ROBUST BOUNDARY LEARNING FOR MULTI-CLASS CLASSIFICATION PROBLEMS

Yoshikazu Washizawa

Brain Science Institute, RIKEN
washizawa@brain.riken.jp

Seiji Hotta

Tokyo University of Agriculture and Technology
s-hotta@cc.tuat.ac.jp

ABSTRACT

The objective of pattern classification is minimizing generalization errors for innumerable unknown samples. In the structural risk minimization (SRM) principle, both empirical errors and complexities of classifiers are minimized instead of minimizing generalization errors. We define a criterion about both of empirical errors and complexities for multi-class classifiers directly, and propose a perceptron-based linear classifier obtained as the minimum solution of the criterion. Due to this direct measurement, our classifier is robust against outliers and mislabeled training samples. We discuss the advantages of our classifier by comparing with conventional classifiers such as support vector machines and neural networks. We verify classification ability of our classifier by experiments on benchmark datasets.

Index Terms— support vector machines, structural risk minimization, regularization, multi-class classification

1. INTRODUCTION

The objective of pattern classification is minimizing generalization errors for innumerable unknown samples. Since we cannot collect infinite samples and estimate probability densities of them, we minimize an error of finite samples instead of that of infinite ones. This is called *empirical risk minimization* (ERM). However it is well-known that ERM does not always minimize generalization errors. Let $\eta = R - R^{\text{emp}}$ be a difference between a generalization error R and an empirical error R^{emp}. In the *structural risk minimization* (SRM) principle, $R^{\text{emp}} + \eta$ is minimized instead of minimizing R [1].

It is well-known that η has following two properties:

1. η is monotonically decreasing with respect to the number of samples.

2. η is monotonically increasing with respect to complexities of classifiers.

The first property is derived from statistics. The second is so-called Occam's razor principle which is mainly derived from experiences, and also suggested from bias-variance decomposition, *minimum description length* (MDL) principle, and *probably approximately correct* (PAC) learning framework [2, 3].

Since the number of samples is limited in general classification problems, η should be suppressed by using the second property, and several approaches are proposed. For example, we can apply "early stopping" to iterative learning such as the back propagation in multi-layer neural networks, and use model selection such as selection of the number of hidden units for neural networks. In subspace classifiers and rank reduced regressions, rank limitation can be applied to them. The cost functions of *support vector machines* (SVMs) include (Tikhonov-Phillips) regularization [4], and regularization is also used in weight decay neural networks [2].

In this paper, we propose a perceptron-based linear classifier that is designed with novel learning algorithms for multi-class classification problems. In our learning, a criterion is defined by measuring R^{emp} and η directly using regularization. Due to this direct measurement, our learning is robust against outliers and mislabeled training samples compared to SVMs. In a classification phase, an unknown sample is classified into the class that maximizes an inner product of an unknown sample and model parameters. The minimum empirical error classifier is introduced [5]. However, it is only for binary classification problems, and there is no learning method that counts error in learning samples and regularization simultaneously. Since we explain our classifier in comparison to SVMs, we start from brief explanation of SVM in Section 2. Then Section 3 describes our learning algorithms. Experimental results for benchmark datasets are shown in Section 4. We discuss our classifier with conventional classifiers in Section 5.

2. SUPPORT VECTOR MACHINES

Let $\boldsymbol{x} \in \mathbb{R}^d$ be an input vector. The decision function of linear SVM is given by

$$f(\boldsymbol{x}) = \text{sign}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b),\ \mathbb{R}^d \to \{-1, +1\}, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes an inner product, $\text{sign}(a)$ indicates a sign of a, and $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are model parameters.

Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^l \subset \mathbb{R}^d \times \{-1, +1\}$ be a set of labeled training samples. In hard margin SVM, $\boldsymbol{w}$ and b can be obtained from

solution of a convex quadratic optimization problem [1]:

$$\begin{array}{ll} \min\limits_{\boldsymbol{w}} : & \dfrac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} : & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 \quad {}^\forall i. \end{array} \tag{2}$$

Usually the problem is solved by using its dual problem. Note that the constraint means that an empirical error R^{emp} is zero, and a complexity of the model is measured by $\|\boldsymbol{w}\|^2$.

