Doubly Sparse Bayesian Kernel Logistic Regression

Atsushi Kojima and Toshihisa Tanaka
Department of Electrical and Electronic Engineering
Tokyo University of Agriculture and Technology, Tokyo, Japan
E-mail: kojima15@sip.tuat.ac.jp, tanakat@cc.tuat.ac.jp
Tel/Fax: +81-42-388-7123

***Abstract*—When input patterns have redundant features in regression analysis or pattern recognition, the prediction accuracy is likely to be lowered. For a kernel regression in a reproducing kernel Hilbert space, as the number of observed input patterns increases, the dimension of parameters increases since a kernel regression model using the kernel method is represented by the linear sum of kernel functions corresponding to input patterns. This can yield overfitting. This paper proposes a method for simultaneously selecting features and model coefficients. To express a sparsity of the features and the weight coefficients, we generate binary vectors, where all the elements are 0 or 1 sampled from the beta process. The proposed method can select effective features and estimate sparse weight coefficients by introducing the binary vectors into the kernel regression model. Numerical examples support the efficacy of the proposed method.**

I. Introduction

Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a couple of an input object and the desired output value. The supervised learning analyzes the training data and produces an inferred function, which can be used for mapping new examples. We use a technique called regression analysis. Regression analysis is a statistical technique for investigating and modeling the relationship between two variables. This technology can extract hidden patterns from large amounts of data and classify and predict unknown data.

The goal is to learn input-output mapping function based on the set of N training examples, $\{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(N)}, y^{(N)})\}$, where $x^{(i)}$ is the feature vector of the i-th example and $y^{(i)}$ is its label. We employ the square error or the cross-entropy as the error function depending on the problem to be solved. In general, in the case of the regression problem, we use a square error and regard the linear regression model as a regression model. Also, in the case of the classification problem, we employ the cross-entropy and regard the logistic regression [1] as the regression model. The logistic regression model is a well known two-value classification method in the field of machine learning. This model predicts the probability of the categorical dependent variables.

Many situations require nonlinear regression models since systems in the real environments can be modeled nonlinear. One of the nonlinear regression models is a regression model using a kernel method [2] in a reproducing kernel Hilbert space (RKHS). It is possible to construct the nonlinear regression model effectively by the RKHS space and positive definite kernel by using the kernel method. Applying the kernel method to the logistic regression as done for support vector machine (SVM) [3], [4], a robust non-linear version of the logistic regression is obtained called kernel logistic regression (KLR) [5].

When the observed patterns have redundant feature values or irrelevant feature values to object variable, the prediction accuracy of the model is reduced. Hence, it is necessary to select useful feature for improving the prediction accuracy [6]. Traditional feature selection methods can be classified into three types; filter method, wrapper method, and embedded method by the selection criteria [6]. The filter method selects features as a pre-treatment independently of the learning the classifier. Typical techniques include Fisher Score [7], Laplacian Score [8], a technique based on correlation of each feature and label [9], and a technique based on mutual information of each feature and label are proposed [10]. These techniques computational cost are low, but the accuracies of the models are low since they dont require the constructing of learning models.

Another implementation of KRL is the model selection. KLR algorithms have linearly growing structures with number of training samples since their regression models are represented by the linear sum of kernel functions corresponding to input patterns. This poses both computational issues. Hence, it is necessary to select the appropriate model for the kernel regression model in RKHS.

One model selection method is to use an ℓ_0-norm or an ℓ_1-norm [11]–[19]. Another is to use Bayesian inference to obtain sparse parameter [20]. The effectiveness of these techniques have been confirmed. In terms of Bayesian inference, the regularization using the ℓ_1-norm assumes the Laplace distribution to the prior distribution of parameters. In the Laplace distribution, parameters of the model are closer to 0, and the higher probability is assigned. However, according to the definition of sparsity, it is necessary to assume a probability distribution with 0 or non-zero binary variable. Since the model selection using Bayesian estimation is dependent on the prior distribution of setting parameters, the model selection by the regularization using the ℓ_1-norm cannot be assumed the process of generating the appropriate model. Therefore, in the sparse parameter estimation, a nonparametric Bayesian method using beta process has been proposed. In recent studies, it

was reported that we can get sparse parameters in a factor analysis [21], sparse linear regression [22], and sparse kernel regression [23].

This paper proposes a dual sparsification method, which is based on sampling, for regression coefficients and feature values of the input pattern in the KLR model in RKHS. In the proposed method, to express a sparsity of the features and the weight coefficients, we generate binary vectors, where all the elements are 0 or 1 sampled from the beta process. We can estimate the binary vectors by deriving an estimation algorithm of the posterior distribution using Gibbs sampling [24]. Numerical examples support the efficacy of our proposed method.

