Fully Data-Driven Optimization of Gaussian Parameters for Kernel Classifier

Kosuke Fukumori and Toshihisa Tanaka
Tokyo University of Agriculture and Technology, Tokyo, Japan
E-mail: fukumori17@sip.tuat.ac.jp, tanakat@cc.tuat.ac.jp Tel/Fax: +81-42-388-7134

***Abstract*—This paper establishes a fully data-driven online estimation method of Gaussian kernel parameters for a kernel logistic regression. The kernel logistic regression is a nonlinear classification model that effectively uses kernel methods, which are one of the techniques to construct effective nonlinear systems with a reproducing kernel Hilbert space (RKHS) induced from a positive semi-definite kernel. Since a performance of the kernel logistic regression with RKHS depends on the kernels to build the model, it is important to select appropriate kernel parameters. In this paper, we propose a method to optimize the precisions (the preciprocal of the variance) at learning for the kernel logistic regression using Gaussian kernels. In addition, the kernel means are also updated to increase the generalization ability. For up to date method of kernel coefficients, we introduce ℓ_1-regularization to supress the number of support vectors. A numerical experiment supports the validity of the proposed method.**

I. Introduction

Machine learning is a technology for classifying or predicting unknown data based on observed data. In particular, supervised learning is a method for estimating labels from unknown data using pairs of an example data and the corresponding label. There are numerous research results on supervised learning. A representative example of machine learning method is the RBF support vector vachine (RBF-SVM) [1] using the kernel method [2]. The potential advantage of kernel method is that a linear method can be directly applied to a nonlnear mapping of an input signal. Thus, innner product on hish-dimensional space to which this mapping belongs to cannnot be calculated explicitly, however, it can be calculated with a kernel function by transferring high-dimensional space to a reproducing kernel Hilbert space (RKHS). It is known that Gaussian kernel, which is one of the representative kernel functions, can express continuous functions with high accuracy [3]. RBF-SVM achieves high estimation performance by expressing ability of nonlinear identification boundary and generalization ability based on geometric margin maximization [4], [5], [6], [7]. However RBF-SVM has a problem that the posterior probability of the class can not be obtained, and it can be used for two-class problems by nature.

On the other hand, a classical method which is widely used as a binary classifier is Logistic Regression (LR) [8]. It has been reported [9] that the kernel logistic regression (KLR) combining LR with a kernel function has the same discrimination performance as SVM. It is thus a powerful and flexible nonlinear classification model [10], [11], [12], [13]. As well as LR, KLR has the advantage of obtaining the posterior probability of the class, and it is easy to extend to multiple classes.

The composition of kernel logistic regression is expressed by the sum of weighted kernels corresponding to feature vectors of the training set. Therefore, since the discrimination performance for the training set is strengthened, they arises a problem of causing over learning. To solve this problem, many KLRs use an ℓ_2-regularization to suppress over-learning [10], [11], [12]. Moreover, using an ℓ_1-regularization, it is possible to construct a more sparse model [14].

In order to improve the estimation ability, selection of kernel parameters is one of the important issues. The Gaussian kernel is a widely used powerful kernel function for the KLR. The parameters of the Gaussian kernel is the kernel precision and mean. When using the Gaussian kernel for the kernel function of KLR, the kernel precision and the kernel mean are parameters. In the conventional KLR, the kernel precision is treated as a hyper parameter, and the kernel mean is identical to a sample in the training data [10]. In the classification problem, the kernel precision is generally determined by grid search. In the context of kernel adaptive regression, a method of updating the kernel precision [15] and a method of updating the kernel mean [16], [17], [18] have been proposed. Furthermore, a method of integrating these methods and simultaneously optimizing both the precision and the mean of the kernel has been proposed [19]. These methods are fully data-driven, and thus, the search in a finite set of points in the grid is no longer necessary. However, this method is applied only to on-line learning in regression models, and application methods to classification models have not been established.

In this paper, we propose a fully data-driven method for learning parameters of the Gaussial kernel in the KLR. In addition, to update the kernel coefficients, we use an ℓ_1-regularization and prevent over-learning by constructing a sparse model. Numerical experiments support the efficacy of the proposed method. For the experiment, we use 18 datasets of binary classification available in UCI Machine Learning Repository [20]. We verify the effectiveness of the proposed kernel optimization method by comparing the classification performance of RBF-SVM and the proposed method.