For non-linearly separable cases, a softmargin technique is adopted to hard margin SVM as follows:

$$\begin{array}{ll} \min\limits_{\boldsymbol{w},\xi_i} : & \dfrac{1}{2}\|\boldsymbol{w}\|^2 + C\sum\limits_{i=1}^{l}\xi_i \\ \text{subject to} : & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - \xi_i \quad {}^\forall i \\ & \xi_i \geq 0 \quad {}^\forall i, \end{array} \tag{3}$$

where ξ_i $(i = 1, \ldots, l)$ are called slack variables. In this case, the first term of the objective measures a complexity of the model and the second measures an empirical error, and a hyper-parameter C controls balances of the complexity and the empirical error.

SVMs are binary classifiers, we therefore should often use extensions of binary classifiers such as one-against-all method that does not give an optimal boundary for multi-class problems. For overcoming this difficulty, Multi-class SVM has been proposed for solving multi-class problems directly [6]. However, since softmargin SVM and multi-class SVM measure an empirical error with summation of excesses from its boundary, they are sensitive against outliers and mislabeled training samples [7].

3. OPTIMAL BOUNDARY CLASSIFIER

Now we discuss our perceptron-based linear classifier with learning. In our learning, we define a criterion about both of empirical errors and complexities for multi-class classifiers directly. The model parameters are given by minimizing the criterion. The objective function of our learning is not convex, so we derive the local minimum of it by gradient-based optimization. In a classification phase, an unknown sample is classified into the class that maximizes an inner product of an unknown sample and model parameters.

3.1. Model

Consider a multi-class perceptron model:

$$f(\boldsymbol{x}) = \underset{i=1,\ldots,c}{\operatorname{argmax}} \langle \boldsymbol{e}_i, \boldsymbol{W}^\top \boldsymbol{x} \rangle, \quad \mathbb{R}^d \to \{1, \ldots, c\}, \tag{4}$$

where c is the number of classes, and $\boldsymbol{e}_i \in \mathbb{R}^c$ is a vector of which the ith element is one and the others are zero. $\boldsymbol{W} = [\boldsymbol{w}_1 \ \ldots \ \boldsymbol{w}_c] \in \mathbb{R}^{d\times c}$ is a parameter matrix of the model. In this model, an unknown sample $\boldsymbol{x}$ is classified into the class that maximizes $\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle$.

The model that contains a constant term can be realized by pre-mapping $\boldsymbol{x} \mapsto [\boldsymbol{x}^\top \ 1]^\top$. Also higher model can be realized by pre-mapping such as a quadratic model of two dimensional input vector; $[x_1, x_2]^\top \mapsto [x_1, x_2, x_1^2, x_1x_2, x_2^2]^\top$.

3.2. Measurement of empirical error

Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^l \subset \mathbb{R}^d \times \{1, \ldots, c\}$ be a set of labeled training samples. The ith sample is classified correctly if the following inequality is satisfied:

$$\langle \boldsymbol{e}_{y_i}, \boldsymbol{W}^\top \boldsymbol{x}_i \rangle > \max_{j=\{1,\ldots,c\}\setminus\{y_i\}} \langle \boldsymbol{e}_j, \boldsymbol{W}^\top \boldsymbol{x}_i \rangle.$$

This condition has an ambiguity with respect to a scalar multiplier of $\boldsymbol{W}$. In other words, infinitesimal $\boldsymbol{W}$ and huge $\boldsymbol{W}$ might be equivalent, and the complexity cannot be suppressed by regularization. To remove the ambiguity, consider the following function h

$$h(\boldsymbol{x}_i, y_i, \boldsymbol{W}) = \max_{j=\{1,\ldots,c\}\setminus\{y_i\}} \langle \boldsymbol{e}_j - \boldsymbol{e}_{y_i}, \boldsymbol{W}^\top \boldsymbol{x}_i \rangle + \delta, \tag{5}$$

where δ is a positive constant. $h(\boldsymbol{x}_i, y_i, \boldsymbol{W}) \leq 0$ denotes the ith sample classified correctly. The constant δ corresponds to the minimum margin in SVMs. By using this function h, the total empirical error R^{emp} is given by

$$R^{\text{emp}}(\boldsymbol{W}) = \frac{1}{l}\sum_{i=1}^{l} u\Big(h(\boldsymbol{x}_i, y_i, \boldsymbol{W})\Big), \tag{6}$$

where $u(\cdot)$ is a step function:

$$u(x) = \begin{cases} 1 & (x \geq 0) \\ 0 & (x < 0). \end{cases} \tag{7}$$

In learning, optimization procedure requires derivations of objective functions. Hence, a sigmoid function $s(x, a)$ or a robust hinge function $r(x, a)$ [7] is used instead of the step function $u(x)$.