II. Kernel Logistic Regression in RKHS

It is a well known two-value classification method in the field of statistical learning. The logistic regression predicts the probability of the categorical dependent variables. Let $\mathcal{X} \subset \mathbb{R}^D$ and $x \in \mathcal{X}$ denote the input space and the input pattern. Given the set of N training data $\left\{\boldsymbol{x}^{(i)}\right\}_{i=0}^{N-1}$ and the corresponding label $\left\{t^{(i)}\right\}_{i=0}^{N-1}$, $t^{(i)} \in \{0, 1\}$. Consider the problem of solving the discriminant function $f(\boldsymbol{x})$ to give $y = \sigma(f(\boldsymbol{x}))$, where $\left\{y^{(i)}\right\}_{i=0}^{N-1}$ is outputs of the logistic regression model and $\sigma(\cdot)$ is a sigmoid function. The logistic regression model is described as

$$f(\boldsymbol{x}) = \sum_{j=0}^{D-1} w_j x_j, \tag{1}$$

$$y = \sigma(f(\boldsymbol{x})) = \frac{1}{1 + \exp\left(-f(\boldsymbol{x})\right)}, \tag{2}$$

where $w_j \in \mathbb{R}$ is the regression parameter. Let the regression function f be the elements in RKHS $\mathcal{H}$. By representer theorem [2], $f(\boldsymbol{x})$ is described as

$$f(\boldsymbol{x}) = \sum_{j=0}^{N-1} \alpha^{(j)} k(\boldsymbol{x}^{(j)}, \boldsymbol{x}), \tag{3}$$

where $k(\cdot, \cdot) \in \mathcal{H}$ is a reproducing kernel and meets reproductive property $f(\boldsymbol{\chi}) = \langle f(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{\chi})\rangle$ and $\boldsymbol{\alpha} \in \mathbb{R}^N$ is the coefficient vector.

In the kernel logistic regression, a single output variable $t^{(i)}$ follows a Bernoulli probability function that tales on the value 1 with probability $\sigma(f(\boldsymbol{x}^{(i)}))$ and 0 with probability $1 - \sigma(f(\boldsymbol{x}^{(i)}))$. We use maximum likelihood to determine the parameters of the logistic regression model. The likelihood function with respect to $\boldsymbol{y}$ is described as

$$p(\boldsymbol{t}|\boldsymbol{y}) \sim \prod_{i=0}^{N-1} \text{Bernoulli}\left(t^{(i)}|y^{(i)}\right) \tag{4}$$

$$= \prod_{i=0}^{N-1} \left\{y^{(i)}\right\}^{t^{(i)}} \left\{1 - y^{(i)}\right\}^{1-t^{(i)}}. \tag{5}$$

The optimal model parameters are found by minimizing the error function representing the negative log-likelihood of the data. The error function is given by cross-entropy as

$$E(\boldsymbol{\alpha}) = -\ln p(\boldsymbol{t}|\boldsymbol{y}) = -\sum_{i=0}^{N-1} \left\{t^{(i)} \ln y^{(i)} + (1 - t^{(i)}) \ln (1 - y^{(i)})\right\}. \tag{6}$$

III. Sparse Kernel Logistic Regression Based on Graphical Model

We formulate the problem of finding sparse input feature x_k and the coefficient vector $\boldsymbol{\alpha}$. To solve the problem, we propose a method for feature selection and model selection in sparse kernel regression model.

A. Formulation of Sparse Kernel Logistic Regression in RKHS

In [25], the binary vectors $\boldsymbol{\zeta}$ and $\boldsymbol{z}$, where all elements are either 0 or 1, are introduced to promote the sparsity of the KLR model in RKHS. The model is given as

$$f(\boldsymbol{x}) = \sum_{j=0}^{N-1} (\alpha^{(j)} \odot z^{(j)}) k(\boldsymbol{x} \odot \boldsymbol{\zeta}, \boldsymbol{x}^{(j)} \odot \boldsymbol{\zeta}), \tag{7}$$

$$y = \sigma(f(\boldsymbol{x})) = \frac{1}{1 + \exp\left(-f(\boldsymbol{x})\right)}. \tag{8}$$

