Confusion-matrix-based Kernel Logistic Regression for Imbalanced Data Classification

Miho Ohsaki, *Member, IEEE*, Peng Wang, Kenji Matsuda, *Member, IEEE*, Shigeru Katagiri, *Fellow, IEEE*, Hideyuki Watanabe, *Member, IEEE*, and Anca Ralescu, *Senior, IEEE*

Abstract—There have been many attempts to classify imbalanced data, since this classification is critical in a wide variety of applications related to the detection of anomalies, failures, and risks. Many conventional methods, which can be categorized into sampling, cost-sensitive, or ensemble, include heuristic and task dependent processes. In order to achieve a better classification performance by formulation without heuristics and task dependence, we propose confusion-matrix-based kernel logistic regression (CM-KLOGR). Its objective function is the harmonic mean of various evaluation criteria derived from a confusion matrix, such criteria as sensitivity, positive predictive value, and others for negatives. This objective function and its optimization are consistently formulated on the framework of KLOGR, based on minimum classification error and generalized probabilistic descent (MCE/GPD) learning. Due to the merits of the harmonic mean, KLOGR, and MCE/GPD, CM-KLOGR improves the multifaceted performances in a well-balanced way. This paper presents the formulation of CM-KLOGR and its effectiveness through experiments that comparatively evaluated CM-KLOGR using benchmark imbalanced datasets.

Index Terms—Imbalanced Data, Confusion Matrix, Kernel Logistic Regression, Minimum Classification Error and Generalized Probabilistic Descent

1 Introduction

Data that consists of two classes, in which the number and/or proportion of instances extremely differ between the classes, is called imbalanced data (See Fig. 1). Typically, one class has a large amount of instances (i.e., the majority) and is of less interest and labeled as negative. The other class has a small amount of instances (i.e., the minority) and is of more interest and labeled as positive. We frequently encounter this type of data in real problems related to anomalies, failures, and risks, such as medical diagnosis, oil spill detection, and banking fraud monitoring [1], [2], [3], [4], [5].

However, classifiers which have no mechanism to handle imbalance often lead to a useless result that rare but serious cases are ignored, e.g., a 95% accuracy can be easily achieved by ignoring 5% cancer patients. On the other hand, it is also problematic to regard many healthy people as cancer patients, since it results in costs for needless clinical tests and treatments. Considering these requirements, it is highly needed, especially in biomedical fields, to make a well-balanced improvement in all the evaluation criteria derived from a confusion matrix.

Because of the importance and difficulty of imbalanced data classification, many attempts have been made to develop imbalanced data classifiers. Conventional methods are categorized into those based on sampling, misclassification costs, or an ensemble of classifiers, and they share a similar approach that is aimed at correcting the imbalance [1], [2], [3], [4], [5]. These methods are specifically developed to deal with imbalanced data, and achieve a better performance than other classifiers. However, their approach has heuristic and task dependent aspects, and hence is less general.

In order to solve the conventional problems and achieve high performance, this paper proposes a novel imbalanced data classifier, which we call confusion-matrix-based kernel logistic regression (CM-KLOGR). Aiming to well-balancedly raise the values of all the evaluation criteria derived from a confusion matrix, CM-KLOGR combines the following elements into a consistent formulation: the harmonic mean of evaluation criteria derived from a confusion matrix, kernel logistic regression (KLOGR) [6], [7], and minimum classification error and generalized probabilistic descent (MCE/GPD) learning [8]. For efficient and effective optimization, pretraining based on the discriminative model approach and retraining based on the discriminant function approach are introduced [9].

Although one may think that CM-KLOGR is just another method based on misclassification costs [1], [2], [3], [4], [5], it is distinct from such cost-sensitive methods. A conventional

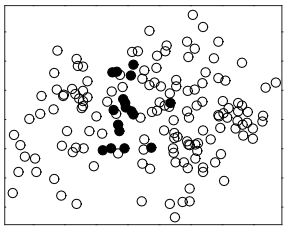


Fig. 1. Imbalanced data consisting of the majority, less interesting, and negative class ○, and the minority, more interesting, and positive class ●.

- *M. Ohsaki, P. Wang, K. Matsuda, and S. Katagiri are with Graduate School of Science and Engineering, Doshisha University, 1-3 Tataramiyakodani, Kyotanabe-shi, Kyoto 610-0321, Japan.*
- *H. Watanabe is with National Institute of Information and Communications Technology, 3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan.*
- *A. Ralescu is with Department of Electrical Engineering and Computing Systems, College of Engineering and Applied Science, University of Cincinnati, 812 Rhodes Hall, Cincinnati, Ohio 45221-0030, USA.*

Manuscript received June 14, 2016; revised February 24, 2017.

cost-sensitive classifier indirectly increases the values of the evaluation criteria through an objective function defined by the costs that were set subjectively or in a trial-and-error way by a user. In contrast, CM-KLOGR directly increases these values by embedding the evaluation criteria into its objective function. Thus, it has an ability to lead to a well-balanced improvement of these criteria with no user intervention.

This paper is organized as follows. Section 1 presents the relevant background and our objective for imbalanced data classification. Section 2 reviews the conventional imbalanced data classifiers and discusses their abilities and limitations. Section 3 provides the concepts and techniques that are the elements of our CM-KLOGR. Section 4 is devoted to the proposal and formulation of CM-KLOGR by integrating these elements. Section 5 reports Experiment I, in which the performance of CM-KLOGR was evaluated by comparison with kernel logistic regression (KLOGR), support vector machine (SVM), and their sampling versions. Section 6 reports Experiment II, which evaluated how CM-KLOGR works in cases that emphasize specific evaluation criteria. Section 7 concludes the paper.

2 Conventional Classifiers for Imbalanced Data

In the light of classification stages, conventional methods are categorized into preprocessing, special-purpose learning, postprocessing, and hybrid approaches [5]. Focusing on elementary techniques to correct the imbalance, they can be categorized into sampling, cost-sensitive, and ensemble approaches [1], [2], [3], [4], [5]. Sampling methods [10], [11], [12], [13], [14], [15], [16], [17] are a kind of preprocessing rather than classifiers themselves. They attempt to improve a single classifier by reducing the classification bias in terms of the bias-variance decomposition [9], [18]; undersampling, oversampling, or strategic sampling are used to compensate for imbalanced data.

In order to equalize the number or proportion of instances between the majority and minority classes, undersampling deletes the part of the training data that belongs to the majority class, and oversampling duplicates the part of the training data that belongs to the minority class. Strategic sampling is an advanced version of undersampling and oversampling. It estimates the distance and/or distribution of data, and then this information is used for a strategy to remove disturbance instances and generate beneficial virtual instances. It has been reported that undersampling and oversampling based on simple random selection work but not very well, and strategic sampling works better when its strategy is adequate. Distance thresholding or clustering is essential in the sampling methods, and their settings (such as the definition of distance, the value of the threshold, and the number of clusters) are based on heuristics depending on the applications.

Cost-sensitive methods [12], [15], [19], [20], [21], [22] attempt to reduce the classification bias of a single classifier, as well as the sampling methods attempt. They try to improve the classifier by reflecting information about the significance of the classification results into the objective function. Such information is represented by separate costs for classifying an instance into the majority class and for classifying an instance into the minority class. Specifically, the costs are put on the numbers of true negatives, false negatives, true positives, and false negatives. Although the cost-sensitive methods have an ability to raise the classification performance, their success depends on application-specific costs. It is required to adjust the costs based on user's subjectivity or trials and errors, because the objective function is a sum of the costs (not the evaluation criteria), and the way in which it improves the classification is indirect.

The idea of ensemble methodology has been proposed for classification in general [9], [18]. We explain the general ensemble methods here, because the ensemble methods specific to imbalanced data classification [13], [14], [16], [21], [23], [24], [25] have the same characteristics as the general ensemble methods. The ensemble methods are the way to combine classifiers, primarily aiming at the reduction of classification variance in terms of the bias-variance decomposition. They train a set of base classifiers to complement each other, and make decisions based on a committee of these classifiers. They are categorized into bagging, boosting, or stacking [18], [26], [27], [28], [29].

In bagging, training of classifiers is accomplished with the replacement of bootstrap samples that are randomly and duplicately selected from training data. The objective function is defined by the majority vote of the classifiers. An effective example of bagging is random forest that includes variable sampling and consists of decision trees. Unlike bagging, boosting evolves the committee process by weighting. It assigns a weight for each classifier to each sample, updates these weights according to the loss of misclassification, and makes a decision by weighted majority voting. Stacking iteratively trains the classifiers and their weights in a manner of cross-validation, and its decision making is based on a weighted majority vote.

The ensemble methods for imbalanced data classification share the benefits of the original ensemble methods, in that the classification variance is low, and the theoretical background has been established [9], [18]. These benefits make the methods promising, but there exists a serious issue, namely, how to define an objective function that is suitable for imbalanced data. Conventional ensemble methods use an objective function of sampling or cost-sensitive methods, and inevitably suffer from the same problems from which these methods suffer.

3 Conceptual and Technical Elements for Our Classifier

3.1 Kernel Logistic Regression (KLOGR)

CM-KLOGR extends and combines the concepts and techniques of KLOGR, MCE/GPD, and F-Measure. KLOGR [6], [7] is the kernelized version of logistic regression (LOGR) [9], [18] that is based on the discriminative model approach and a common classifier in biomedical fields. LOGR provides both the predicted class and its estimated posterior probability, which is important as a confidence measure in such fields [30], [31]. KLOGR inherits this advantage and also overcomes the disadvantage that LOGR cannot achieve high performance due to its linearity; KLOGR does this by the kernelization which generates nonlinear boundaries. In

a previous study [32], KLOGR was applied to an imbalanced biomedical dataset, and superior to the one-nearest neighbor method, multivariate linear regression, LOGR, regularized LOGR, and SVM. Other biomedical studies also applied KLOGR and showed its effectiveness [33], [34], [35], [36]. Because of those, we focus on KLOGR.