$$s(x, a) = \frac{1}{1 + \exp(-ax)} \tag{8}$$

$$r(x, a) = \begin{cases} 0 & (x < 0) \\ \frac{1}{a}x & (0 \leq x < a) \\ 1 & (x > a). \end{cases} \tag{9}$$

3.3. Measurement of complexity

The complexity of our model is measured by regularization as well as multi-class SVMs:

$$L_1(\boldsymbol{W}) = \|\boldsymbol{W}\|_F^2 = \sum_{i=1}^{c} \|\boldsymbol{w}_i\|^2, \tag{10}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

1. Initialize $\boldsymbol{W}$.
2. Iterate the following steps until $\boldsymbol{W}$ converges
   (a) Calculate the gradient $\frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$.
   (b) Solve the following linear search problem:

$$\hat{\alpha} = \underset{\alpha>0}{\operatorname{argmin}} J(\boldsymbol{W} - \alpha \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}) \tag{12}$$

   (c) Update $\boldsymbol{W}$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \hat{\alpha} \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}} \tag{13}$$

Fig. 1. Learning algorithm using a gradient method

This measurement can be extended to Tikhonov regularization [8] by using a semi-positive definite regularization matrix P:

$$L_2(\boldsymbol{W}) = \langle \boldsymbol{W}, P\boldsymbol{W} \rangle_S = \|P^{1/2}\boldsymbol{W}\|_F^2, \tag{11}$$

where $\langle \cdot, \cdot \rangle_S$ denotes the Hilbert-Schmidt inner product. If P is an identity matrix, $L_2(\boldsymbol{W})$ is equivalent to $L_1(\boldsymbol{W})$.

If input vectors are added the constant term by pre-mapping, multiplier to the constant term may not be loss of a complexity. For example, in the criteria of SVM (2) and (3), the objective $\|\boldsymbol{w}\|^2$ does not contain the constant term b. In this case, $P = \mathrm{diag}([1, \ \ldots \ , 1, 0])$ is used, where $\mathrm{diag}(\boldsymbol{v})$ is a diagonal matrix of which diagonal elements are $\boldsymbol{v}$.

3.4. Learning algorithms

The objective function is defined by linear combination of the empirical error and the model complexity:

$$J(\boldsymbol{W}) = R^{\mathrm{emp}}(\boldsymbol{W}) + \beta L(\boldsymbol{W}), \tag{14}$$

where β is a regularization parameter that controls the balance of the empirical error $R^{\mathrm{emp}}(\boldsymbol{W})$ and the complexity $L(\boldsymbol{W})$, and $L(\boldsymbol{W})$ is acceptable either $L_1(\boldsymbol{W})$ or $L_2(\boldsymbol{W})$. The optimal $\hat{\boldsymbol{W}}$ such that minimizes $J(\boldsymbol{W})$ is expected to minimize generalization errors derived from the SRM theory.

The objective function $J(\boldsymbol{W})$ is minimized by existing optimization techniques such as a gradient method, a conjugate gradient method or a (quasi-)Newton method. Here, we show learning algorithms based on a gradient method and a conjugate gradient method.

Let $\frac{\partial f}{\partial \boldsymbol{W}}$ be a matrix of which (i, j) element is $\frac{\partial f}{\partial \boldsymbol{W}_{ij}}$. For current $\boldsymbol{W}$, let k_i be

$$k_i = \max_{j=\{1,\ldots,c\}\setminus\{y_i\}} \langle \boldsymbol{e}_j, \boldsymbol{W}^\top \boldsymbol{x}_i \rangle, \tag{15}$$

1. Initialize $\boldsymbol{W}$.
2. Iterate the following steps until $\boldsymbol{W}$ converges
   (a) Calculate the gradient $\frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$, and let $\boldsymbol{S} = \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$.
   (b) Solve the following linear search problem:

$$\hat{\alpha} = \underset{\alpha>0}{\operatorname{argmin}} J(\boldsymbol{W} - \alpha \boldsymbol{S}) \tag{19}$$

   (c) Update $\boldsymbol{W}$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \hat{\alpha}\boldsymbol{S} \tag{20}$$

   (d) If the number of iterations is the multiple number of $c \times d$, go to (a), else

$$\boldsymbol{S} \leftarrow \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}} - \lambda \boldsymbol{S}, \tag{21}$$

   where $\lambda = \|\frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}\|_F^2 / \|\boldsymbol{S}\|_F^2$.