B. Generative Model

Assume that the parameters in (7) are generated from the beta process model [23] in sparse KLR:

$$\boldsymbol{y}|\boldsymbol{t}, \boldsymbol{\zeta} \sim \prod_{i=0}^{N-1} \text{Bernoulli}\left(y^{(i)}|t^{(i)}, \boldsymbol{\zeta}\right), \tag{9}$$

$$\boldsymbol{y}|\boldsymbol{t}, \boldsymbol{z} \sim \prod_{i=0}^{N-1} \text{Bernoulli}\left(y^{(i)}|t^{(i)}, z^{(i)}\right), \tag{10}$$

$$\boldsymbol{\alpha} \sim N(\boldsymbol{0}, \xi^{-1} \boldsymbol{I}_N), \tag{11}$$

$$\zeta_k|\eta_k \sim \text{Bernoulli}(\eta_k), \tag{12}$$

$$\eta_k \sim \text{Beta}(\tau, \upsilon), , \tag{13}$$

$$z^{(i)}|q^{(i)} \sim \text{Bernoulli}(q^{(i)}), \tag{14}$$

$$q^{(i)} \sim \text{Beta}(\beta, \gamma), \tag{15}$$

where $\boldsymbol{I}_N$ is the identity matrix of size N. The graphical model for the above generative model is illustrated in Fig. 1. In the above equations, ζ_k denotes the Bernoulli distribution with hyperparameter η_k and $z^{(i)}$ denotes the Bernoulli distribution with hyperparameter $q^{(i)}$, where η_k and $q^{(i)}$ have the beta distribution that is the conjugate prior of Bernoulli distribution. Besides $\boldsymbol{\alpha}$ is the Gaussian distribution with average $\boldsymbol{0}$ and covariance matrix $\xi^{-1}\boldsymbol{I}_N$, where ξ is a hyperparameter for the variance of $\boldsymbol{\alpha}$.

IV. Inference

When we observe $\boldsymbol{y}$, the posterior with respect to $\boldsymbol{\zeta}$ is given by,

$$p(\boldsymbol{\zeta}|\boldsymbol{t}, \boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{t}, \boldsymbol{\zeta}) p(\boldsymbol{\zeta}). \tag{16}$$

We find $\boldsymbol{\zeta}$ by maximizing this posterior distribution. Then, the posterior with respect to $\boldsymbol{z}$ is also given by

$$p(\boldsymbol{z}|\boldsymbol{t}, \boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{t}, \boldsymbol{z}) p(\boldsymbol{z}). \tag{17}$$

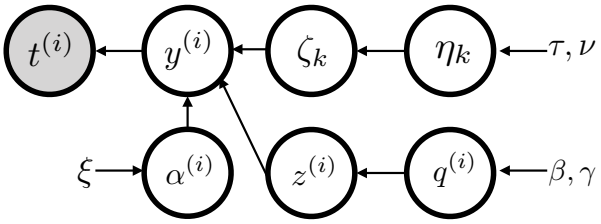


Fig. 1: Graphical model for observed patterns.

We find $\boldsymbol{z}$ by maximizing this posterior distribution. Finally, we estimate $\boldsymbol{\alpha}$ from the posterior distribution:

$$p(\boldsymbol{\alpha}|\boldsymbol{t}, \boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{t}, \boldsymbol{z})p(\boldsymbol{z}). \tag{18}$$

In the following, these parameters are derived.

A. Prior Distribution for Binary Vectors

Prior distributions of $\boldsymbol{\zeta}$ and $\boldsymbol{z}$ are derived from (12)–(15). When we choose $\tau = \frac{a_\zeta}{D}, \upsilon = 1$ in (13), we get corresponding prior distribution $p(\boldsymbol{\zeta})$ by marginalizing $p(\boldsymbol{\zeta}, \boldsymbol{\eta})$ using (12) and (13) which is described as

$$\begin{aligned} p(\boldsymbol{\zeta}) &= \int p(\boldsymbol{\zeta}, \boldsymbol{\eta}) d\boldsymbol{\eta} \\ &= \int p(\boldsymbol{\zeta}|\boldsymbol{\eta}) p(\boldsymbol{\eta}) d\boldsymbol{\eta} \\ &= \int \prod_{k=0}^{D-1} p(\zeta_k|\eta_k) p(\eta_k) d\eta_k \\ &= \frac{a_\zeta}{D} \cdot \frac{\Gamma(\sum_{k=0}^{D-1} \zeta_k + a_\zeta)\Gamma(D - \sum_{k=0}^{D-1} \zeta_k + 1)}{\Gamma(D + a_\zeta + 1)}. \end{aligned} \tag{19}$$