The source for estimating the posterior probabilities of classes in KLOGR, $y_k(\mathbf{x}; \boldsymbol{\alpha}_k, b_k)$ (for simplicity, denoted as $y_k(\mathbf{x})$), is shown in Eq. (1). This is defined as a weighted sum of the kernels for the k-th class, parameterized by the parameter vector $\boldsymbol{\alpha}_k = [\alpha_{1k}, \alpha_{2k}, \cdots, \alpha_{Nk}]^T$ and the bias term b_k. $\mathbf{x}$ is the feature vector to be classified, and $\mathbf{x}_m$ is the feature vector of the m-th instance in the training data. $\mathcal{K}(\mathbf{x}, \mathbf{x}_m)$ is the kernel function that represents the similarity between $\mathbf{x}$ and $\mathbf{x}_m$, and $\boldsymbol{\kappa}(\mathbf{x})$ is a vector containing $\mathcal{K}(\mathbf{x}, \mathbf{x}_m)$ for $m = 1$ to N. The most frequently used function is the Gaussian kernel $\mathcal{K}(\mathbf{x}, \mathbf{x}_m) = \exp(-||\mathbf{x} - \mathbf{x}_m||^2/2\sigma^2)$, in which σ is a hyperparameter.

$$y_k(\mathbf{x}) = \sum_{m=1}^{N} \alpha_{mk} \mathcal{K}(\mathbf{x}, \mathbf{x}_m) + b_k = \boldsymbol{\alpha}_k^T \boldsymbol{\kappa}(\mathbf{x}) + b_k \quad (1)$$

Using $y_k(\mathbf{x})$, the k-th class posterior probability $\Pr(C_k|\mathbf{x})$ is defined as Eq. (2) in the form of a softmax function, where K is the number of classes.

$$\Pr(C_k|\mathbf{x}) = \frac{\exp(y_k(\mathbf{x}))}{\sum_{l=1}^{K} \exp(y_l(\mathbf{x}))} \quad (2)$$

The objective function, $J(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_K)$, is the cross-entropy error function with a regularization term shown in Eq. (3). This function indicates how well the posterior probabilities of classes are estimated under the L^2-norm constraint. There are some choices on how to set b_k: augmenting $\mathbf{x}$ to embed b_k into the vector $\boldsymbol{\alpha}_k$ (b_k is one of the variables in the objective function), and adjusting b_k outside of training (b_k is fixed to 0 when training). The second choice, which is comparatively common for the application of LOGR to biomedical data, is selected in the present study. Hence, Eq. (3) is a function of $\boldsymbol{\alpha}_k$ only.

$$J(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_K) = \quad (3)$$
$$-\sum_{n=1}^{N} \sum_{k=1}^{K} \delta_{k_n,k} \ln \Pr(C_k|\mathbf{x}_n) + \frac{\lambda}{2} \sum_{k=1}^{K} \boldsymbol{\alpha}_k^T \boldsymbol{\mathcal{K}} \boldsymbol{\alpha}_k$$

where the Kronecker delta function $\delta_{k_n,k}$ counts one when k is identical to the correct class k_n of the n-th instance, namely a correct classification. The weight λ represents how much emphasis is put on the regularization term, and it is a hyperparameter. $\boldsymbol{\mathcal{K}}$ denotes the Kernel matrix in which the elements are the values of the kernel function $\mathcal{K}(\mathbf{x}, \mathbf{x}')$, as calculated for all combinations of instances $\mathbf{x}$ and $\mathbf{x}'$ in the training data.

Although the original KLOGR does not have a regularization term, it is recommended to use one, since restricting the search range in a parameter space leads to a more stable performance with a smaller classification variance [9], [18]. Any types of norms such as the L^1, L^2, or higher order ones are acceptable for regularization. The L^2-norm is a reasonable choice, since it ensures a clear theoretical relationship between KLOGR and both SVM and the Gaussian process classification (GPC) [9], [18].

The objective function of SVM [37], [38] consists of two terms: an empirical hinge loss to penalize incorrect classification and a geometric margin to ensure generalization. Maximizing the geometric margin of SVM is identical to minimizing the L^2-norm of the parameters except the bias term, in a two-class classification. This suggests that the L^2-norm regularization of KLOGR maximizes the geometric margin, as in SVM. In terms of GPC [9], KLOGR with L^2-norm regularization is the simplest implementation of GPC.

With regard to the parameters α_{mk}, $J(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_K)$ is convex, and hence the unique optimal point in the parameter space is reachable by the gradient descent method. In contrast, the hyperparameters, which are the width of the Gaussian kernel σ and the weight on the regularization term λ, must be set before training. Depending on the formulation, the bias term of the regression function b_k in Eq. (1) may also need to be set after training. It is common for classifiers, including KLOGR, to set the hyperparameters (and the bias term, if one is needed) by performing a grid search using validation dataset [39]. In KLOGR, after setting the parameters and hyperparameters, the classification decision is made by selecting the class with the highest estimated probability.

KLOGR is an effective classifier that can draw nonlinear boundaries and provides the posterior probabilities of classes as a confidence, of which effectiveness was shown in the literatures [32], [33], [34], [35], [36]. Hence, it is expected that KLOGR achieves higher performance by introducing a new objective function, which is specific to imbalanced data.

3.2 Minimum Classification Error and Generalized Probabilistic Descend (MCE/GPD)

MCE/GPD [8] is a learning method based on the discriminant function approach [9], which directly controls the class boundaries, unlike the approaches based on distribution estimation. It has been successfully applied to speech recognition, and intensively extended and improved [8], [40], [41], [42]. In MCE/GPD, the discriminant function $y_k(\mathbf{x}; \boldsymbol{\Lambda}_k)$ abbreviated as $y_k(\mathbf{x})$ is defined in Eq. (4).

$$y_k(\mathbf{x}) = f(\boldsymbol{\Lambda}_k, \mathbf{x}) \quad (4)$$

where $\boldsymbol{\Lambda}_k$ denotes a set of parameters for the k-th class. Any differential positive functions of $\boldsymbol{\Lambda}_k$ are acceptable as $f(\boldsymbol{\Lambda}_k, \mathbf{x})$. For instance, the simplest function can be $y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + b_k$, where $\mathbf{w}_k$ and b_k denote the parameter vector of input variables and the bias term, respectively.

The misclassification measure d_{k_n} is defined in Eq. (5).

$$d_{k_n}(\mathbf{x}_n) = -y_{k_n}(\mathbf{x}_n) + \left[\frac{1}{K-1} \sum_{j,\ j \neq k_n} y_j(\mathbf{x}_n)^\eta \right]^{\frac{1}{\eta}} \quad (5)$$
$$\approx -y_{k_n}(\mathbf{x}_n) + \max_{j,\ j \neq k_n} y_j(\mathbf{x}_n) \quad (\text{if } \eta \to \infty)$$

where η is a positive constant for a parametric maximum selection operation, and k_n represents the correct class of the n-th instance. $y_j(\mathbf{x}_n)$ estimates the degree of belonging of the n-th instance to the j-th class. $y_{k_n}(\mathbf{x}_n)$ has a similar meaning but is specific to the case in which the correct class of $\mathbf{x}_n$ is the k-th class. The negative and positive values of d_{k_n} mean correct and incorrect classifications, respectively; d_{k_n} represents the signed degree of classification correctness

(less than 0) or incorrectness (more than 0). It is essential for Eq. (5) that $y_k(\mathbf{x}_n)$ be positive, and thus if necessary, $y_k(\mathbf{x}_n)$ must be normalized to be positive by such as logarithmic and/or exponential transformations [8].

The function defined in Eq. (6) is a differentiable smoothed 0-1 loss function that penalizes misclassification in the form of a sigmoid function. The hyperparameter $\epsilon > 0$ determines the smoothness, that is, how close the function is to a 0-1 step function. The large and small values of ϵ tend to cause overfitting and underfitting, respectively, and thus its value should be properly set. The objective function $J(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \cdots, \mathbf{\Lambda}_K)$ is then formulated as the sum of the loss over the N instances of training data, as shown in Eq. (7). This objective function makes MCE/GPD directly pursue a reduction in misclassifications, and makes it possible to use the gradient descent method for optimization.

$$l(d_{k_n}(\mathbf{x}_n)) = \frac{1}{1 + \exp(-\epsilon d_{k_n}(\mathbf{x}_n))} \tag{6}$$

$$J(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \cdots, \mathbf{\Lambda}_K) = \frac{1}{N} \sum_{n=1}^{N} l(d_{k_n}(\mathbf{x}_n)) \tag{7}$$

MCE/GPD takes a straightforward route to the correct classification, and it has the capacity to express any evaluation criteria that are based on the smoothed 0-1 loss function. It is expected to achieve better performance by formulating an objective function and its learning process, which is based on MCE/GPD and specific to imbalanced data classification. However, if we do so for CM-KLOGR, such an objective function is not convex, and that causes difficulties with parameter optimization. In contrast, the objective function of KLOGR is convex and leads to smooth parameter optimization, similar to that of SVM. Taking these aspects into account, it is worthwhile to bring out the potentials of MCE/GPD and KLOGR in imbalanced data classification by their combination.

Generally, classification is categorized into the generative model, the discriminative model, and the discriminant function approaches [9]. For parameter optimization when the objective function is nonconvex, it is common to combine pretraining based on the generative/discriminative model approach and retraining based on the discriminant function approach, e.g., fine-tuning after clustering [8], fine-tuning after distribution estimation [43], etc. This suggests a way to unlock the potentials of MCE/GPD and KLOGR.