TABLE I
DEFINITIONS OF SYMBOLS

N	Number of training set
i	Index for element of training set
j	Index for element of support vectors that constitute a model
k	Number of current learning iteration
$\boldsymbol{x}\in\mathbb{R}^m$	m-dimensional feature vector
$y\in\{0,1\}$	True value of the class label to which the feature vector $\boldsymbol{x}$ belongs
$\hat{y}\in\{0,1\}$	Predicted value of the class label to which the feature vector $\boldsymbol{x}$ belongs
$f(\cdot)$	Sum of weighted feature vectors or weighted kernels
h	Kernel coefficient
$\phi(\cdot)$	Activation function
$\mathcal{K}(\cdot,\boldsymbol{x})$	Kernel function

II. Kernel Logistic Regression and Support Vectors

In this section, we describe the kernel logistic regression and a construction method of support vectors using ℓ_1-regularization for describing the proposed method. Table I shows definitions of symbols used for explanation.

A. Kernel logistic regression in RKHS

Kernel logistic regression is a model extended by introducing the kernel method [2], [21] into LR in order to solve nonlinear classification problem. In the construction of LR, the following sigmoid function:

$$\phi(f(\boldsymbol{x}))=\frac{1}{1+\exp(-f(\boldsymbol{x}))} \tag{1}$$

is defined as an activation function, where $f(\boldsymbol{x})$ is expressed by the inner product of the feature vector and the weight vector $\boldsymbol{w}$ as:

$$f(\boldsymbol{x})=\boldsymbol{w}^\top\boldsymbol{x}. \tag{2}$$

Let the $f(\boldsymbol{x})$ be the elements in RKHS when extending LR to KLR. By representer theorem [2], $f(\boldsymbol{x})$ is described as:

$$f(\boldsymbol{x})=\sum_{i=1}^{N}h^{(i)}\mathcal{K}\Big(\boldsymbol{x},\boldsymbol{x}^{(i)}\Big). \tag{3}$$

In KLR, the output of Eq. (1) is regarded as the probability $P(y=0|\boldsymbol{x})$ that the feature vector $\boldsymbol{x}$ is classified to the class label $y=0$. Therefore, the probability $P(y=0|\boldsymbol{x})$ is given by:

$$P(y=0|\boldsymbol{x})=\frac{1}{1+\exp(-f(\boldsymbol{x}))}. \tag{4}$$

On the other hand, the probability $P(y=1|\boldsymbol{x})$ is given by:

$$\begin{aligned}P(y=1|\boldsymbol{x})&=1-P(y=0|\boldsymbol{x})\\&=\frac{\exp(-f(\boldsymbol{x}))}{1+\exp(-f(\boldsymbol{x}))}.\end{aligned} \tag{5}$$

These probabilities are the model outputs of KLR. Thus, KLR is a model that inputs a feature vector and outputs classification probabilities. In this paper, we define the following classification rules to treat KLR as classifier:

$$\hat{y}=\underset{l\in\{0,1\}}{\operatorname{argmax}}P(y=l|\boldsymbol{x}). \tag{6}$$

By using Eq. (6)as a model predicted value of class label, KLR can be applied in the binary classification.

For parameter learning of KLR, cross-entropy which is the log-likelihood of the Bernoulli distribution is adopted as cost function:

$$J=-\frac{1}{N}\sum_{i=1}^{N}\Big[y^{(i)}\log(\phi(f(\boldsymbol{x})))+(1-y^{(i)})\log(1-\phi(f(\boldsymbol{x})))\Big]. \tag{7}$$

The fitting of KLR is equivalent to the minimization problem of the cost function [22]. In this case, the kernel coefficient h is a updated parameter.

B. Construction method of support vectors using ℓ_1-regularization

As can be seen in Eq. (3), the input of the activation function is expressed as the sum of the kernels determined by the feature vectors of all the training data. It means that larger the number N of training set is, the higher the calculation cost is. Therefore, a method of constructing a sparse model by deleting unimportant terms in the sum of the kernels has been proposed. Here we call the kernel functions that construct the model as support vectors and describe construction method using ℓ_1-regularization for the support vectors.

Define the data index set of the support vectors as:

$$\mathcal{J}:=\{j_1,j_2,...,j_r\}\subset\{0,1,...,N-1\}, \tag{8}$$

the support vectors are a set of kernel functions expressed as $\Big\{\mathcal{K}(\cdot,\boldsymbol{x}^{(j)})\Big\}_{j\in\mathcal{J}}$. Therefore, Eq. (3) can be expressed as:

$$f(\boldsymbol{x})=\sum_{j\in\mathcal{J}}h^{(j)}\mathcal{K}\Big(\boldsymbol{x},\boldsymbol{x}^{(j)}\Big). \tag{9}$$