Fig. 2. Learning algorithm using the conjugate gradient method

that indicates the maximum class label except y_i for $\boldsymbol{x}_i$. Then we have

$$\frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}} = \frac{\partial R^{\mathrm{emp}}(\boldsymbol{W})}{\partial \boldsymbol{W}} + \beta \frac{\partial L(\boldsymbol{W})}{\partial \boldsymbol{W}}, \tag{16}$$

where

$$\frac{\partial R^{\mathrm{emp}}(\boldsymbol{W})}{\partial \boldsymbol{W}} = \frac{1}{l} \sum_{i=1}^{l} u'(h(\boldsymbol{x}_i, y_i, \boldsymbol{W})) \boldsymbol{x}_i (\boldsymbol{e}_{k_i} - \boldsymbol{e}_{y_i})^\top, \tag{17}$$

and

$$\frac{\partial L(\boldsymbol{W})}{\partial \boldsymbol{W}} = 2P^{1/2}\boldsymbol{W}, \tag{18}$$

where u is replaced by either the sigmoid function (8) or the robust hinge function (9). Consequently, the learning algorithms using gradient and conjugate gradient methods are given as Figs. (1) and (2), respectively. To avoid local minima, the parameter of sigmoid and robust hinge functions a is reduced from a larger value to smaller one during learning.

The objective function $J(\boldsymbol{W})$ includes max() in the function h, its derivation therefore is not continuous. However, it is rare case that samples are on this edge. Even if samples are on the edge, linear search programs (12) and (19) are evaluated by the original objective function $J(\boldsymbol{W})$, and $J(\boldsymbol{W})$ decreases monotonically. The gradient learning algorithm can be extended to adaptive or on-line learning easily. In such case, a fixed small learning coefficient $\epsilon > 0$ is used instead of $\hat{\alpha}$. This algorithm is useful when new labeled samples are added after $\boldsymbol{W}$ are fixed.

Table 1. Classification errors and standard deviations for UCI benchmarks

Problem	Proposed [%]	SVM [%]	p-value[%]	MSVM [%]
Balance-scale	12.97 ± 4.24	12.10 ± 4.22	7.36>	11.06 ± 3.97
Ecoli	13.54 ± 6.64	12.88 ± 6.23	23.24>	12.58 ± 6.11
glass	32.57 ± 9.22	34.90 ± 8.80	3.44<	37.90 ± 9.10
iris	3.93 ± 5.20	4.80 ± 5.37	12.37<	18.40 ±10.2
letter	20.23 ± 0.78	43.23 ± 4.97	0<	24.46 ± 0.78
new-thyroid	4.00 ± 4.48	3.86 ± 4.52	41.13>	5.56 ± 5.56
optdigits	3.09 ± 0.72	3.31 ± 0.70	1.52<	3.29 ± 0.68
pendigits	4.39 ± 0.56	6.49 ± 0.68	0<	5.12 ± 0.57
teaching-ae	44.13 ±12.6	49.40 ±13.5	0.71<	46.20±11.1
wine	1.67 ± 3.01	1.67 ± 2.90	=	3.78 ± 4.24

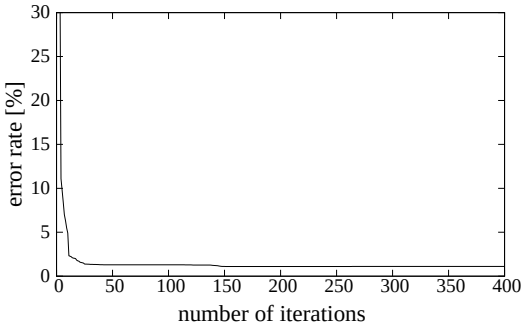

Fig. 3. Training error during learning.

4. EXPERIMENTS

We show experimental results for the UCI benchmark datasets [9] and the handwritten digit dataset USPS [10]. First, accuracies of our classifier for UCI and USPS datasets are shown. Next, the robustness against mislabeled samples of our learning is shown using the USPS dataset. Our classifier is implemented with C using the gradient method with robust hinge function (9) on a standard PC that has Core 2 Quad 2.66GHz CPU, 8GB RAM. We used SVMlight [11] and SVMmulticlass [12] for SVM and *multi-class SVM* (MSVM), respectively.