3.3 Evaluation Criteria for Classification Performance

Multifaceted evaluation criteria are required for the evaluation of the performance of imbalanced data classification. Such criteria that can be derived from a confusion matrix include sensitivity (Sens), specificity (Spec), positive predictive value (PPV), negative predictive value (NPV), and accuracy (Acc) [5], [44], [45], [46]. Sens, Spec, and PPV are known as true positive rate (TPR) or recall, true negative rate (TNR), and precision, respectively. Another commonly used criterion is the area under the curve of the receiver operating characteristic (AUC), which accumulates the points of two criteria, such as Sens and Spec, over their different parameter settings. Therefore, AUC cannot be used when evaluating the performance under an optimal parameter setting. Sens, Spec, PPV, NPV, and Acc are defined in Eqs. (8) to (12), respectively, where N is the total number of instances, N_{TP} is the number of true positives, N_{TN} is the number of true negatives, N_{FP} is the number of false positives, and N_{FN} is the number of false negatives [5], [44], [45].

$$\mathrm{Sens} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}} + N_{\mathrm{FN}}} \tag{8}$$

$$\mathrm{Spec} = \frac{N_{\mathrm{TN}}}{N_{\mathrm{TN}} + N_{\mathrm{FP}}} \tag{9}$$

$$\mathrm{PPV} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}} + N_{\mathrm{FP}}} \tag{10}$$

$$\mathrm{NPV} = \frac{N_{\mathrm{TN}}}{N_{\mathrm{TN}} + N_{\mathrm{FN}}} \tag{11}$$

$$\mathrm{Acc} = \frac{N_{\mathrm{TP}} + N_{\mathrm{TN}}}{N} \tag{12}$$

A classifier with no adjustment for imbalanced data tends to assign all the instances to the majority (negative) class. In that case, Spec, NPV, and Acc are all large, and that makes the classification appear to be successful. However, it is actually a failure as indicated by the low values of Sens and PPV; the classifier overlooks the instances in the minority (positive) class of more interest, as if they are not interesting. Figuratively speaking, sick people are ignored and not treated. Let us consider another extreme case in which a classifier tends to assign all the instances to the minority (positive) class, due to too much imbalance correction. It achieves high Sens, but does not necessarily increase PPV and results in low Spec, NPV, and Acc; the classifier picks up the instances in the majority (negative) class of less interest, as if they are interesting. Healthy people are given needless treatment, and this causes the waste of medical expense.

These cases show the necessity of a well-balanced improvement to these criteria, leading to the combinational use of these criteria for training. Especially, domains such as biomedicine are supposed to evaluate the classification performance using not only Sens and PPV, but also Spec and NPV (and occasionally Acc). In fact, the combination of the words, sensitivity, specificity, positive predictive value, and negative predictive value, received more than 16000 hits on PubMed [47], which is one of the most widely used databases of biomedical literature.

When using multiple evaluation criteria, it is difficult to evaluate their balance and total performance. This difficulty is common in information retrieval as in imbalanced data classification. For comprehensive evaluation, information retrieval algorithms use F-measure [48], the harmonic mean of Sens and PPV, namely recall and precision. The intent is to balance them in a way that is more sensitive to the difference in their values than the arithmetic mean. F-measure suggests how to synthesize and utilize evaluation criteria for imbalanced data.

4 Proposal of Confusion-Matrix-Based Kernel Logistic Regression (CM-KLOGR)

4.1 Ideas Behind CM-KLOGR

We propose a novel classifier: a confusion-matrix-based kernel logistic regression (CM-KLOGR) [49]. Its main idea is to directly improve various evaluation criteria, while balancing them each other, by the formulation of a consistent learning mechanism based on F-measure [48], KLOGR [6], [7], and MCE/GPD [8]. In this section, the detailed ideas are discussed in order of the model structure, objective function, and optimization of CM-KLOGR.

TABLE 1
List of symbols used for the formulation of CM-KLOGR.

K: number of classes.
N: number of instances in the training data.
$\mathbf{x}$: feature vector.
$\mathbf{x}_m$: feature vector of the m-th instance in the training data.
C: class.
C_k: k-th class, where $k \in \{1, \cdots, K\}$. In two-class classification, C_1 corresponds to the negative class and C_2 to the positive.
$\mathcal{K}(\mathbf{x}, \mathbf{x}_m)$: kernel function that represents the similarity between $\mathbf{x}$ and $\mathbf{x}_m$.
$\boldsymbol{\kappa}(\mathbf{x})$: vector consisting of the kernel functions; $[\mathcal{K}(\mathbf{x}, \mathbf{x}_1), \mathcal{K}(\mathbf{x}, \mathbf{x}_2), \cdots \mathcal{K}(\mathbf{x}, \mathbf{x}_N)]^T$.
$\boldsymbol{\mathcal{K}}$: Kernel matrix consisting of the kernel function vectors; $[\boldsymbol{\kappa}(\mathbf{x}_1), \boldsymbol{\kappa}(\mathbf{x}_2), \cdots \boldsymbol{\kappa}(\mathbf{x}_N)]$.
$\boldsymbol{\alpha}$: parameter vector of weights for the similarities of an instance to the others in the training data; $[\alpha_1, \alpha_2, \cdots, \alpha_N]^T$.
$\boldsymbol{\alpha}_k$: parameter vector for the k-th class.
b: bias term which is a scalar.
b_k: bias term for the k-th class.
$y_k(\mathbf{x})$: a kernel regression function for the k-th class, of which parameters are $\boldsymbol{\alpha}_k$ and b_k.
$\Pr(C_k\|\mathbf{x})$: posterior probability of the k-th class when $\mathbf{x}$ is input.
k_n: variable to indicate the correct class of the n-th instance.
$\Pr(C_{k_n}\|\mathbf{x}_n)$: posterior probability of the correct class indicated by k_n, i.e., C_{k_n} when $\mathbf{x}_n$ is input.
$d_{k_n}(\mathbf{x}_n)$: misclassification measure when classifying the n-th instance of which correct class is C_{k_n}.
η: positive constant to parametrically formulate maximum selection.
$l(d_{k_n}(\mathbf{x}_n))$: smoothed 0-1 loss function that penalizes a misclassification.
ϵ: positive constant to determine the smoothness of the loss function.
N_{TP}, N_{FP}, N_{TN}, and N_{FN}: numbers of the true positive, false positive, true negative, and false negative instances in the training data.
$\delta_{l,k}$: Kronecker delta function of which the value is 1 when $l = k$ and 0 when $l \neq k$.
f_i: i-th evaluation criterion.
ξ_i: numerator of the i-th evaluation criterion function.
ψ_i: denominator of the i-th evaluation criterion function.
J: objective function used in retraining.
J_{HM}: first term of the objective function which is the harmonic mean of the evaluation criteria.
J_{L2}: second term of the objective function which is the L^2-norm regularization.
N_{ec}: number of evaluation criteria.
γ_i: weight on the i-th evaluation criterion.
S_γ: summation of γ_i over all the i, $S_\gamma = \sum_{i=1}^{N_{\mathrm{ec}}} \gamma_i$.
λ: weight that balances the harmonic mean of the evaluation criteria and the L^2-norm regularization.

As the framework on which to develop CM-KLOGR and its pretraining, KLOGR is selected because of the reasons below. It has the ability to draw flexible nonlinear class boundaries by kernelization; to converge to the optimal point in the parameter space, due to the convexity of the objective function; and to derive the posterior probabilities of the classes, which can be used as a confidence measure. Compared to classifiers based on the generative model approach [9] that also provide the probabilities, KLOGR has fewer parameters and is expected to work well even if the training data is small. It actually worked well for imbalanced biomedical datasets [32], [33], [34], [35], [36].

The key point of CM-KLOGR is to introduce a new objective function. This function can be defined based on the idea of F-measure (Precisely speaking, based on the harmonic mean which is a more general concept than F-measure). The harmonic mean is sensitive to the difference between its components, and consequently balances them. It is thus reasonable to define the objective function of CM-KLOGR as the harmonic mean of various evaluation criteria. Combining various evaluation criteria may seem redundant, because of their trade-off. This is theoretically true if the classifier is perfectly optimized. In such a case, increasing some criteria causes a decrease in the others. However, on the way to the optimal setting, there is room to simultaneously increase various evaluation criteria.

MCE/GPD shows the way to the formulation and optimization of the new objective function. It has the ability to formulate the evaluation criteria derived from a confusion matrix in conjunction with a smoothed 0-1 loss function; to straightforwardly improve these evaluation criteria; and to make the gradient descent method applicable. For CM-KLOGR, it is promising to formulate the evaluation criteria and their harmonic mean, and optimize it via MCE/GPD.

There is a problem to overcome, the difficulty in optimization because of the nonconvexity of this objective function. Therefore, a two-stage training is adopted, which consists of pretraining based on the generative/discriminative model approach and retraining based on the discriminant function approach. In CM-KLOGR, the parameters of KLOGR are initialized using the cross-entropy error function in pretraining, and fine-tuned using the harmonic mean in retraining. Additionally, the L^2-norm regularization, which works as geometric margin maximization, is introduced to these two objective functions. It is expected that CM-KLOGR will lead to smooth optimization and generalization due to the two-stage training and the regularization.

Note that a classifier proposed in the literature [50] has some similarities to CM-KLOGR with respect to a discriminant function approach. However, it differs from CM-KLOGR in that it was not intended for the classification of imbalanced data, its framework was logistic regression with no kernelization, it embedded only F-measure, and optimization difficulties due to nonconvexity were left unresolved. CM-KLOGR overcomes these remaining problems.

4.2 Formulation of CM-KLOGR

This section defines and formulates the model structure, objective function, and optimization of CM-KLOGR. Table 1 lists the symbols used for that. The formulation of CM-KLOGR starts from that of KLOGR generally defined for multi-class classification, and is specialized for two-class imbalanced data classification. As in KLOGR (see Eqs.(1) and (2)), CM-KLOGR has the model structure to estimate the posterior probabilities of the classes by substituting the regression function of the kernels in Eq. (13) into the softmax function in Eq. (14).

$$y_k(\mathbf{x}) = \sum_{m=1}^{N} \alpha_{mk} \mathcal{K}(\mathbf{x}, \mathbf{x}_m) + b_k = \boldsymbol{\alpha}_k^T \boldsymbol{\kappa}(\mathbf{x}) + b_k \tag{13}$$

$$\Pr(C_k|\mathbf{x}) = \frac{\exp(y_k(\mathbf{x}))}{\sum_{l=1}^{K} \exp(y_l(\mathbf{x}))} \tag{14}$$

For pretraining and retraining, two objective functions and their respective optimization processes are formulated.

In the pretraining process, the objective function is the cross-entropy error function, and the optimization process is the gradient descent method; these are identical to those used in KLOGR. The parameter setting accomplished by pretraining is passed into retraining as an initialization status, and it is then fine-tuned by the gradient descent method, using the new objective function, as developed in the following steps.