In the [14], an ℓ_1-regularization term is added into the cost function to promote the sparsity. The cost function in this case is given as:

$$J_{\ell_1}=J+\lambda\sum_{j\in\mathcal{J}}\Big|h^{(j)}\Big|, \tag{10}$$

where λ is the regularization parameter. To minimizing this cost function, we can apply the foward-backward splitting method (FOBOS) [14], since Eq. (10) is a convex function. The update rule is then given as:

$$h_{k+1}^{(j)}=\operatorname{sign}\Big(\alpha_k^{(j)}\Big)\Big[\Big|\alpha_k^{(j)}\Big|-\lambda\eta_h\Big]_+, \tag{11}$$

where

$$\alpha_k^{(j)}=h_k^{(j)}-\eta_h\left.\frac{\partial J(h)}{\partial h}\right|_{h=h_k^{(j)}}.$$

Also, η_h is a learning rate for the kernel coefficient h and k is a number of current learning iteration, respectively. If $h_{k+1}^{(j)}=0$, $\mathcal{K}(\cdot,\boldsymbol{x}^{(j)})$ is removed from the set of support vectors.

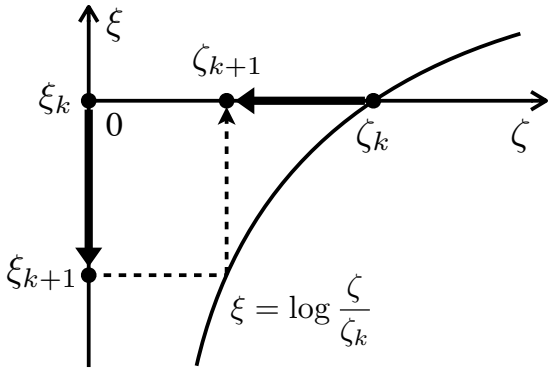


Fig. 1. Update image of ζ in $\mathbb{R}^+$ by variable transformation. Transform ζ_k to ξ_k and update it to ξ_{k+1}. Then inverse transform ξ_{k+1} to ζ_{k+1}.

III. Optimization of kernel parameters for KLR fitting

A widely-used kernel funciton is the Gaussian kernel which is a celebrated example of positive semi-definite kernels. With the Gaussian kernel, a set of support vectors is represented for $j\in J$ as:

$$\mathcal{K}\Big(\cdot,\boldsymbol{x}^{(j)}\Big)=\exp\bigg(-\zeta\Big\|\cdot-\boldsymbol{x}^{(j)}\Big\|^2\bigg), \tag{12}$$

where ζ and $\boldsymbol{x}^{(j)}$ are the parameters called the precision and the mean of the Gaussian kernel function, respectively. In this section, we propose a method to optimize both the precisions and the means at fitting the KLR using the Gaussian kernel. By the proposed method, the precisions and the means are optimized simultaneously when the update of the kernel coefficients to increase the generalization ability. For learning of kernel coefficients, we introduce an ℓ_1-regularization to reduce the number of support vectors.

In the proposed fitting method, the kernel mean is updated. Therefore, the kernel mean represented as Eq. (12) can be different from the feature vector $\boldsymbol{x}$ of the training set. Furthermore, in the proposed method, the kernel precision of the support vector is regarded as a variable. Therefore, in the proposed method, the sum $f(\boldsymbol{x})$ of kernel functions can be expressed as:

$$\begin{aligned} f(\boldsymbol{x})&=\sum_{j\in\mathcal{J}} h^{(j)}\mathcal{K}\Big(\boldsymbol{x},\boldsymbol{x}^{(j)};\zeta_k^{(j)}\Big)\\ &=\sum_{j\in\mathcal{J}} h^{(j)}\exp\bigg(-\zeta_k^{(j)}\Big\|\boldsymbol{x}-\boldsymbol{c}_k^{(j)}\Big\|^2\bigg), \end{aligned} \tag{13}$$

where $\zeta_k^{(j)}$ and $\boldsymbol{c}_k^{(j)}$ are the kernel precision and kernel mean regarded as variables, respectively.

A. Updating the kernel precisions

The kernel precision ζ must be an element of a manifold of the positive real numbers, which is denoted by $\mathbb{R}^+$. When applying the steepest descent (SD) method [23] which is an algorithm for optimization problem directly, the kernel precision is updated in $\mathbb{R}$. In order to avoid this constraint, the proposed method converts to an optimization problem in $\mathbb{R}$ given by:

$$\xi(\zeta)=\log\frac{\zeta}{\zeta_k}, \tag{14}$$

where ζ_k is the current estimate value at the kth update [19]. Fig. 1 shows the updating procedure by the variable transformation. After the transformation, by the SD method, the update rule for ξ is given for $j\in J$ as:

$$\begin{aligned} \xi_{k+1}^{(j)}&=\xi\Big(\zeta_k^{(j)}\Big)-\eta_\xi\left.\frac{\partial J(\zeta)}{\partial\xi}\right|_{h=h_k^{(j)},\boldsymbol{c}=\boldsymbol{c}_k^{(j)},\zeta=\zeta_k^{(j)}}\\ &=\underbrace{\log\left(\frac{\zeta_k^{(j)}}{\zeta_k^{(j)}}\right)}_{=0}-\eta_\xi\left.\frac{\partial J(\zeta)}{\partial\xi}\right|_{h=h_k^{(j)},\boldsymbol{c}=\boldsymbol{c}_k^{(j)},\zeta=\zeta_k^{(j)}}\\ &=-\eta_\xi\zeta_k^{(j)}\left.\frac{\partial J(\zeta)}{\partial\zeta}\right|_{h=h_k^{(j)},\boldsymbol{c}=\boldsymbol{c}_k^{(j)},\zeta=\zeta_k^{(j)}}, \end{aligned} \tag{15}$$

where

$$\begin{aligned} &\left.\frac{\partial J(\zeta)}{\partial\zeta}\right|_{h=h_k^{(j)},\boldsymbol{c}=\boldsymbol{c}_k^{(j)},\zeta=\zeta_k^{(j)}}=\\ &\frac{1}{N}\sum_{i=0}^{N-1}\Big[y^{(i)}-\phi\big(f(\boldsymbol{x}^{(i)})\big)\Big]h_k^{(j)}\mathcal{K}(\boldsymbol{x}^{(i)},\boldsymbol{c}_k^{(j)};\zeta_k^{(j)})\Big\|\boldsymbol{x}^{(i)}-\boldsymbol{c}_k^{(j)}\Big\|^2, \end{aligned} \tag{16}$$

and η_ξ is a learning rate for ξ. By the normalization, $\xi=0$ is the reference point for updating η_ξ. Therefore, ζ can be update stably even if it is small. By the inverse transformation given as $\zeta(\xi)\Big|_{\xi=\xi_{k+1}^{(j)}}=\zeta_{k+1}^{(j)}$, the update rule for ζ is obtained as:

$$\zeta_{k+1}^{(j)}=\zeta_k^{(j)}\exp\Big(\xi_{k+1}^{(j)}\Big). \tag{17}$$

B. Updating the kernel means

The kernel mean $\boldsymbol{c}$ of each support vector is updated to minimize the cost function as Eq. (7). By the SD method, the update rule for $\boldsymbol{c}$ is obtained for $j\in J$ as:

$$\boldsymbol{c}_{k+1}^{(j)}=\boldsymbol{c}_k^{(j)}-\eta_{\boldsymbol{c}}\left.\frac{\partial J(\boldsymbol{c})}{\partial\boldsymbol{c}}\right|_{h=h_k^{(j)},\boldsymbol{c}=\boldsymbol{c}_k^{(j)},\zeta=\zeta_k^{(j)}}, \tag{18}$$

where

$$\begin{aligned} &\left.\frac{\partial J(\boldsymbol{c})}{\partial\boldsymbol{c}}\right|_{h=h_k^{(j)},\boldsymbol{c}=\boldsymbol{c}_k^{(j)},\zeta=\zeta_k^{(j)}}=\\ &-\frac{2}{N}\sum_{i=0}^{N-1}\Big[y^{(i)}-\phi\big(f(\boldsymbol{x}^{(i)})\big)\Big]h_k^{(j)}\zeta_k^{(j)}\mathcal{K}(\boldsymbol{x}^{(i)},\boldsymbol{c}_k^{(j)};\zeta_k^{(j)})(\boldsymbol{x}^{(i)}-\boldsymbol{c}_k^{(j)}), \end{aligned} \tag{19}$$

and $\eta_{\boldsymbol{c}}$ is a learning rate for $\boldsymbol{c}$.

Algorithm 1 Model fitting of the KLR-PM

Input: Training set $\{(\boldsymbol{x}^{(i)},y^{(i)})\}_{i\in\{0,1,\cdots,N-1\}}$

Output: Support vectors $\{\mathcal{K}(\cdot,\boldsymbol{c}^{(j)};\zeta^{(j)})\}_{j\in\mathcal{J}}$ and corresponding coefficients $\{h^{(j)}\}_{j\in\mathcal{J}}$ and

// Set

ℓ_1-regularization parameter λ;

Learning rate for coefficient η_h, for kernel precision after transformation η_ξ and for kernel mean $\eta_{\boldsymbol{c}}$;

Number of maximum learning epoch k_{max};

// Initialize

$\mathcal{J}_0\leftarrow\{0,1,\cdots,N-1\}$;