Conversely, when we choose $\beta = \frac{a_z}{N}, \gamma = 1$ in (15), we get corresponding prior distribution $p(\boldsymbol{z})$ by marginalizing $p(\boldsymbol{z}, \boldsymbol{q})$ using (12) and (13) which is described as

$$\begin{aligned} p(\boldsymbol{z}) &= \int p(\boldsymbol{z}, \boldsymbol{q}) d\boldsymbol{q} \\ &= \int p(\boldsymbol{z}|\boldsymbol{q}) p(\boldsymbol{q}) d\boldsymbol{q} \\ &= \int \prod_{i=0}^{N-1} p(z^{(i)}|q^{(i)}) p(q^{(i)}) dq^{(i)} \\ &= \frac{a_z}{N} \cdot \frac{\Gamma(\sum_{i=0}^{N-1} z^{(i)} + a_z)\Gamma(N - \sum_{i=0}^{N-1} z^{(i)} + 1)}{\Gamma(N + a_z + 1)}. \end{aligned} \tag{20}$$

As $p(\boldsymbol{\zeta})$ and $p(\boldsymbol{z})$ do not change even if replacing 1 and 0 or $\boldsymbol{\zeta}$ and $\boldsymbol{z}$ themselves with each other in the $\boldsymbol{\zeta}$ and $\boldsymbol{z}$, it is necessary to consider $\boldsymbol{\zeta}$ and $\boldsymbol{z}$ which are equivalent even if interchanging elements. D_0 and D_1 are the numbers of zero and one included in $\boldsymbol{\zeta}$, respectively. N_0 and N_1 are the numbers of zero and one included in $\boldsymbol{z}$. Since the number of combinations of the elements in $\boldsymbol{\zeta}$ and $\boldsymbol{z}$ are $\frac{D!}{D_0!D_1!}$ and $\frac{N!}{N_0!N_1!}$, respectively, the distribution of $\boldsymbol{\zeta}$ and $\boldsymbol{z}$ are equivalent to the distributions of D_1 and N_1:

$$\begin{aligned} p(\boldsymbol{\zeta}) &= p(D_1) \\ &= \frac{D!}{D_0!D_1!} a_\zeta \frac{\Gamma(D_1 + a_\zeta)\Gamma(D - D_1 + 1)}{\Gamma(D + a_\zeta + 1)}, \end{aligned} \tag{21}$$

$$\begin{aligned} p(\boldsymbol{z}) &= p(N_1) \\ &= \frac{N!}{N_0!N_1!} a_z \frac{\Gamma(N_1 + a_z)\Gamma(N - N_1 + 1)}{\Gamma(N + a_z + 1)}, \end{aligned} \tag{22}$$

where a_ζ and a_z are parameters of the Poisson distribution.

B. Likelihood Function for Observed Pattern

The likelihood function to select feature for the model is given as

$$\begin{aligned} &p(\boldsymbol{y}|\boldsymbol{t}, \boldsymbol{\zeta}) \\ &\sim \prod_{i=0}^{N-1} \text{Bernoulli}\left(y^{(i)}|t^{(i)}, \boldsymbol{\zeta}\right) \\ &= \prod_{i=0}^{N-1} \left(\frac{1}{1 + \exp\left(-f_{fe}(\boldsymbol{x}^{(i)})\right)}\right)^{t^{(i)}} \left(1 - \left(\frac{1}{1 + \exp\left(-f_{fs}(\boldsymbol{x}^{(i)})\right)}\right)\right)^{1-t^{(i)}}, \end{aligned} \tag{23}$$

where $f_{fs}(\boldsymbol{x}^{(i)})$ is the discriminant function for feature selection (fs) by given as

$$f_{fs}(\boldsymbol{x}^{(i)}) = \sum_{j=0}^{N-1} \alpha^{(j)} k(\boldsymbol{x}^{(i)} \odot \boldsymbol{\zeta}, \boldsymbol{x}^{(j)} \odot \boldsymbol{\zeta}). \tag{24}$$

Then, the likelihood function to select sparse logistic regression coefficients is given as

$$\begin{aligned} &p(\boldsymbol{y}|\boldsymbol{t}, \boldsymbol{z}) \\ &\sim \prod_{i=0}^{N-1} \text{Bernoulli}\left(y^{(i)}|t^{(i)}, z^{(i)}\right) \\ &= \prod_{i=0}^{N-1} \left(\frac{1}{1 + \exp\left(-f_{ms}(\tilde{\boldsymbol{x}}^{(i)})\right)}\right)^{t^{(i)}} \left(1 - \left(\frac{1}{1 + \exp\left(-f_{ms}(\tilde{\boldsymbol{x}}^{(i)})\right)}\right)\right)^{1-t^{(i)}}, \end{aligned} \tag{25}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x} \odot \boldsymbol{\zeta}$ is input after feature selection represented as