We begin by associating the class posterior probabilities of KLOGR with the misclassification measure of MCE/GPD. Here, $\Pr(C_k|\mathbf{x})$ in Eq. (14), which is the posterior probability of the k-th class given $\mathbf{x}$, is regarded as the discriminant function of this class. By substituting it into Eq. (5), the misclassification measure d_{k_n} of Eq. (15) is obtained.

$$\begin{aligned} & d_{k_n}(\mathbf{x}_n) \qquad (15)\\ & = -\Pr(C_{k_n}|\mathbf{x}_n) + \left[\frac{1}{K-1}\sum_{j,\ j\neq k_n}\Pr(C_j|\mathbf{x}_n)^{\eta}\right]^{\frac{1}{\eta}} \\ & \approx -\Pr(C_{k_n}|\mathbf{x}_n) + \max_{j,\ j\neq k_n}\Pr(C_j|\mathbf{x}_n) \quad (\text{if } \eta\to\infty) \end{aligned}$$

The smoothed 0-1 loss function in Eq. (16) penalizes misclassifications based on the sign and absolute value of the misclassification measure $d_{k_n}(\mathbf{x}_n)$.

$$l(d_{k_n}(\mathbf{x}_n)) = \frac{1}{1+\exp(-\epsilon d_{k_n}(\mathbf{x}_n))} \quad (\epsilon>0) \qquad (16)$$

By treating this loss as an approximate count of misclassifications, the numbers of true positives, false positives, true negatives, and false negatives are defined as shown in Eqs. (17) to (20), respectively. These numbers are specific to two-class classification, and hence, we set $k_n = 1$ for the negative class C_1 and $k_n = 2$ for the positive class C_2. In Eq. (17), $l(d_{k_n}(\mathbf{x}_n))$ represents the count of incorrect classifications, and accordingly, $1-l(d_{k_n}(\mathbf{x}_n))$ represents the count of correct classifications. $\delta_{k_n,2}$, which is multiplied to $1-l(d_{k_n}(\mathbf{x}_n))$, picks up a case when the correct class is C_2. Eq. (17), the summation of the multiplication of these terms, therefore represents the number of true positives. Similar interpretations apply to Eqs. (18) to (20).

$$N_{\text{TP}} \approx \sum_{n=1}^{N}(1-l(d_{k_n}(\mathbf{x}_n)))\delta_{k_n,2} \qquad (17)$$

$$N_{\text{FP}} \approx \sum_{n=1}^{N} l(d_{k_n}(\mathbf{x}_n))\delta_{k_n,1} \qquad (18)$$

$$N_{\text{TN}} \approx \sum_{n=1}^{N}(1-l(d_{k_n}(\mathbf{x}_n)))\delta_{k_n,1} \qquad (19)$$

$$N_{\text{FN}} \approx \sum_{n=1}^{N} l(d_{k_n}(\mathbf{x}_n))\delta_{k_n,2} \qquad (20)$$

Substituting N_{TP}, N_{FP}, N_{TN}, and N_{FN} into Eqs. (8) to (12), the evaluation criteria Sens, Spec, PPV, NPV, and Acc are defined as shown in Eqs. (21) to (25), respectively.

$$f_1 = \text{Sens} = \frac{N_{\text{TP}}}{N_{\text{TP}}+N_{\text{FN}}} = \xi_1\psi_1^{-1} \qquad (21)$$

$$f_2 = \text{Spec} = \frac{N_{\text{TN}}}{N_{\text{TN}}+N_{\text{FP}}} = \xi_2\psi_2^{-1} \qquad (22)$$

$$f_3 = \text{PPV} = \frac{N_{\text{TP}}}{N_{\text{TP}}+N_{\text{FP}}} = \xi_3\psi_3^{-1} \qquad (23)$$

$$f_4 = \text{NPV} = \frac{N_{\text{TN}}}{N_{\text{TN}}+N_{\text{FN}}} = \xi_4\psi_4^{-1} \qquad (24)$$

$$f_5 = \text{Acc} = \frac{N_{\text{TP}}+N_{\text{TN}}}{N} = \xi_5\psi_5^{-1} \qquad (25)$$

where ξ_i and ψ_i denote the upper and lower terms of each fraction, respectively. They are introduced to simplify the result of objective function differentiation.

HM, the weighted harmonic mean of the evaluation criteria (Sens, Spec, PPV, NPV, and Acc), is defined as in Eq. (26), where S_γ is the summation of all the weights as shown in Table 1. This HM is able to represent any combinations of these criteria by assigning proper weights on them. Note that it can be used not only for training (parameter setting), but also for validating (hyperparameter and cutoff setting) and testing (generalized performance evaluation). The default setting of the weight γ_i is to assign 1 for all i (or for all except $i=5$ corresponding to Acc). However, to meet the needs of applications, γ_i can be determined by the importance of the i-th evaluation criterion. Actually, γ_i is set differently in the evaluation experiments.

$$\text{HM} = \frac{1}{S_\gamma}\left[\left(\sum_{i=1}^{N_{\text{ec}}}\frac{\gamma_i}{f_i}\right)\right]^{-1} \qquad (26)$$

The objective function of CM-KLOGR for retraining $J(\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\cdots,\boldsymbol{\alpha}_K)$, simplified as J, is defined as in Eq. (27). Its first term J_{HM} is the weighted harmonic mean of the evaluation criteria, which is defined in Eq. (26). For indicating that HM is used in training, we replaced the symbol HM with J_{HM}. The second term J_{L2} of the objective function is the L^2-norm regularization.

$$\begin{aligned} J &= J_{\text{HM}} + J_{\text{L2}} \qquad (27)\\ &= -\left[\frac{1}{S_\gamma}\left(\sum_{i=1}^{N_{\text{ec}}}\frac{\gamma_i}{f_i}\right)\right]^{-1} + \frac{\lambda}{2}\sum_{k=1}^{K}\boldsymbol{\alpha}_k^T\mathcal{K}\boldsymbol{\alpha}_k \end{aligned}$$

Starting from a favorable initial setting obtained by pretraining using Eq. (3), the parameters $\boldsymbol{\alpha}_k$ are fine-tuned by retraining using Eq. (27). In retraining, the first term of Eq. (27) will improve all the evaluation criteria in a well-balanced way, and the second one will avoid overfitting.

For the optimization by the gradient descent method, the objective function J is differentiated with regard to $\boldsymbol{\alpha}_{k'}$, where k' indicates each class, C_1 or C_2. The differentiation $\frac{\partial J}{\partial \boldsymbol{\alpha}_{k'}}$ is divided into two terms in Eq. (28). The second term $\frac{\partial J_{\text{L2}}}{\partial \boldsymbol{\alpha}_{k'}}$ can be directly calculated. It is necessary to decompose the first term $\frac{\partial J_{\text{HM}}}{\partial \boldsymbol{\alpha}_{k'}}$ by applying the chain rule. The first part of the decomposition result $\frac{\partial J_{\text{HM}}}{\partial f_i}$ can be directly calculated. The second part $\frac{\partial f_i}{\partial \boldsymbol{\alpha}_{k'}}$ requires a further application of the chain rule, which traces back from f_i to $\boldsymbol{\alpha}_{k'}$ through ξ_i, ψ_i, $l(d_{k_n}(\mathbf{x}_n))$, $d_{k_n}(\mathbf{x}_n)$, $\Pr(C_k|\mathbf{x})$, and $y_k(\mathbf{x})$.

$$\begin{aligned} \frac{\partial J}{\partial \boldsymbol{\alpha}_{k'}} &= \frac{\partial J_{\text{HM}}}{\partial \boldsymbol{\alpha}_{k'}} + \frac{\partial J_{\text{L2}}}{\partial \boldsymbol{\alpha}_{k'}} \qquad (28)\\ &= \sum_{i=1}^{N_{\text{ec}}}\frac{\partial J_{\text{HM}}}{\partial f_i}\frac{\partial f_i}{\partial \boldsymbol{\alpha}_{k'}} + \lambda\mathcal{K}\boldsymbol{\alpha}_{k'} \end{aligned}$$

The final result of differentiation $\frac{\partial J}{\partial \boldsymbol{\alpha}_{k'}}$ is shown as Eq. (29) in the next page. The values of the parameters are updated in each epoch of the retraining, following the rule

$$
\begin{aligned}
\frac{\partial J}{\partial \boldsymbol{\alpha}_{k'}} = \; & -2S_\gamma \frac{\gamma_1}{(f_1)^2}\left[\sum_{h=1}^{N_{\mathrm{ec}}}\frac{\gamma_h}{f_h}\right]^{-2}\sum_{n=1}^{N}\left[\psi_1^{-1}(-\delta_{k_n,2})-\xi_1\psi_1^{-2}\times(0)\right] \\
& \times\epsilon l(d_{k_n}(\mathbf{x}_n))(1-l(d_{k_n}(\mathbf{x}_n)))(-1)^{\delta_{k',k_n}}\Pr(C_{k_n}|\mathbf{x}_n)(1-\Pr(C_{k_n}|\mathbf{x}_n))\boldsymbol{\kappa}(\mathbf{x}_n) \\
& -2S_\gamma \frac{\gamma_2}{(f_2)^2}\left[\sum_{h=1}^{N_{\mathrm{ec}}}\frac{\gamma_h}{f_h}\right]^{-2}\sum_{n=1}^{N}\left[\psi_2^{-1}(-\delta_{k_n,1})-\xi_2\psi_2^{-2}\times(0)\right] \\
& \times\epsilon l(d_{k_n}(\mathbf{x}_n))(1-l(d_{k_n}(\mathbf{x}_n)))(-1)^{\delta_{k',k_n}}\Pr(C_{k_n}|\mathbf{x}_n)(1-\Pr(C_{k_n}|\mathbf{x}_n))\boldsymbol{\kappa}(\mathbf{x}_n) \\
& -2S_\gamma \frac{\gamma_3}{(f_3)^2}\left[\sum_{h=1}^{N_{\mathrm{ec}}}\frac{\gamma_h}{f_h}\right]^{-2}\sum_{n=1}^{N}\left[\psi_3^{-1}(-\delta_{k_n,2})-\xi_3\psi_3^{-2}(-\delta_{k_n,2}+\delta_{k_n,1})\right] \\
& \times\epsilon l(d_{k_n}(\mathbf{x}_n))(1-l(d_{k_n}(\mathbf{x}_n)))(-1)^{\delta_{k',k_n}}\Pr(C_{k_n}|\mathbf{x}_n)(1-\Pr(C_{k_n}|\mathbf{x}_n))\boldsymbol{\kappa}(\mathbf{x}_n) \\
& -2S_\gamma \frac{\gamma_4}{(f_4)^2}\left[\sum_{h=1}^{N_{\mathrm{ec}}}\frac{\gamma_h}{f_h}\right]^{-2}\sum_{n=1}^{N}\left[\psi_4^{-1}(-\delta_{k_n,1})-\xi_4\psi_4^{-2}(-\delta_{k_n,1}+\delta_{k_n,2})\right] \\
& \times\epsilon l(d_{k_n}(\mathbf{x}_n))(1-l(d_{k_n}(\mathbf{x}_n)))(-1)^{\delta_{k',k_n}}\Pr(C_{k_n}|\mathbf{x}_n)(1-\Pr(C_{k_n}|\mathbf{x}_n))\boldsymbol{\kappa}(\mathbf{x}_n) \\
& -2S_\gamma \frac{\gamma_5}{(f_5)^2}\left[\sum_{h=1}^{N_{\mathrm{ec}}}\frac{\gamma_h}{f_h}\right]^{-2}\sum_{n=1}^{N}\left[\psi_5^{-1}(-\delta_{k_n,2}-\delta_{k_n,1})-\xi_5\psi_5^{-2}\times(0)\right] \\
& \times\epsilon l(d_{k_n}(\mathbf{x}_n))(1-l(d_{k_n}(\mathbf{x}_n)))(-1)^{\delta_{k',k_n}}\Pr(C_{k_n}|\mathbf{x}_n)(1-\Pr(C_{k_n}|\mathbf{x}_n))\boldsymbol{\kappa}(\mathbf{x}_n) \\
& +\lambda\boldsymbol{\mathcal{K}}\boldsymbol{\alpha}_{k'}
\end{aligned}
\tag{29}
$$