$\left\{\boldsymbol{c}_0^{(j)}\right\}_{j\in\mathcal{J}_0}\leftarrow\{\boldsymbol{x}^{(i)}\}_{i=\{0,1,\cdots,N-1\}}$;

$\left\{\zeta_0^{(j)}\right\}_{j\in\mathcal{J}_0}\leftarrow\{1,\cdots,1\}$;

$\left\{h_0^{(j)}\right\}_{j\in\mathcal{J}_0}\leftarrow\{1,\cdots,1\}$;

// Learn

$k\leftarrow0$

while $k\neq k_{max}$ **do**

Update $\left\{\zeta_k^{(j)}\right\}_{j\in\mathcal{J}_k}$ to $\left\{\zeta_{k+1}^{(j)}\right\}_{j\in\mathcal{J}_k}$ by (15), (17);

Update $\left\{\boldsymbol{c}_k^{(j)}\right\}_{j\in\mathcal{J}_k}$ to $\left\{\boldsymbol{c}_{k+1}^{(j)}\right\}_{j\in\mathcal{J}_k}$ by (18);

Update $\left\{h_k^{(j)}\right\}_{j\in\mathcal{J}_k}$ to $\left\{h_{k+1}^{(j)}\right\}_{j\in\mathcal{J}_k}$ by (22);

$\mathcal{J}_{k+1}\leftarrow\{\}$;

for $j^*\in\mathcal{J}_k$ **do**

if $h_{k+1}^{(j^*)}=0$ **then**

Remove $\zeta_{k+1}^{(j^*)}$ from $\left\{\zeta_{k+1}^{(j)}\right\}_{j\in\mathcal{J}_k}$;

Remove $\boldsymbol{c}_{k+1}^{(j^*)}$ from $\left\{\boldsymbol{c}_{k+1}^{(j)}\right\}_{j\in\mathcal{J}_k}$;

Remove $h_{k+1}^{(j^*)}$ from $\left\{h_{k+1}^{(j)}\right\}_{j\in\mathcal{J}_k}$;

else

$\mathcal{J}_{k+1}\leftarrow\mathcal{J}_{k+1}\cup\{|\mathcal{J}_{k+1}|\}$;

end if

end for

$k\leftarrow k+1$;

end while

Output Support vectors $\{\mathcal{K}(\cdot,\boldsymbol{c}^{(j)};\zeta^{(j)})\}_{j\in\mathcal{J}_{k-1}}$ and corresponding coefficients $\{h^{(j)}\}_{j\in\mathcal{J}_{k-1}}$

C. Fitting the KLR

Before starting the process, the proposed method constructs a set of support vectors based on the feature vectors of training set for $j\in J_0$ as follows:

$$\mathcal{K}\left(\cdot,\boldsymbol{c}_0^{(j)};\zeta_0^{(j)}\right)=\mathcal{K}\left(\cdot,\boldsymbol{x}^{(j)};\zeta_0^{(j)}\right), \tag{20}$$

where

$$\mathcal{J}_0=\{0,1,...,N-1\}. \tag{21}$$

In the fitting of KLR, we combine the update methods for the parameters in III-A and III-B into the kernel coefficient update method with the ℓ_1-regularization described in II-B. It is possible to reduce the number of support vectors and optimize the kernel parameters. The support vectors and the kernel coefficients are constructed for $j\in J$ as follows:

$$h_{k+1}^{(j)}=\mathrm{sign}\left(\alpha_k^{(j)}\right)\left[\left|\alpha_k^{(j)}\right|-\lambda\eta_h\right]_+, \tag{22}$$

where

$$\begin{aligned}\alpha_k^{(j)}&=h_k^{(j)}+\eta_h\left.\frac{\partial J(h)}{\partial h}\right|_{h=h_k^{(j)},\boldsymbol{c}=\boldsymbol{c}_k^{(j)},\zeta=\zeta_k^{(j)}}\\&=h_k^{(j)}+\eta_h\frac{1}{N}\sum_{i=0}^{N-1}\left[y^{(i)}-\phi\left(f\left(\boldsymbol{x}^{(i)}\right)\right)\right]\mathcal{K}\left(\boldsymbol{x}^{(i)},\boldsymbol{c}_k^{(j)};\zeta_k^{(j)}\right).\end{aligned}$$

When $h_{k+1}^{(j)}\approx0$, remove $\mathcal{K}\left(\cdot,\boldsymbol{c}_{k+1}^{(j)};\zeta_{k+1}^{(j)}\right)$ from the set of support vectors. The KLR which is applied these update methods is named *KLR-PM*. The all of procedures are summarized in Algorithm 1.