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} \odot \boldsymbol{\zeta}, \tag{26}$$

and $f_{fs}(\tilde{\boldsymbol{x}}^{(i)})$ is the discriminant function for model selection (ms) by given as

$$f_{ms}(\tilde{\boldsymbol{x}}^{(i)}) = \sum_{j=0}^{N-1} (\alpha^{(j)} \odot z^{(j)}) k(\tilde{\boldsymbol{x}}^{(i)}, \tilde{\boldsymbol{x}}^{(j)}). \tag{27}$$

C. Maximization of Posterior Distribution by MCMC

We maximize (16) with respect to $\boldsymbol{\zeta}$ and (17) with respect to $\boldsymbol{z}$. Since it is difficult to directly sample from posterior $p(\boldsymbol{\zeta}|\boldsymbol{t}, \boldsymbol{y})$ and $p(\boldsymbol{z}|\boldsymbol{t}, \boldsymbol{y})$, we find the posterior of ζ_k and $z^{(i)}$ from $p(\zeta_k = 1|\boldsymbol{t}, \boldsymbol{\zeta}_{-k}, \boldsymbol{y})$ and $p(z^{(i)} = 1|\boldsymbol{z}^{(-i)}, \boldsymbol{t}, \boldsymbol{y})$ by Gibbs sampling (GS) [24] which is one of The Markov Chain Monte Carlo (MCMC) algorithm. We show an algorithm for proposed method to Algorithm 1.

Here, $\boldsymbol{\zeta}_{-k}$ is $\boldsymbol{\zeta}_{-k} = \{\zeta_0, \ldots, \zeta_{D-1}\} - \{\zeta_k\}$. Then, from Bayes' theorem, $p(\zeta_k = 1|\boldsymbol{t}, \boldsymbol{\zeta}_{-k}, \boldsymbol{y})$ is given as

$$p(\zeta_k = 1|\boldsymbol{\zeta}_{-k}, \boldsymbol{t}, \boldsymbol{y}) \propto p(\zeta_k = 1|\boldsymbol{\zeta}_{-k}) p(\boldsymbol{t}|\zeta_k = 1, \boldsymbol{\zeta}_{-k}, \boldsymbol{y}). \quad (28)$$

Similarly, $\boldsymbol{z}^{(-i)}$ is $\boldsymbol{z}^{(-i)} = \left\{z^0, \ldots, z^{N-1}\right\} - \left\{z^{(i)}\right\}$. Then, from Bayes' theorem, $p(z^{(i)} = 1|\boldsymbol{z}^{(-i)}, \boldsymbol{t}, \boldsymbol{y})$ is given as

$$p(z^{(i)} = 1|\boldsymbol{z}^{(-i)}, \boldsymbol{t}, \boldsymbol{y}) \propto p(z^{(i)} = 1|\boldsymbol{z}^{(-i)}) p(\boldsymbol{t}|z^{(i)} = 1, \boldsymbol{z}^{(-i)}, \boldsymbol{y}). \quad (29)$$

Then, the posterior of $\boldsymbol{\alpha}$ is given as

$$p(\boldsymbol{\alpha}|\boldsymbol{t}, \boldsymbol{y}) \propto p(\boldsymbol{t}|\boldsymbol{\alpha}, \boldsymbol{y}) p(\boldsymbol{\alpha}). \quad (30)$$

Since it is also difficult to directry sample from posterior, we estimate the parameters of the KLR by using a variatoinal Bayesian (VB) inference [20]. In [20], we can obtain the corresponding variational approximation $q(\boldsymbol{\alpha})$ to the posterior of regression parameters $\boldsymbol{\alpha}$ using EM algorithm, giving a Gaussian variational posterior of the form:

$$q(\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\alpha}|m_N, S_N). \quad (31)$$

E-step:

$$\boldsymbol{m}_N = S_N \left(S_0^{-1} \boldsymbol{m}_0 + \sum_{i=0}^{N-1} (t^{(i)} - \frac{1}{2}) K_i \right), \quad (32)$$

$$S_N = S_0^{-1} + 2 \sum_{i=0}^{N-1} \lambda(\rho_{old}^{(i)}) K_i K_i^\top, \quad (33)$$

$$\lambda(\rho_{old}^{(i)}) = -\frac{1}{4\rho_{old}^{(i)}} \tanh\left(\frac{\rho_{old}^{(i)}}{2}\right), \quad (34)$$

M-step:

$$\boldsymbol{\rho}_{new} = \sqrt{K^\top (S_N + \boldsymbol{m}_N \boldsymbol{m}_N^\top) K}, \quad (35)$$

where K is the kernel gram matrix defined $K_{ij} = k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$.