defined in Eq. (30), where the learning rate ρ is a positive constant. The values of $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ are obtained for the negative and the positive classes, respectively.

$$
\boldsymbol{\alpha}_k \leftarrow \boldsymbol{\alpha}_k - \rho\frac{\partial J}{\partial \boldsymbol{\alpha}_{k'}}|_{\boldsymbol{\alpha}_{k'}=\boldsymbol{\alpha}_k} \tag{30}
$$

As is common in classifiers, including SVM, KLOGR, and CM-KLOGR, the bias term b_k has a considerable effect on the performance; this is especially true in the classification of imbalanced data. For SVM, it is not possible to set b_k in a dual space defined by kernelization. Instead, there is a way to do so in an original space using the values of $\alpha_{mk}\mathcal{K}(\mathbf{x},\mathbf{x}_m)$ [38]. In the methods based on LOGR including KLOGR, b_k is frequently called cutoff and treated as a separate parameter to set after training. b_k in CM-KLOGR is handled in the same manner.

4.3 Setting of Hyperparameters and Cutoff

In general, classifiers have two kinds of variables to set, hyperparameters and parameters, and if shifting is needed, a bias term (it can be a part of parameters, depending how to handle it). In contrast to that parameter setting is often discussed in details, hyperparameter and bias term settings are not so. It is difficult in nature to set hyperparameters in a fully systematic way; Hyperparameter setting inevitably includes trial-and-error procedures. However, considering its effect on performance, which may be strong when data is imbalanced, it is worth discussing how to design the setting of hyperparameters. Bias term setting affects classification performance, has widely different two approaches, and should be discussed, too.

It is common for classifiers to set the hyperparameters by performing a grid search using validation data that differs from the training data. In CM-KLOGR, the hyperparameters are the width of the Gaussian kernel σ, the weight on the regularization term λ, and the smoothness of the loss function ϵ, and they are set outside of training by this common way.

Regarding the setting of the bias term b_k, as mentioned briefly in Section 3.1, there are two alternatives according to whether b_k is included or not in the objective function. The first method embeds b_k as α_0 into the objective function by augmenting the Kernel matrix $\boldsymbol{\mathcal{K}}$ using a dummy kernel, and it optimizes b_k during training [51], [52], [53]. The second method fixes $b_k = 0$ during training, and it optimizes b_k after training [6], [7], [51], [54]. In the biomedical field, the bias term is often called cutoff, and it is set using the second method. To be precise, the bias term and the cutoff have differences in their roles and optimization. For this reason, in the present study they are differentiated as follows: The bias term is the intercept of an objective function, and it is set in training; the cutoff is the threshold of the discriminant functions, and it is set after training.

The second method is selected for CM-KLOGR, and it makes the objective function of CM-KLOGR similar to that of SVM. Besides being used for setting the hyperparameters, a grid search with validation data is used to set the cutoff. The classification decision is made by the rule that includes the cutoff in Eq. (31).

$$
C(\mathbf{x}) = \begin{cases} C_1, & \text{iff } \; g_2(\mathbf{x}) - g_1(\mathbf{x}) \leq \textit{cutoff} \\ C_2, & \text{iff } \; g_2(\mathbf{x}) - g_1(\mathbf{x}) > \textit{cutoff} \end{cases} \tag{31}
$$

where $C_k, k \in \{1, 2\}$ is a class (1 and 2 mean negative and positive, respectively), $g_k(\mathbf{x}) = \Pr(C_k|\mathbf{x})$ is the estimated posterior probability, and *cutoff* denotes the cutoff.

In order to accurately estimate classification performance, it is as important as the processes of hyperparameter, parameter, and cutoff setting, how to divide and feed a dataset into these processes, especially when data is imbalanced. The dividing and feeding should be designed not to change the nature of the imbalanced data. It is the part of designing experiments and discussed in Section 5.2.

5 Experiment I: Evaluation under Same Weights on Evaluation Criteria

5.1 Purpose and Conditions

CM-KLOGR was empirically evaluated by comparing its performance with those of competitive classifiers on several datasets. Experiment I assumed the strictest case in which both positives and negatives must be exhaustively and correctly classified, such as medical diagnosis with minimal errors. Therefore the HM of Sens, Spec, PPV, and NPV was used for training CM-KLOGR, validating all the classifiers, and testing them. Note that Acc was excluded due to its redundancy; the weights γ_i were set to 1 for Sens, Spec, PPV, and NPV, and to 0 for Acc.

KLOGR, SVM, and these methods combined with undersampling and oversampling were selected as competitors, in consideration of the following. What distinguishes CM-KLOGR is its comprehensive objective function, the harmonic mean of evaluation criteria, and its consistent learning process with no user intervention. Examining the effect of these aspects is the focal point, and we should avoid mixing the effect of difference in model structure into this examination. KLOGR is the basis of CM-KLOGR. It shares the same model structure with CM-KLOGR, and its objective function is that in the pretraining of CM-KLOGR. Therefore, it can be the baseline of performance. SVM is the most common kernel method, and it has a similar model structure to that of CM-KLOGR, except its hinge-loss-based objective function. Thus, KLOGR and SVM were selected.

Sampling methods, which are a preprocessing techniques rather than classifiers, were also used. We selected and combined the simple sampling methods (undersampling and oversampling) with KLOGR and SVM to compare to CM-KLOGR. Strategic sampling was not adopted, because, from the standpoint to clarify the effect of the objective function of CM-KLOGR, it will be out of focus to compare this effect and the effect of specific strategies in preprocessing. By the comparison of CM-KLOGR to the simple sampling methods, not only its effectiveness but also a perspective on the use of sampling for CM-KLOGR were examined.

The reasons why cost-sensitive and ensemble methods were not used is as below: As far as we know, many cost-sensitive methods require heuristic and task dependent processes, while CM-KLOGR does not. The present experiments did not aim to examine the effect of such processes, but to clarify the fundamental effectiveness of CM-KLOGR brought by its objective function. Ensemble methods are the way to combine classifiers, and hence they are not competitive to CM-KLOGR which is a single classifier. Note that we understand the importance of cost-sensitive and ensemble methods and also the necessity to compare and/or combine them with CM-KLOGR. After confirming the fundamental effectiveness of CM-KLOGR, such comparisons and/or combinations should be considered in the next stage of our study.

Regarding the other experimental conditions, the Gaussian kernel was used in common for the classifiers. For KLOGR and SVM combined with the sampling methods, to make the numbers of negatives and positives equal, the number of negatives was reduced by undersampling and that of positives was increased by oversampling. The imbalanced datasets summarized in Table 2 were used, that have different proportions of majority and minority (or negative and positive) instances [55], [56]. Prior to kernelization, features were normalized to be dimensionless with mean of 0 and standard deviation of 1 and then were augmented, in the original input variable space. This normalization is effective for hyperparameter and parameter settings, since it makes the search ranges close to each other among different datasets.

5.2 Evaluation Process Design

It is difficult to precisely evaluate the classification performance for imbalanced data, especially when the data is small. Because of the imbalance and the small number of instances, the distribution of instances often differs between the training, validation, and test sets. The difference in distribution makes performance evaluation imprecise, and consequently, it sometimes leads to improper setting. For this solution and the fair comparison of the classifiers, the following processes were designed to divide and feed datasets, to set the hyperparameters, parameters, and cutoff, and to estimate the performance.

TABLE 2
Specifications of benchmark datasets. Maj. and Min. denote the majority and minority classes, respectively.

Name of Datasets	Number of Features	Size Maj., Min. (Total)	Ratio Maj./Min.
Breast	10	458, 241 (699)	1.90
Haberman	3	225, 81 (306)	2.78
Ecoli-pp	7	284, 52 (336)	5.46
Ecoli-imu	7	301, 35 (336)	8.60
Pop_failures	18	494, 46 (540)	10.74
Yeast-1_vs_7	7	429, 30 (459)	14.30

TABLE 3
Search conditions for hyperparameter and cutoff setting. In SVM, c is a box constraint which has a similar role to λ.