TABLE II
THE LIST OF DATASETS FOR THE EXPERIMENT

Dataset	features	samples	ratio of labels
Australian Credit Approval	14	690	383:307
Breast Cancer Wisconsin	9	683	444:239
Climate Model Simulation Crashes	18	540	46:494
Cryotherapy [24], [25]	6	90	42:48
Diabetic Retinopathy Debrecen	19	1151	540:611
German Credit Data	24	1000	700:300
Haberman's Survival	3	306	225:81
Heart	13	270	150:120
Immunotherapy [24], [25]	7	90	19:71
Ionosphere	34	351	225:126
MONK's-1	6	432	216:216
MONK's-2	6	432	290:142
MONK's-3	6	432	204:228
Parkinsons	22	195	48:147
Sonar, Mines vs. Rocks	60	208	97:111
SPECT Heart	22	267	55:212
SPECTF Heart	44	267	55:213
Blood Transfusion Service Center	4	748	570:178

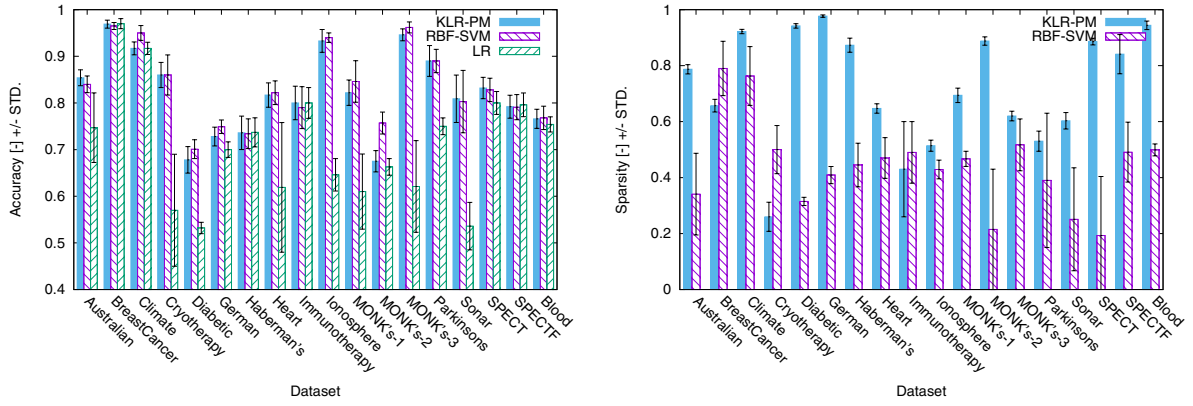
IV. Numerical Experiments

In order to verify the effectiveness of the proposed method, a numerical experiment is presented by using 18 datasets of binary classification published by UCI Machine Learning Repository [20]. Table II shows the datasets used for the experiment. For each dataset, half of the dataset are randomly selected for training and the rest are for test. And each dimension of the feature vector is normalized in the range of $[0,1]$ using the minimum value and the maximum value of the training set.

The models to be compared are the RBF-SVM [1], the LR [8] using an ℓ_2-regularization and the proposed KLR-PM. The ℓ_1-regularization parameter λ of KLR-PM is set as $\lambda=0.005$ in all datasets. On the other hand, the kernel precision γ in the Gaussian kernel, the trade-off parameter C in RBF-SVM and the ℓ_2-regularization parameter in LR are respectively tuned by grid search over the range $\{0.0001,0.001,0.01,0.1,1,10\}$. For adjusting the grid search, the five-fold cross validation

TABLE III
ACCURACIES AND SPARSITIES OF (MEAN ± STD.) OF EACH COMPARED MODEL. THE BEST ACCURACIES AND SPARSITIES ARE HIGHLIGHTED.