V. Numerical examples

Experiments were conducted to confirm the proposed methods. We used a handwritten numeric data set plus the artificial noise to confirm the validity of feature selection by the proposed methods. Furthermore, we used public data sets to consider the application to pattern recognition problems. We used public data sets to consider the application to pattern recognition problems. Throughout the experiments, we chose the kernel function as

$$k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \exp(-\delta \|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\|^2). \quad (36)$$

For comparison, we chose

- linear logistic regression with ℓ_1 (Linear LR-ℓ_1),
- support vector machine (SVM-RBF) [4],
- kernel logistic regression with ℓ_1 regralization (SKLR-ℓ_1),
- feature selection sparse kernel regression using sampling (FS-SKR-Sa) [25],
- sparse kernel logistic regression using sampling and VB (SKLR-SaVB),
- feature selection sparse kernel logistic regression using sampling and VB (FS-SKLR-Sa).

All parameters were adjusted such that all methods result in similar sparsities.

Algorithm 1 Estimation binary vectors $\boldsymbol{\zeta}$ and $\boldsymbol{z}$ by Gibbs sampling

Input: Initial $\boldsymbol{\zeta}_{init}$, Initial $\boldsymbol{z}_{init}$, Initial $\boldsymbol{\alpha}_{init}$
Output: $\boldsymbol{\zeta}, \boldsymbol{z}, \boldsymbol{\alpha}$
for $t = 1$ to T_{gsfs} **do**
  for $k = 1$ to D **do**
    Caluculate the posterior of $\zeta_k = 1$ from (28).
    if (28) $> \mathbb{U}(0, 1)$ **then**
      $\zeta_k = 1$
    else
      $\zeta_k = 0$
    end if
  end for
  Caluculate $\boldsymbol{\alpha}$ by using VB
  Caluculate the posterior of $\boldsymbol{\zeta}^{[t]}$ from (16)
end for
Select $\boldsymbol{\zeta}$ and $\boldsymbol{\alpha}$ which are the largest among $\left[\boldsymbol{\zeta}^{[0]}, \ldots, \boldsymbol{\zeta}^{[T_{gsfs}]}\right]$.
for $\tau = 1$ to T_{gsms} **do**
  for $i = 1$ to N **do**
    Caluculate the posterior of $z^{(i)} = 1$ from (29).
    if (29) $> \mathbb{U}(0, 1)$ **then**
      $z^{(i)} = 1$
    else
      $z^{(i)} = 0$
    end if
  end for
  Caluculate $\boldsymbol{\alpha}$ by using VB
  Caluculate the posterior of $\boldsymbol{z}^{[\tau]}$ from (17)
end for
Select $\boldsymbol{z}$ and $\boldsymbol{\alpha}$ which are the largest among $\left[\boldsymbol{z}^{[0]}, \ldots, \boldsymbol{z}^{[T_{gsms}]}\right]$.

Algorithm 2 Estimation parmeter $\boldsymbol{\alpha}$ by (VB) inference

Input: Initial $\boldsymbol{\alpha}_{init}$, Initial $\boldsymbol{\rho}_{init} \sim \mathbb{U}(0, 1)$, Initial $S_0 = \xi^{-1} \boldsymbol{I}_N$
Output: $\boldsymbol{\alpha}$
for $t = 1$ to T_{vb} **do**
  E-step: Compute the variational posterior using $\boldsymbol{\rho}_{old}$.
  Caluculate $\lambda^{[t]}(\boldsymbol{\rho}_{old})$ from (34).
  Caluculate $S^{[t]}$ from (33).
  Caluculate $m^{[t]}$ from (32).
  M-step: Re-estimate $\boldsymbol{\rho}_{new}$.
  Caluculate $\boldsymbol{\rho}_{new}^{[t]}$ from (35).
end for
Sampling $\boldsymbol{\alpha}$ from (31).