Classifiers	Hyperparameters and Cutoff	
CM-KLOGR	σ:	0.1 to 5.0 with a step of 0.1
	λ:	0.1 to 5.0 with a step of 0.1
	ϵ:	1, 5, 10, 20, 40, and 80
	cutoff:	-1.0 to 1.0 with a step of 0.01
KLOGR	σ:	0.1 to 5.0 with a step of 0.1
	λ:	0.1 to 5.0 with a step of 0.1
	cutoff:	-1.0 to 1.0 with a step of 0.01
SVM	σ:	0.1 to 5.0 with a step of 0.1
	c:	0.1 to 5.0 with a step of 0.1
	cutoff:	-1.0 to 1.0 with a step of 0.01

TABLE 4
Experiment I: Performance 1 for CM-KLOGR, KLOGR, and SVM, where HM is the harmonic mean of Sens, Spec, PPV, and NPV.

Breast

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	95.83	95.65	92.00	97.78	95.27
KLOGR	100.00	95.65	92.31	100.00	**96.88**
SVM	100.00	93.48	88.89	100.00	95.36

Haberman

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	75.00	82.61	60.00	90.48	**75.25**
KLOGR	62.50	78.26	50.00	85.71	66.18
SVM	37.50	86.96	50.00	80.00	56.60

Ecoli-pp

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	93.10	71.43	100.00	**89.40**
KLOGR	100.00	93.10	71.43	100.00	**89.40**
SVM	100.00	86.21	55.56	100.00	80.64

Ecoli-imu

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	50.00	96.67	66.67	93.55	**71.38**
KLOGR	50.00	96.67	66.67	93.55	**71.38**
SVM	50.00	93.33	50.00	93.33	65.12

Pop_failures

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	95.92	71.43	100.00	**90.04**
KLOGR	80.00	97.96	80.00	97.96	88.07
SVM	80.00	95.92	66.67	97.92	83.09

Yeast-1_vs_7

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	88.37	37.50	100.00	**68.99**
KLOGR	100.00	81.40	27.27	100.00	58.01
SVM	66.67	93.02	40.00	97.56	65.57

TABLE 5
Experiment I: Performance 2 for CM-KLOGR, KLOGR, and SVM, where HM is the harmonic mean of Sens, Spec, PPV, and NPV.

Breast

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	95.83	100.00	100.00	97.87	**98.40**
KLOGR	100.00	95.65	92.31	100.00	96.88
SVM	100.00	95.65	92.31	100.00	96.88

Haberman

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	87.50	82.61	63.64	95.00	**80.36**
KLOGR	50.00	100.00	100.00	85.19	77.31
SVM	50.00	100.00	100.00	85.19	77.31

Ecoli-pp

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	100.00	100.00	100.00	**100.00**
KLOGR	100.00	96.55	83.33	100.00	94.43
SVM	100.00	100.00	100.00	100.00	**100.00**

Ecoli-imu

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	50.00	100.00	100.00	93.75	**78.95**
KLOGR	50.00	96.67	66.67	93.55	71.38
SVM	50.00	100.00	100.00	93.75	**78.95**

Pop_failures

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	95.92	71.43	100.00	90.04
KLOGR	80.00	97.96	80.00	97.96	88.07
SVM	80.00	100.00	100.00	98.00	**93.67**

Yeast-1_vs_7

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	100.00	100.00	100.00	**100.00**
KLOGR	66.67	100.00	100.00	97.73	88.43
SVM	100.00	51.16	12.50	100.00	33.46

T% of the instances in a dataset are set aside for testing, and the remaining $(100-T)$% of them are split into S subsets for training and validating by the S-hold cross-validation [9], [18]. In our experiments, $T = 10$ and $S = 10$. The S-hold cross-validation is applied for each of the hyperparameter and cutoff settings, through Steps 1, 2, and 3.

In Step 1 to set the hyperparameters and the parameters, a grid search is performed under a fixed cutoff at 0. The S-fold cross-validation is applied on each cross point on the grid, that corresponds to each hyperparameter setting. In each fold, after setting the parameters with a training set composed of the $S-1$ subsets, the performance is estimated with a validation set, that is, the remaining subset. As a result, the hyperparameters and the parameters are set to the values that achieved the maximum average performance over the S folds (this is called "validation performance"). In Step 2 to set the cutoff, a grid search is performed similarly to Step 1, under the best hyperparameter and parameter settings obtained in Step 1. In Step 3, for making the parameter setting robust, the classifier is retrained with the final training set composed of the merged S subsets, under the best hyperparameter and cutoff settings give by Steps 1 and 2. Finally, under the best hyperparameter, parameter, and cutoff settings, the generalized performance is estimated with the test set, the T% of the data (this is called "test performance").

In the experiments, the range and step size shown in Table 3 were used for the grid search in the setting steps. Note that the values of ϵ were determined by changing the smoothness of the loss function in Eq. (16) with a fixed step of angle. Classification is sensitive to the cutoff, and thus a finer search step was used. For setting the parameters (i.e., training), as is obvious, the objective function of a classifier was employed. For setting the hyperparameters and the cutoff (i.e., validation), HM, the harmonic mean of Sens, Spec, PPV, and NPV was used, to lead a classifier to the increase in all these evaluation criteria.

Even though the evaluation processes are designed carefully, the difference in distribution between the validation and test sets may still remain and make evaluation imprecise. To address this, two kinds of evaluation results were discussed. Performance 1: test performance obtained under the hyperparameter and cutoff settings which achieved the best validation performance. This is a reasonable estimate of generalization ability, but is possibly influenced by a nuisance factor, i.e., the distribution difference. Performance 2: ideal test performance under the hyperparameter and cutoff settings which achieved the best test performance. This is another reasonable estimate of generalization ability, representing an ideal situation when the distribution of instances is the same between the validation and test sets, and the hyperparameters and the cutoff are truly optimal.

5.3 Results and Discussion

5.3.1 Performance of CM-KLOGR compared to those of KLOGR and SVM

Table 4 shows Performance 1 (the test performance under the best settings of the hyperparameters and the cutoff based on validation performance). Table 5 shows Performance 2 (the ideal test performance under those based on test performance). These tables include the results of CM-KLOGR, KLOGR, and SVM only, for simplicity. The results of the sampling methods will be provided later. The best performances are indicated by bold font.

TABLE 6
Experiment I: Performance 1 for CM-KLOGR, KLOGR with under/oversampling (KLOGR-US and KLOGR-OS), and SVM with under/oversampling (SVM-US and SVM-OS), where HM is the harmonic mean of Sens, Spec, PPV, and NPV.

Breast

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	95.83	95.65	92.00	97.78	95.27
KLOGR-US	100.00	95.65	92.31	100.00	**96.88**
KLOGR-OS	100.00	95.65	92.31	100.00	**96.88**
SVM-US	100.00	91.30	85.71	100.00	93.85
SVM-OS	100.00	89.13	82.76	100.00	92.37

Haberman

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	75.00	82.61	60.00	90.48	**75.25**
KLOGR-US	100.00	00.00	25.81	50.00	00.00
KLOGR-OS	87.50	52.17	38.89	92.31	59.57
SVM-US	50.00	82.61	50.00	82.61	62.30
SVM-OS	50.00	78.26	44.44	81.82	59.26

Ecoli-pp

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	93.10	71.43	100.00	**89.40**
KLOGR-US	100.00	82.76	50.00	100.00	76.80
KLOGR-OS	100.00	89.66	62.50	100.00	84.83
SVM-US	100.00	93.10	71.43	100.00	**89.40**
SVM-OS	100.00	93.10	71.43	100.00	**89.40**

Ecoli-imu

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	50.00	96.67	66.67	93.55	**71.38**
KLOGR-US	75.00	76.67	30.00	95.83	57.02
KLOGR-OS	50.00	86.67	33.33	92.86	55.32
SVM-US	00.00	100.00	50.00	88.24	00.01
SVM-OS	50.00	93.33	50.00	93.33	65.12

Pop_failures

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	95.92	71.43	100.00	90.04
KLOGR-US	100.00	04.08	09.62	100.00	10.84
KLOGR-OS	80.00	97.96	80.00	97.96	88.07
SVM-US	100.00	97.96	83.33	100.00	**94.77**
SVM-OS	40.00	97.96	66.67	94.12	65.75

Yeast-1_vs_7

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	88.37	37.50	100.00	**68.99**
KLOGR-US	100.00	02.33	06.67	99.99	06.67
KLOGR-OS	100.00	74.42	21.43	100.00	49.93
SVM-US	00.00	100.00	50.00	93.48	00.01
SVM-OS	33.33	81.40	11.11	94.59	28.00

TABLE 7
Experiment I: Performance 2 for CM-KLOGR, KLOGR with under/oversampling (KLOGR-US and KLOGR-OS), and SVM with under/oversampling (SVM-US and SVM-OS), where HM is the harmonic mean of Sens, Spec, PPV, and NPV.

Breast

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	95.83	100.00	100.00	97.87	**98.40**
KLOGR-US	100.00	95.65	92.31	100.00	96.88
KLOGR-OS	100.00	95.65	92.31	100.00	96.88
SVM-US	100.00	95.65	92.31	100.00	96.88
SVM-OS	95.83	100.00	100.00	97.87	**98.40**

Haberman

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	87.50	82.61	63.64	95.00	**80.36**
KLOGR-US	87.50	73.91	53.85	94.44	73.91
KLOGR-OS	87.50	78.26	58.33	94.74	77.06
SVM-US	50.00	100.00	100.00	85.19	77.31
SVM-OS	50.00	100.00	100.00	85.19	77.31

Ecoli-pp

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	100.00	100.00	100.00	**100.00**
KLOGR-US	100.00	96.55	83.33	100.00	94.43
KLOGR-OS	100.00	93.10	71.43	100.00	89.40
SVM-US	100.00	96.55	83.33	100.00	94.43
SVM-OS	100.00	96.55	83.33	100.00	94.43

Ecoli-imu

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	50.00	100.00	100.00	93.75	**78.95**
KLOGR-US	50.00	100.00	100.00	93.75	**78.95**
KLOGR-OS	50.00	96.67	66.67	93.55	71.38
SVM-US	50.00	96.67	66.67	93.55	71.38
SVM-OS	50.00	100.00	100.00	93.75	**78.95**

Pop_failures

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	95.92	71.43	100.00	**90.04**
KLOGR-US	100.00	00.00	09.26	50.00	00.00
KLOGR-OS	100.00	93.88	62.50	100.00	85.74
SVM-US	00.00	100.00	50.00	90.74	00.01
SVM-OS	80.00	97.96	80.00	97.96	88.07

Yeast-1_vs_7

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	100.00	100.00	100.00	**100.00**
KLOGR-US	66.67	95.35	50.00	97.62	71.77
KLOGR-OS	100.00	97.67	75.00	100.00	91.80
SVM-US	100.00	88.37	37.50	100.00	68.99
SVM-OS	100.00	88.37	37.50	100.00	68.99

In Table 4, for HM which is the harmonic mean of Sens, Spec, PPV, and NPV, out of six datasets, CM-KLOGR achieved the best for five, KLOGR did so for three, and SVM for zero datasets, respectively. CM-KLOGR ranked best most frequently. A trend can be seen that CM-KLOGR worked better according to the increase in imbalance (refer the ratio of majority to minority given in Table 2). Concretely speaking, CM-KLOGR did not perform best under low imbalance, tied for first place with KLOGR under moderate imbalance, and performed best under high imbalance.