4.1. UCI benchmarks

We tested our classifier on the several UCI benchmark datasets by comparing with linear SVM and linear MSVM. Table 1 lists classification errors, their standard deviations and p-values of one side t-test. Inequality sings indicate better accuracy rate in our classifier and linear SVM. Mean values of accuracies and standard deviations were estimated from 100 independent outcomes. Parameters were estimated with 10-fold cross validation. As shown in this table, our learning outperformed linear SVM and linear MSVM in many cases. Especially, in the case of the large number of classes such as letters, optdigits and pendigits, one-against-all linear SVM did not perform well. Note again that our learning is formalized as multi-class problems directly, so we do not require some extensions of binary classifiers such as a one-against-all method.

Figure 3 shows an error rate for training samples during learning using the iris dataset. As shown in this figure, the error decreased with the number of iteration increased. This means that our algorithms converge stably.

4.2. Handwritten digits recognition

We tested our classifier on the handwritten digit image dataset USPS [10]. The USPS dataset consists of 7,291 training and 2,007 test images. The size of images is 16×16 pixels. In experiments, intensities of images were directly used as feature vectors. For preprocessing, we normalized $\boldsymbol{x}/\|\boldsymbol{x}\|$ for $\|\boldsymbol{x}\| = 1$. We compared our classifier with *other classifiers*: linear SVM, linear MSVM, Fisher discriminant, and perceptron using multi-regression.

Table 2. Error rates for handwritten digits recognition problem

Method	Parameter	Error rate [%]
Proposed	$\beta = 10^{-7}, \delta = 2$	8.47
Linear SVM	$C = 10$	8.42
Linear Multi-class SVM	$C = 5 \times 10^6$	8.57
Fisher discriminant	—	11.81
Perceptron	—	12.31

Table 2 shows error rates of individual classifiers with their parameter values. Parameters were estimated with 10-fold cross validation. As shown in this table, the generalization ability of our classifier was almost the same as those of SVM and MSVM. In addition, our learning outperformed Fisher discriminant and perceptron.

Finally, we examined the robustness against mislabeled samples of our classifier by randomly replacing training class labels. We randomly separated 7,291 training data and 2,007 test data from whole dataset, then we replace labels of part of training samples randomly. The mean error rates and standard deviations of SVM and our classifier over 100 trials are shown in Table 3 and Figure 4. As shown in these table and figure, our classifier is robust to mislabeled samples more than SVM.

5. DISCUSSION

5.1. Calculation cost

From Eq. (17), calculation cost of our learning is the first order with respect to the number of samples l. It takes about 15 minutes for 1,000 iterations for USPS training data of which size is 7,291, and this is longer than linear SVM. However, as we describe in the next section, our learning admits of improvement regarding calculation cost. In a classification

Table 3. Error rates for handwritten digits recognition problem including label errors

	rate of mislabeled training data [%]			
Method	0	10	20	30
Proposed	5.69 ± 0.50	5.93 ± 0.53	6.35 ± 0.50	6.33 ± 0.49
SVM	5.59 ± 0.47	6.40 ± 0.55	7.30 ± 0.54	8.66 ± 0.61
MSVM	5.35 ± 0.45	6.57 ± 0.57	7.81 ± 0.60	9.00 ± 0.64

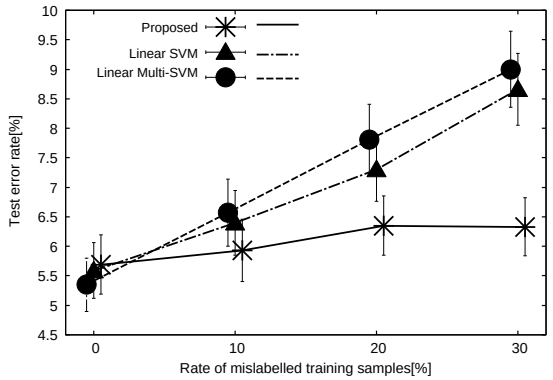

Fig. 4. Error rates for handwritten digits recognition problem including label errors

phase, $d \times c$ times multiplications and search of c elements are required. This is the same as other linear classifiers.