A. Dataset

1) Handwritten numeric dataset with noise: We use a handwritten numeric dataset from the scikit-learn [26] of Python. In

TABLE I: Summary of the characteristics of the UCI datasets

Datasets	#Example	#Examples/Classes	#Features
australian	690	383 – 307	14
breast canser	683	444 – 239	9
heart	270	150 – 120	13
diabetic	1151	540 – 611	19
ionosphere	351	225 – 126	34
sonar	208	97 – 111	60

TABLE II: Results of 2-class classification for digit with noise.

Algorithm	Accuracy	F1-value	Sparsity
Linear LR-ℓ_1	0.938 ± 0.824	0.940 ± 0.00779	0.302 ± 0.0219
KLR-ℓ_1	0.959 ± 0.00891	0.959 ± 0.00912	0.282 ± 0.239
SVM-RBF	0.954 ± 0.00600	0.953 ± 0.00683	0.257 ± 0.0372
FS-SKR-Sa	0.925 ± 0.0713	0.928 ± 0.00669	0.198 ± 0.0396
SKLR-SaVB	0.960 ± 0.0130	0.952 ± 0.0291	0.407 ± 0.0390
FS-SKLR-SaVB	0.976 ± 0.0926	$\mathbf{0.972 \pm 0.0761}$	0.547 ± 0.0909

this experiment, we chose two numbers: "3" and "8" since they are relatively similar. They are grayscale images scaled to the range between zero and one. To confirm the robustness against noise, a rondom noise generatred from uniform distribution $\sim \mathbb{U}(0,1)$ was added to the input patterns. Fig. 2 shows the digits without and with additive noise, respectively. The numbers of "3" and "8" in the dataset are 183 and 174.

2) UCI machine learning repositry dataset: The performances of proposed methods are tested on six real benchmark datasets from UCI machine learning repository [27]. The 10 datasets are:

1) Australian Sign Language signs Data Set (australian)
2) Breast Cancer Data Set (breast canser)
3) Heart Disease Data Set (heart)
4) Diabetic Retinopathy Debrecen Data Set (diabetic)
5) Ionosphere Data Set (ionosphere)
6) Connectionist Bench (Sonar, Mines vs. Rocks) Data Set (sonar)

The main characteristics of these datasets are illustrated in Table I, which presents, for each dataset: number of examples (#Examples), number of examples per class (#Examples/Classes), number of features (#Features).

B. Evaluation of Model

To test the proposed methods, we have used the classification Accuracy and F1-value (F1) in five-fold cross validation. The F1 is the ability to correctly retrieve the positive data by combining the precision defined by (37) and recall defined by (38) obtained for the positive class.

$$\text{Precision} = \frac{TP}{TP+FP}, \tag{37}$$

$$\text{Recall} = \frac{TP}{TP+FN}, \tag{38}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}, \tag{39}$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{40}$$

In the (37), (38), and (39), TP refers to the number of true positive examples in test data, FP is the number of false positive examples and FN stands for the number of false negative exmamples.

C. Result

Table II shows the result of five-fold cross validation for digits with noise. Fig. 3 shows the result of feature selection using the proposed method (FS-SKLR-SaVB). It is seen that the proposed method (FS-SKLR-SaVB) achieves higher F1-value than the others. This implies the efficacy of feature selection. In addition, it is reasonable to assume that the likelihood function of the observed patterns follows the Bernoulli distribution in the classification problem since the FS-SKLR-SaVB shows better model performance than FS-SKR-Sa.

Table III shows the results of five-fold cross-validation in the UCI dataset. The FS-SKLR-Sa achieved high F1-value in some datasets.

VI. Conclusion

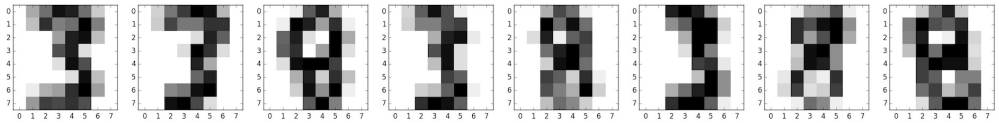
This paper proposed a method for simultaneously selecting features and model coefficients for kernel logistic regression in RKHS. In order to express a sparsity of the features and the weight coefficients, we generate a binary vector where all the elements is either 0 or 1 sampled from the beta process. It was shown that the proposed method could effective features and estimate sparse weight coefficients. Numerical examples supported the efficacy of the proposed method.

Acknowledgment

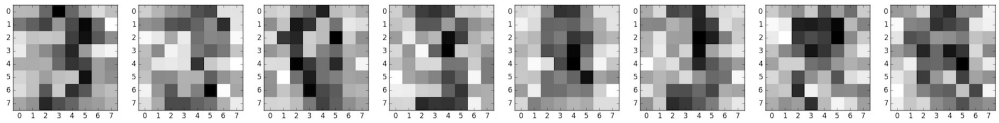
This work is supported by JSPS KAKENHI Grant Number 17H01760.