In Table 5, out of six datasets, CM-KLOGR achieved the best for five, KLOGR did so for zero, and SVM for three datasets, respectively. CM-KLOGR ranked best almost perfectly, and this suggests that CM-KLOGR has a higher potential to maximize its performance than KLOGR and SVM have. Summarizing the results in Tables 4 and 5, CM-KLOGR worked better than KLOGR and SVM.

5.3.2 *Performance of CM-KLOGR compared to those of KLOGR and SVM with the sampling methods*

We move onto the results obtained by CM-KLOGR, KLOGR with undersampling (KLOGR-US), KLOGR with oversampling (KLOGR-OS), SVM with undersampling (SVM-US), and SVM with oversampling (SVM-OS). Table 6 shows Performance 1 (the test performance), and Table 7 shows Performance 2 (the ideal test performance).

In Table 6, for HM which is the harmonic mean of Sens, Spec, PPV, and NPV, out of six datasets, CM-KLOGR achieved the best for four, and KLOGR-US, KLOGR-OS, SVM-US, and SVM-OS did so for one or two datasets, respectively. Although the number of wins was not so large, CM-KLOGR ranked best most frequently and was more stable in performance than the sampling methods were.

In Table 7, out of six datasets, CM-KLOGR achieved the best for six, and KLOGR-US, KLOGR-OS, SVM-US, and SVM-OS did so for zero to two datasets, respectively. CM-KLOGR ranked best perfectly, and this suggests that CM-KLOGR has a higher potential to maximize its performance than the sampling methods have. In summary of the results in Tables 6 and 7, CM-KLOGR worked better than the simple sampling methods. Remember that sampling is a kind of preprocessing, and CM-KLOGR is not exclusive to sampling. By the results that CM-KLOGR worked better as a whole, and the sampling methods were effective for some datasets, the positive perspective of combining CM-KLOGR with sampling was suggested.

TABLE 8
Experiment II: Performance 1 for CM-KLOGR, KLOGR, and SVM, where HM is the harmonic mean of Sens and PPV.

Breast

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	95.83	95.65	92.00	97.78	93.88
KLOGR	100.00	95.65	92.31	100.00	**96.00**
SVM	100.00	93.48	88.89	100.00	94.12

Haberman

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	75.00	82.61	60.00	90.48	**66.67**
KLOGR	62.50	78.26	50.00	85.71	55.56
SVM	62.50	73.91	45.45	85.00	52.63

Ecoli-pp

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	93.10	71.43	100.00	**83.33**
KLOGR	100.00	93.10	71.43	100.00	**83.33**
SVM	100.00	86.21	55.56	100.00	71.43

Ecoli-imu

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	50.00	96.67	66.67	93.55	**57.14**
KLOGR	50.00	96.67	66.67	93.55	**57.14**
SVM	50.00	93.33	50.00	93.33	50.00

Pop_failures

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	60.00	97.96	75.00	96.00	66.67
KLOGR	80.00	97.96	80.00	97.96	**80.00**
SVM	80.00	97.96	80.00	97.96	**80.00**

Yeast-1_vs_7

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	100.00	100.00	100.00	**100.00**
KLOGR	100.00	81.40	27.27	100.00	42.86
SVM	66.67	93.02	40.00	97.56	50.00

TABLE 9
Experiment II: Performance 2 for CM-KLOGR, KLOGR, and SVM, where HM is the harmonic mean of Sens and PPV.

Breast

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	95.83	100.00	100.00	97.87	**97.87**
KLOGR	100.00	95.65	92.31	100.00	96.00
SVM	100.00	95.65	92.31	100.00	96.00

Haberman

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	75.00	86.96	66.67	90.91	**70.59**
KLOGR	87.50	73.91	53.85	94.44	66.67
SVM	50.00	100.00	100.00	85.19	66.67

Ecoli-pp

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	100.00	100.00	100.00	**100.00**
KLOGR	100.00	96.55	83.33	100.00	90.91
SVM	100.00	100.00	100.00	100.00	**100.00**

Ecoli-imu

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	75.00	93.33	60.00	96.55	**66.67**
KLOGR	50.00	96.67	66.67	93.55	57.14
SVM	50.00	100.00	100.00	93.75	**66.67**

Pop_failures

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	95.92	71.43	100.00	83.33
KLOGR	80.00	97.96	80.00	97.96	80.00
SVM	80.00	100.00	100.00	98.00	**88.89**

Yeast-1_vs_7

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	100.00	100.00	100.00	**100.00**
KLOGR	66.67	100.00	100.00	97.73	80.00
SVM	66.67	93.02	40.00	97.56	50.00

5.3.3 Comprehensive Discussion

Based on the discussions in Sections 5.3.1 and 5.3.2, it can be concluded that CM-KLOGR outperformed its competitors (KLOGR and SVM with and without under/oversampling methods) in many conditions. CM-KLOGR worked well under the default of equal weights on the four evaluation criteria, and its good performance under different weights is expected, too. This is examined in Experiment II.

Having confirmed the effectiveness of CM-KLOGR, consider now its relation to the conventional imbalanced data classification methods, the cost-sensitive, sampling, and ensemble ones. CM-KLOGR can be interpreted to be upward compatible to cost-sensitive methods; the crucial difference is that CM-KLOGR directly increases the values of the evaluation criteria, with no subjective or trial-and-error cost setting. In principle, CM-KLOGR can work with not only sampling methods but also ensemble methods. It will be worthwhile to examine the performance of CM-KLOGR combined with sampling and/or ensemble methods for a possible further improvement.

6 Experiment II: Evaluation under Different Weights on Evaluation Criteria

6.1 Purpose and Conditions

In Experiment I, with a default setting to assign equal weights to each of the four evaluation criteria, CM-KLOGR increased the values of HM. It is easy for CM-KLOGR to assign different weights depending on the importance of the evaluation criteria, and thus Experiment II examines how CM-KLOGR works with different weights.

With regard to the weights, two types of cases were assumed. One was that positives had a considerably higher priority than negatives, and the overlooking and misrecognition of positives were penalized. For example, infected people must be detected, and uninfected people must be screened out. Sens and PPV were emphasized by assigning a weight of 1 on them and 0 on the other evaluation criteria. The HM of Sens and PPV (in other words, recall and precision), which is equal to F-measure [48], was used for training CM-KLOGR, validating all the classifiers, and testing all the classifiers. The other was a case in which both positives and negatives were prioritized, and the overlooking of positives and negatives were penalized, such as that infected and uninfected people must be detected. Sens and Spec were put emphasis, and HM was their harmonic mean with a weight of 1 on Sens and Spec and 0 on the others. The HM of Sens and Spec was used as well as the HM of Sens and PPV in the former case. The remaining conditions were the same as those of Experiment I.

6.2 Evaluation Process Design

For dividing and feeding datasets, setting the hyperparameters, parameters, and cutoff, and estimating the classification performance, the same processes in Experiment I were applied (See Section 5.2). In Steps 1, 2, and 3, the HM of Sens and PPV was used in the former case, and the HM of Sens and Spec was used in the latter case.

6.3 Results and Discussion

6.3.1 Performance of CM-KLOGR to raise Sens and PPV

The results obtained by CM-KLOGR, KLOGR, and SVM are provided and discussed in details, but because of space limitations, those by KLOGR and SVM with sampling are

TABLE 10
Experiment II: Performance 1 for CM-KLOGR, KLOGR, and SVM, where HM is the harmonic mean of Sens and Spec.

Breast

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	95.83	93.48	88.46	97.73	94.64
KLOGR	100.00	95.65	92.31	100.00	**97.78**
SVM	100.00	93.48	88.89	100.00	96.63

Haberman

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	75.00	82.61	60.00	90.48	**78.62**
KLOGR	75.00	78.26	54.55	90.00	76.60
SVM	50.00	78.26	44.44	81.82	61.02

Ecoli-pp

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	89.66	62.50	100.00	94.54
KLOGR	100.00	93.10	71.43	100.00	**96.43**
SVM	100.00	86.21	55.56	100.00	92.59

Ecoli-imu

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	50.00	90.00	40.00	93.10	64.29
KLOGR	75.00	93.33	60.00	96.55	**83.17**
SVM	75.00	80.00	33.33	96.00	77.42

Pop_failures

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	81.63	35.71	100.00	89.89
KLOGR	100.00	93.88	62.50	100.00	**96.84**
SVM	100.00	93.88	62.50	100.00	**96.84**

Yeast-1_vs_7

Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	88.37	37.50	100.00	**93.83**
KLOGR	100.00	81.40	27.27	100.00	89.74
SVM	100.00	81.40	27.27	100.00	89.74

TABLE 11
Experiment II: Performance 2 for CM-KLOGR, KLOGR, and SVM, where HM is the harmonic mean of Sens and Spec.