5.2. Further extensions

In the learning process, samples that are far from a margin, in other words, $\boldsymbol{x}_i$ such that $h(\boldsymbol{x}_i, y_i, \boldsymbol{W})$ is very large or very small, are not effective for learning. In the case of making use of the robust hinge function (9), derivation is zero if $h(\boldsymbol{x}_i, y_i, \boldsymbol{W}) < 0$ or $h(\boldsymbol{x}_i, y_i, \boldsymbol{W}) > a$. These samples can be removed from a learning set temporary, and this removing would improve speed of learning. These samples are correspond to a non-support vector set in SVMs.

This model can be easily extended to a multi-labeling case that means one sample belongs to several classes. Suppose that $\boldsymbol{x}_i$ belongs to n_i classes $y_i^1, \ldots, y_i^{n_i}$. Then an empirical error can be measured by

$$
\begin{aligned}
h(\boldsymbol{x}_i, \boldsymbol{y}_i^k, \boldsymbol{W}) &= \max_{j=\{1,\ldots,c\}\setminus\{y_i^1,\ldots,y_i^{n_i}\}} \langle \boldsymbol{e}_j - \boldsymbol{e}_{y_i^k}, \boldsymbol{W}^\top \boldsymbol{x}_i \rangle + \delta \\
R^{\mathrm{emp}} &= \frac{1}{l} \sum_{i=1}^{l} \frac{1}{n_i} \sum_{k=1}^{n_i} u\Big(h(\boldsymbol{x}_i, y_i^k, \boldsymbol{W})\Big).
\end{aligned}
$$

For example, in text genre labeling problems, a multi-labeling model is important.

Kernel methods using non-linear mapping $\Phi(\cdot)$ and a kernel function $k(\cdot,\cdot)$ [13] can be also applied to our classifier. Then $\boldsymbol{W}$ can be expressed by linear combination of mapped samples:

$$
\begin{aligned}
\boldsymbol{w}_i &= \sum_{j=1}^{l} a_{ji} \Phi(\boldsymbol{x}_j) && (22) \\
\boldsymbol{W} &= [\Phi(\boldsymbol{x}_1)\ \Phi(\boldsymbol{x}_2)\ \ldots\ \Phi(\boldsymbol{x}_l)]A, && (23)
\end{aligned}
$$

where $A \in \mathbb{R}^{l \times c}$ is a matrix of which (j, i) element is a_{ji}. Then

$$
\begin{aligned}
\boldsymbol{W}^\top \Phi(\boldsymbol{x}) &= A^\top [\Phi(\boldsymbol{x}_1)\ \Phi(\boldsymbol{x}_2)\ \ldots\ \Phi(\boldsymbol{x}_l)]^\top \Phi(\boldsymbol{x}) \\
&= A^\top \boldsymbol{k}(\boldsymbol{x}), && (24)
\end{aligned}
$$

and

$$
\begin{aligned}
\|\boldsymbol{W}\|_F^2 &= \mathrm{Trace}(A^\top [\Phi(\boldsymbol{x}_1)\ \Phi(\boldsymbol{x}_2)\ \ldots\ \Phi(\boldsymbol{x}_l)]^\top \\
&\quad [\Phi(\boldsymbol{x}_1)\ \Phi(\boldsymbol{x}_2)\ \ldots\ \Phi(\boldsymbol{x}_l)]A) \\
&= \mathrm{Trace}(A^\top K A) = \|K^{1/2} A\|_F^2, && (25)
\end{aligned}
$$

where a vector $\boldsymbol{k}(\boldsymbol{x}) = [k(\boldsymbol{x}_1, \boldsymbol{x}), k(\boldsymbol{x}_2, \boldsymbol{x}), \ldots, k(\boldsymbol{x}_l, \boldsymbol{x})]^\top \in \mathbb{R}^l$ is called empirical kernel map, and a matrix $K \in \mathbb{R}^{l \times l}$ of which (i, j) element is $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is called a kernel gram matrix [13]. Thus by replacing input samples to an empirical kernel vector: $\boldsymbol{x} \mapsto \boldsymbol{k}(\boldsymbol{x})$ and let a regularization matrix P in Eq. (11) be a kernel gram matrix, $P = K$, then the algorithm to obtain $\boldsymbol{W}$ is changed to the algorithm to obtain A directly, and kernel methods can be realized easily.

5.3. Comparison with other classifiers

5.3.1. Support vector machines

As described above, our classifier has many similarities with SVMs. Here, the difference between SVMs and our classifier are discussed. As described in Section 2, SVM is sensitive against outliers and mislabeled training samples because the summation of ξ_i is used for a measurement of an empirical error. On the other hand, our classifier is robust against outliers and mislabeled samples because it employs direct an empirical error measurement.