References

[1] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 215–242, 1958.
[2] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.
[3] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. ACM, 1992, pp. 144–152.
[4] 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.
[5] 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.
[6] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, no. Mar, pp. 1157–1182, 2003.
[7] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, 2012.
[8] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in Neural Information Processing Systems*, vol. 186, 2005, p. 189.
[9] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, 1999.
[10] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.



(a) Digits without noise.

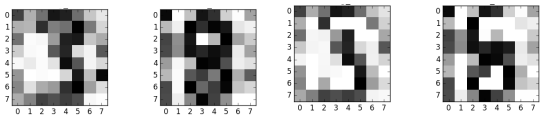


(b) Digits with noise.

Fig. 2: Digits datasets

TABLE III: F1-value of two classification for all data.

Algorithm	australian	breast cancer	heart	diabetic	ionosphere	sonar
Linear LR-ℓ_1	0.801 ± 0.510	0.917 ± 0.309	0.752 ± 0.603	0.601 ± 0.433	0.765 ± 0.123	0.706 ± 0.233
KLR-ℓ_1	0.821 ± 0.133	0.955 ± 0.422	0.778 ± 0.613	0.613 ± 0.233	0.725 ± 0.532	0.704 ± 0.355
SVM-RBF	0.822 ± 0.110	0.955 ± 0.575	$\mathbf{0.797 \pm 0.231}$	$\mathbf{0.628 \pm 0.398}$	0.775 ± 0.492	0.724 ± 0.355
FS-SKR-Sa	0.791 ± 0.345	0.931 ± 0.128	0.751 ± 0.831	0.592 ± 0.233	0.715 ± 0.671	0.710 ± 0.233
SKLR-SaVB	0.817 ± 0.233	0.953 ± 0.821	0.794 ± 0.203	0.618 ± 0.187	0.725 ± 0.745	0.716 ± 0.255
FS-SKLR-SaVB	$\mathbf{0.828 \pm 0.251}$	$\mathbf{0.964 \pm 0.233}$	0.795 ± 0.133	0.620 ± 0.355	$\mathbf{0.785 \pm 0.142}$	$\mathbf{0.726 \pm 0.143}$



(a) Digits before feature selection. (b) Digits after feature selection.

Fig. 3: Digits after feature selection.

[11] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[12] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, vol. 209, no. 1, pp. 237–260, 1998.

[13] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[14] C. Kang, S. Liao, S. Xiang, and C. Pan, "Kernel sparse representation with local patterns for face recognition," in *18th IEEE International Conference on, Image Processing (ICIP)*, 2011, pp. 3009–3012.

[15] S. Gao, I. W. Tsang, and L.-T. Chia, "Sparse representation with kernels," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 423–434, 2013.

[16] J. Goodman *et al.*, "Exponential priors for maximum entropy models." in *HLT-NAACL*, 2004, pp. 305–312.

[17] S.-I. Lee, H. Lee, P. Abbeel, and A. Y. Ng, "Efficient l1 regularized logistic regression," in *AAAI*, vol. 6, 2006, pp. 401–408.

[18] G. C. Cawley and N. L. Talbot, "Gene selection in cancer classification using sparse logistic regression with bayesian regularization," *Bioinformatics*, vol. 22, no. 19, pp. 2348–2355, 2006.

[19] K. Koh, S.-J. Kim, and S. Boyd, "An interior-point method for large-scale l1-regularized logistic regression," *Journal of Machine Learning Research*, vol. 8, no. Jul, pp. 1519–1555, 2007.

[20] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[21] J. Paisley and L. Carin, "Nonparametric factor analysis with beta process priors," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 777–784.

[22] B. Chen, J. Paisley, and L. Carin, "Sparse linear regression with beta process priors," in *2010 IEEE International Conference on, Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 1234–1237.

[23] A. Kojima, S. Yasutomi, and T. Tanaka, "Sparse kernel regression based on nonparametric bayesian model," *IEICE SIP*, pp. 335–340, Mar. 2016.

[24] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 721–741, 1984.

[25] A. Kojima, T. Wadai, and T. Tanaka, "Dual-sparsification of kernel regression based on graphical model," in *ITC-CSCC: International Technical Conference on Circuits Systems, Computers and Communications*, 2017, pp. 192–195.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml