affiliation:
Neuroscience Research Institute, National Institute of Advanced Industrial Science and Technology

Address:
Umezono 1-1-1, Tsukuba, Ibaraki 305-8568, Japan

Brief Biographical History:
1981- Joined Electrotechnical Laboratory
1990-1991 Visiting Scientist, Institute for Information Technology, NRC, Ottawa, Canada
2001- Neuroscience Research Institute, National Institute of Advanced Industrial Science and Technology

Main Works:
• “Iterative Weighted Least Squares Algorithms for Neural Networks Classifiers,” New Generation Computing, Vol.12, pp.375-394 (1994)

Membership in Learned Societies:
• The Institute of Electronics, Information and Communication Engineers (IEICE)
• IEEE Computer Society
• Information Processing Society of Japan (IPSJ)
• Japanese Neural Network Society (JNNS)
• The Behaviormetric Society of Japan, Japanese Academy of Facial Studies