Dataset	Accuracy ± STD.			Sparsity ± STD.	
	KLR-PM	RBF-SVM	LR	KLR-PM	RBF-SVM
Australian Credit Approval	$\mathbf{0.854 \pm 0.0170}$	0.840 ± 0.0179	0.747 ± 0.0746	$\mathbf{0.787 \pm 0.0165}$	0.341 ± 0.146
Breast Cancer Wisconsin	$\mathbf{0.969 \pm 0.00859}$	0.965 ± 0.00745	0.970 ± 0.0108	$\mathbf{0.657 \pm 0.0225}$	0.790 ± 0.0972
Climate Model Simulation Crashes	$\mathbf{0.917 \pm 0.0138}$	0.950 ± 0.0159	$\mathbf{0.917 \pm 0.0131}$	$\mathbf{0.922 \pm 0.00777}$	0.763 ± 0.105
Cryotherapy	$\mathbf{0.86 \pm 0.027}$	$\mathbf{0.86 \pm 0.043}$	0.57 ± 0.12	0.26 ± 0.052	$\mathbf{0.50 \pm 0.086}$
Diabetic Retinopathy Debrecen	0.678 ± 0.0284	$\mathbf{0.701 \pm 0.0202}$	0.532 ± 0.0122	$\mathbf{0.942 \pm 0.00798}$	0.315 ± 0.0147
German Credit Data	0.728 ± 0.0202	$\mathbf{0.749 \pm 0.0145}$	0.700 ± 0.0166	$\mathbf{0.977 \pm 0.0044}$	0.409 ± 0.0306
Haberman's Survival	0.736 ± 0.0359	0.734 ± 0.0319	$\mathbf{0.737 \pm 0.0312}$	$\mathbf{0.873 \pm 0.0252}$	0.445 ± 0.078
Heart	0.817 ± 0.0261	$\mathbf{0.822 \pm 0.0254}$	0.619 ± 0.139	$\mathbf{0.647 \pm 0.0169}$	0.470 ± 0.0727
Immunotherapy	0.80 ± 0.036	$\mathbf{0.79 \pm 0.045}$	0.80 ± 0.033	0.43 ± 0.17	$\mathbf{0.49 \pm 0.11}$
Ionosphere	$\mathbf{0.933 \pm 0.0245}$	0.940 ± 0.0102	0.646 ± 0.0347	$\mathbf{0.514 \pm 0.0198}$	0.429 ± 0.0334
MONK's-1	0.822 ± 0.0272	$\mathbf{0.846 \pm 0.0445}$	0.610 ± 0.0805	$\mathbf{0.694 \pm 0.0258}$	0.467 ± 0.027
MONK's-2	0.675 ± 0.0228	$\mathbf{0.757 \pm 0.0235}$	0.663 ± 0.0178	$\mathbf{0.888 \pm 0.015}$	0.215 ± 0.215
MONK's-3	0.946 ± 0.0128	$\mathbf{0.962 \pm 0.0117}$	0.621 ± 0.0982	0.620 ± 0.0173	$\mathbf{0.517 \pm 0.0927}$
Parkinsons	$\mathbf{0.89 \pm 0.033}$	$\mathbf{0.89 \pm 0.025}$	0.75 ± 0.018	$\mathbf{0.53 \pm 0.036}$	0.39 ± 0.24
Sonar, Mines vs. Rocks	$\mathbf{0.809 \pm 0.0508}$	0.803 ± 0.0669	0.536 ± 0.0509	$\mathbf{0.603 \pm 0.0295}$	0.251 ± 0.184
SPECT Heart	$\mathbf{0.832 \pm 0.0229}$	0.828 ± 0.0251	0.800 ± 0.0245	$\mathbf{0.886 \pm 0.0120}$	0.193 ± 0.211
SPECTF Heart	0.792 ± 0.0246	0.791 ± 0.0269	$\mathbf{0.796 \pm 0.0254}$	$\mathbf{0.841 \pm 0.0699}$	0.491 ± 0.107
Blood Transfusion Service Center	0.766 ± 0.0205	$\mathbf{0.768 \pm 0.0251}$	0.754 ± 0.0162	$\mathbf{0.944 \pm 0.0153}$	0.499 ± 0.0214



(a) Accuracies (b) Sparsities

Fig. 2. Accuracies and sparsities shown in Table III are displayed as bar graphs. And STDs are indicated by error bars.

with two subsets is used. One subset is used for validation and the remaining subset is used for training For the evaluation, a mean accuracy and a mean sparsity by taking an average over 10 independent realizations are adopted. The accuracy and the sparsity are calculated by:

$$\text{Accuracy} = 1 - \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} y^{(i)} \oplus \hat{y}^{(i)}, \tag{23}$$

$$\text{Sparsity} = 1 - \frac{n_{\text{sup}}}{N}, \tag{24}$$

where $\oplus$ is the operator that describes the exclusive-OR (XOR). Also, N_{test} and n_{sup} are a number of test set and a number of support vectors, respectively.

Table III and Fig. 2 show the results of experiment. It can be seen in Table III and Fig. 2(a) that the accuracies of the KLR-PM achieved almost comparable accuracies to the RBF-SVM, which had parameter tuning using grid search. However, it should be emphasized that the sparsities of the KLR-PM are mostly higher than those of the ohter method in most datasets, as confirmed in Table III and Fig. 2(b). Therefore, KLR-PM can construct a classifier which has generalization performance as high as RBF-SVM with a small number of support vectors.

V. CONCLUSION

We proposed a new kernel optimization method for Kernel logistic regression. Our proposal method updated not only the kernel coefficients, but also the kernel precisions and the kernel means from training set. By using the ℓ_1-regularization for update of the kernel coefficients, it is possible to constitute a sparse model. The numerical experiment for various datasets demonstrated the effectiveness of the proposed method.