One of the advantages of SVMs is that optimization problems are convex that guarantees the global minimum, whereas the objective function of our learning is not convex. However, from several simulation, if a in a sigmoid or robust hinge function is annealed from a larger number to a small one, solutions are almost the same even if $\boldsymbol{W}$ is initialized randomly. This means our classifier converged to almost a global minimum.

Sparse solutions are also an advantage of SVMs. The solution of our original learning is not sparse because it is not solved in its dual problem. However, if kernel methods are applied to our classifier, it is expected that $\boldsymbol{W}$ is sparse when the robust-hinge function is used because an empirical error is measured by the first order of $\boldsymbol{W}$ as well as SVM.

Multi-class SVM was proposed for multi-class classification problems [6]. However it is also sensitive against outliers or mislabeled samples because it uses the same measurement

of an empirical error as SVM. Robust SVM (RSVM) [7] was also proposed. It is a binary classifier, and since its optimization problem includes semi-definite constraint, the problem is difficult to solve.

5.3.2. Neural networks

Neural networks may include a sigmoid function in models, whereas our learning includes it in our objective function. Weight decay neural networks are also introduced for regularization [2]. In these methods, an empirical error is usually measured by the squared error between target variables vector $\boldsymbol{t}_i$ and output $\boldsymbol{f}(\boldsymbol{x}_i)$;

$$R^{\text{emp}} = \sum_{i=1}^{l} \|\boldsymbol{t}_i - \boldsymbol{f}(\boldsymbol{x}_i)\|^2. \tag{26}$$

This measurement differs from a true empirical error. Thus it requires too much complexities to achieve smaller empirical errors, and it is expected that the generalization error is low.

6. CONCLUSIONS

In this paper, we proposed a perceptron-based linear classifier that was designed with novel learning algorithms for multi-class classification problems. In a classification phase, an unknown sample is classified into the class that maximizes an inner product of an unknown sample and model parameters that are obtained with gradient-based optimization. We verified classification abilities of our classifier by experiments on benchmark datasets called UCI and USPS. Experimental results showed that our classifier outperformed other classifiers such as SVMs. Furthermore, our classifier was robust to mislabeled samples more than SVM.

In our learning, we defined a criterion about both of empirical errors and complexities for multi-class classifiers directly and minimized the criterion using gradient-based optimization. Due to this direct measurement, our classifier has several advantages over SVMs and other conventional classifiers. For example, our classifier does not require some extensions of binary classifiers such as a one-against-all method because our method is designed as multi-class classification problems directly. In addition, our classifier is robust against outliers or mislabeled samples because our objective function is defined with an empirical error and a model complexity. We plan to apply our classifier to multi-labeling pattern classification and extend our classifier to a kernel one.

7. REFERENCES

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

[2] C. M. Bishop, *Neural networks for pattern recognition*, Oxford university press, 1995.

[3] M. Anthony, "Probabilistic analysis of learning in artificial neural networks: The PAC model and its valiants," *Neural computingsurveys*, vol. 1, pp. 1–47, 1997.

[4] A. J. Smola, B. Shoölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," *Neural Networks*, vol. 11, pp. 637–649, 1998.

[5] S. Raudys, "Evolution and generalization of a single neurone: I. single-layer perceptron as seven statistical classifiers," vol. 11, no. 2, pp. 283–296, 1998.

[6] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *journal of macine learning research*, vol. 2, pp. 265–292, 2001.

[7] L. Xu, K. Crammer, and D. Schuurmans, "Robust support vector machine training via convex outlier ablation," in *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI)*, 2006.

[8] A. N. Tikhonov and V. Y. Arsenin, *Solution of Ill-posed problems*, V. H. Winston and Sons, 1977.

[9] A. Asuncion and D.J. Newman, "UCI machine learning repository," 2007.

[10] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, , and L.D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, pp. 541–551, 1989.

[11] T. Joachims, "SVM^{light} – light support vector machine, `http://svmlight.joachims.org/`," 2004.

[12] T. Joachims, "SVM^{multiclass}, `http://svmlight.joachims.org/svm_multiclass.html`," 2007.

[13] B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A.J. Smola, "Input space vs. feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1000–1017, Sept. 1999.