ACKNOWLEDGMENT

We would like to thank Dr. Muhammad Sharif Uddin for many fruitful discussions. This work is supported by JSPS KAKENHI grant number 17H01760.

REFERENCES

[1] C.J. Burges, "A tutorial on support vector machines for pattern recognition," Data Mining and Knowledge Discovery, vol.2, no.2, pp.121–167, 1998.

[2] N. Aronszajn, "Theory of reproducing kernels," Trans. Amer. Math. Soc., vol.68, no.9, pp.337–404, 1950.

[3] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," J. Mach. Learn. Res., vol.2, pp.67–93, 2002.

[4] V. Vapnik, The Nature of Statistical Learning Theory, Information Science and Statistics, Springer-Verlag New York, 1999.

[5] V.S. Cherkassky and F.M. Mulier, Learning from Data, Adaptive and Learning Systems for Signal Processing, Communications and Control, Wiley, 1998. Concepts, Theory, and Methods.

[6] P. Bradley and O. Mangasarian, "Massive data discrimination via linear support vector machines," Optimization Methods and Software, vol.13, no.1, pp.1–10, 2000.

[7] O.L. Mangasarian, et al., "Generalized support vector machines," Advances in Neural Information Processing Systems, pp.135–146, 1999.

[8] C.M. Bishop, Pattern Recognition and Machine Learning, New York, NY: Springer, 2006.

[9] J. Zhu and T. Hastie, "Kernel logistic regression and the import vector machine," Journal of Computational and Graphical Statistics, vol.14, no.1, pp.185–205, 2005.

[10] F. Liu, X. Huang, and J. Yang, "Indefinite kernel logistic regression," Proceedings of the 2017 ACM on Multimedia ConferenceACM 2017. 846–853.

[11] T.S. Jaakkola and D. Haussler, "Probabilistic kernel regression models," Proceedings of the 1999 Conference on AI and Statistics, 1999.

[12] S.S. Keerthi, K. Duan, S.K. Shevade, and A.N. Poo, "A fast dual algorithm for kernel logistic regression," Machine Learning, vol.61, no.1, pp.151–165, 2005.

[13] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001.

[14] J. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," Journal of Machine Learning Research, vol.10, no.Dec, pp.2899–2934, 2009.

[15] H. Fan, Q. Song, and S.B. Shrestha, "Kernel online learning with adaptive kernel width," Neurocomputing, vol.175, pp.233–242, 2016.

[16] C. Saide, R. Lengelle, P. Honeine, C. Richard, and R. Achkar, "Dictionary adaptation for online prediction of time series data with kernels," Proc. of 2012 IEEE Statistical Signal Processing Workshop (SSP), 2012. 604-607.

[17] C. Saide, R. Lengelle, P. Honeine, and R. Achkar, "Online kernel adaptive algorithms with dictionary adaptation for MIMO models," IEEE Signal Process. Lett., vol.20, no.5, pp.535–538, 2013.

[18] T. Ishida and T. Tanaka, "Efficient construction of dictionaries for kernel adaptive filtering in a dynamic environment," Proc. of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015), 2015. 3536–3540.

[19] T. Wada and T. Tanaka, "Dictionary learning for Gaussian kernel adaptive filtering with variable kernel center and width," Proc. of 2018 International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018), accepted.

[20] C. Blake and C. Merz, "UCI repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science," 1998.

[21] B. Pan, W.-S. Chen, B. Chen, C. Xu, and J. Lai, "Out-of-sample extensions for non-parametric kernel methods," IEEE Transactions on neural networks and learning systems, vol.28, no.2, pp.334–345, 2017.

[22] G.C. Cawley and N.L. Talbot, "Efficient model selection for kernel logistic regression," Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol.2, 2004. 439–442.

[23] S. Haykin, Adaptive Filter Theory, Upper Saddle River, NJ: Prentice-Hall, 2002.

[24] F. Khozeimeh, R. Alizadehsani, M. Roshanzamir, A. Khosravi, P. Layegh, and S. Nahavandi, "An expert system for selecting wart treatment method," Comput. Biol. Med., vol.81, no.C, pp.167–175, Feb. 2017. https://doi.org/10.1016/j.compbiomed.2017.01.001

[25] F. Khozeimeh, F. Jabbari Azad, Y. Mahboubi Oskouei, M. Jafari, S. Tehranian, R. Alizadehsani, and P. Layegh, "Intralesional immunotherapy compared to cryotherapy in the treatment of warts," International Journal of Dermatology, vol.56, no.4, pp.474–478, 2017.