Breast

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	95.83	100.00	100.00	97.87	**97.87**
KLOGR	100.00	95.65	92.31	100.00	97.78
SVM	100.00	95.65	92.31	100.00	97.78

Haberman

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	87.50	82.61	63.64	95.00	**84.98**
KLOGR	62.50	86.96	62.50	86.96	72.73
SVM	87.50	78.26	58.33	94.74	82.62

Ecoli-pp

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	100.00	100.00	100.00	**100.00**
KLOGR	100.00	96.55	83.33	100.00	98.24
SVM	100.00	100.00	100.00	100.00	**100.00**

Ecoli-imu

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	83.33	44.44	100.00	**90.91**
KLOGR	50.00	96.67	66.67	93.55	65.91
SVM	100.00	70.00	30.77	100.00	82.35

Pop_failures

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	95.92	71.43	100.00	**97.92**
KLOGR	100.00	93.88	62.50	100.00	96.84
SVM	100.00	95.92	71.43	100.00	**97.92**

Yeast-1_vs_7

Ideal Test Performance [%]					
Classifiers	Sens	Spec	PPV	NPV	HM
CM-KLOGR	100.00	100.00	100.00	100.00	**100.00**
KLOGR	66.67	100.00	100.00	97.73	80.00
SVM	100.00	83.72	30.00	100.00	91.14

omitted and mentioned briefly. For CM-KLOGR, KLOGR, and SVM, Tables 8 and 9 show Performance 1 (the test performance) and Performance 2 (the ideal test performance), respectively.

In Table 8, for HM which is the harmonic mean of Sens and PPV, out of six datasets, CM-KLOGR achieved the best for four, KLOGR did so for four, and SVM for one datasets, respectively. CM-KLOGR and KLOGR ranked best most frequently, but in regard to the difference in numerical values, CM-KLOGR is better than KLOGR. In Table 9, CM-KLOGR achieved the best for five, KLOGR did so for zero, and SVM for three datasets, respectively. CM-KLOGR ranked best almost perfectly, and this suggests that CM-KLOGR has a higher potential to maximize its performance than KLOGR and SVM have.

Similar trends to those in Tables 8 and 9 appeared in the comparison to KLOGR and SVM with sampling; CM-KLOGR ranked best most frequently with respect to both Performances 1 and 2. Summarizing all the results, CM-KLOGR worked better than the other classifiers.

6.3.2 Performance of CM-KLOGR to raise Sens and Spec

Similarly to Section 6.3.1, the results obtained by CM-KLOGR, KLOGR, and SVM are mainly discussed here. In Table 10 on Performance 1, for HM which is the harmonic mean of Sens and Spec, out of six datasets, CM-KLOGR achieved the best for two, KLOGR did so for four, and SVM for one datasets, respectively. KLOGR ranked best most frequently, followed by CM-KLOGR. For CM-KLOGR, its clear superiority and trend did not appear. In Table 11 on Performance 2, CM-KLOGR achieved the best for six, KLOGR did so for zero, and SVM for two datasets, respectively. CM-KLOGR ranked best perfectly, and this suggests that CM-KLOGR has a higher potential to maximize its performance than KLOGR and SVM have.

The results by KLOGR and SVM with sampling were not provided to save space, but let us note that, compared to them, CM-KLOGR ranked best most frequently regarding both Performances 1 and 2. In summary of all the results, although CM-KLOGR could not outperform KLOGR in Table 10, the high potential of CM-KLOGR was suggested by its superiority in Table 11 and that to KLOGR and SVM with sampling.

6.3.3 Comprehensive Discussion

As discussed in Sections 6.3.1 and 6.3.2, CM-KLOGR outperformed its competitors (KLOGR and SVM with and without under/oversampling methods) in many conditions. Specifically speaking, CM-KLOGR was more effective to raise the harmonic mean of Sens and PPV, namely F-measure, and that of Sens and Spec. Considering the results of Experiments I and II together, CM-KLOGR worked effectively and flexibly depending on the prioritized evaluation criteria, such as all of Sens, Spec, PPV, and NPV, or two of them (Sens and PPV, or Sens and Spec).

7 Conclusions

We proposed an imbalanced data classifier, the confusion-matrix-based kernel logistic regression (CM-KLOGR). CM-KLOGR aims to directly increase the harmonic mean of evaluation criteria derived from a confusion matrix (sensitivity, specificity, positive predictive value, and negative predictive value), through a consistent learning process realized by KLOGR and minimum classification error and generalized probabilistic descent (MCE/GPD) learning. In

the experiments, CM-KLOGR outperformed KLOGR and support vector machine (SVM) with or without sampling, for several datasets under different settings of the weights on the evaluation criteria. It was confirmed that CM-KLOGR can increase the values of the evaluation criteria in a well-balanced way, adaptively to their priorities.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number 15K00323 and a MEXT-supported Program for the Strategic Research Foundation at Private Universities 2014-2018.

REFERENCES

[1] H. He and E. A. Garcia, "Learning from Imbalanced Data," IEEE Trans. on Knowledge and Data Engineering, vol.21, no.9, pp.1263–1284, 2009.

[2] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-based Approaches," IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol.42, no.4, pp.463–484, 2011.

[3] V. Lopez, A. Fernandez, S. Garcia, V. Palade, F. Herrera, "An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics," Information Sciences, vol.250, no.20, pp.113–141, 2013.

[4] A. Purwar and S. K. Singh, "Issues in Data Mining: A Comprehensive Survey," IEEE Int'l Conf. on Computational Intelligence and Computing Research, doi: 10.1109ICCIC.2014.7238447, 2014.

[5] P. Branco, L. Torgo, and R. P. Ribeiro, "A Survey of Predictive Modeling on Imbalanced Domains," ACM Computing Surveys, vol.49, no.2, article 31, 2016.

[6] V. Roth, "Probabilistic Discriminative Kernel Classifiers for Multi-class Problems," Lecture Notes in Computer Science, vol.2191, pp.246–253, 2001.

[7] J. Zhu and T. Hastie, "Kernel Logistic Regression and the Import Vector Machine," J. of Computational and Graphical Statistics, vol.14, no.1, pp.185–205, 2005.

[8] S. Katagiri, B. H. Juang, and C.-H. Lee, "Pattern Recognition using a Family of Design Algorithms based upon the Generalized Probabilistic Descent Method," Proceedings of the IEEE, vol.86, no.11, pp. 2345–2373, 1998.

[9] C. M. Bishop, "Pattern Recognition and Machine Learning," Springer, 2006.

[10] V. N. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. of Artificial Intelligence Research, pp.321–357, 2002.

[11] H. He, Y. B. Edwardo, A. Garcia, and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," IEEE Int'l Joint Conf. on Neural Networks IJCNN-2008, pp.1322–1328, 2008.

[12] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs Modeling for Highly Imbalanced Classification," IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics, vol.39, no.1, pp.281–288, 2009.

[13] C. Seiffert, T. M. Khoshgoftaar, J. V. Hulse, and A. Napolitano, "RUSBoost: A Hybrid Approach to Alleviating Class Imbalance," IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans, vol.40, no.1, pp.185–197, 2010.

[14] B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos, "Class Imbalance, Redux," IEEE Int'l Conf. on Data Mining ICDM-2011, pp.754–763, 2011.

[15] S. Wang, Z. Li, W. Chao, and Q. Cao, "Applying Adaptive Over-sampling Technique Based on Data Density and Cost-Sensitive SVM to Imbalanced Learning," IEEE Int'l Joint Conf. on Neural Networks IJCNN-2012, doi: 10.1109IJCNN.2012.6252696, 2012.

[16] P. Yang, P. D. Yoo, J. Fernando, B. B. Zhou, Z. Zhang, and A. Y. Zomaya, "Sample Subset Optimization Techniques for Imbalanced and Ensemble Learning Problems in Bioinformatics Applications," IEEE Trans. on Cybernetics, vol.44, no.3, pp.445–455, 2014.

[17] B. Das, N. C. Krishnan, and D. J. Cook, "RACOG and wRACOG: Two Probabilistic Oversampling Techniques," IEEE Trans. on Knowledge and Data Engineering, vol.27, no.1, pp.222–234, 2015.

[18] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction, The Second Edition," Springer, 2014.

[19] X.-Y. Liu and Z.-H. Zhou, "The Influence of Class Imbalance on Cost-Sensitive Learning: An Empirical Study," IEEE Int'l Conf. on Data Mining ICDM-2006, pp.970–974, 2006.

[20] C. L. Castro and A. P. Braga, "Novel Cost-Sensitive Approach to Improve the Multilayer Perceptron Performance on Imbalanced Data," IEEE Trans. on Neural Networks and Learning Systems, vol.24, no.6, pp.888–899, 2013.

[21] B. Krawczyk, "Cost-Sensitive One-vs-One Ensemble for Multi-Class Imbalanced Data," IEEE Int'l Joint Conf. on Neural Networks IJCNN-2016, doi: 10.1109IJCNN.2016.7727503, 2016.

[22] C. Zhang, K. C. Tan, and R. Ren, "Training Cost-sensitive Deep Belief Networks on Imbalance Data Problems," IEEE Int'l Joint Conf. on Neural Networks IJCNN-2016, doi: 10.1109IJCNN.2016.7727769, 2016.

[23] V. Nilulin, G. J. McLachlan, and S. K. Ng, "Ensemble Approach for the Classification of Imbalanced Data," Lecture Notes in Artificial Intelligence, vol.5866, pp.291–300, 2009.

[24] S. Wang and X. Yao, "Relationships between Diversity of Classification Ensembles and Single-Class Performance Measures," IEEE Trans. on Knowledge and Data Engineering, vol.25, no.1, pp.206–2019, 2013.

[25] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, "A Novel Ensemble Method for Classifying Imbalanced Data," Pattern Recognition, vol.48, pp.1623–1637, 2015.

[26] R. E. Schapire, "The Strength of Weak Learnability," Machine Learning, vol.5, issue 2, pp.197–227, 1990.

[27] D. H. Wolpert, "Stacked Generalization," Neural Networks, vol.5, pp.241–259, 1992.

[28] L. Breiman, "Bagging Predictors," Machine Learning, vol.24, issue 2, pp.123–140, 1996.

[29] L. Breiman, "Random Forests," Machine Learning, vol.45, issue 1, pp.5–32, 2001.

[30] I. Nouretdinov, S. Costafreda-Gonzalez, A. Gammerman, A. Chervonenkis, V. Vovk, V. Vapnik, and C. H. Y. Fu, "Machine Learning Classification with Confidence: Application of Transductive Conformal Predictors to MRI-based Diagnostic and Prognostic Markers in Depression," NeuroImage, vol.56, no.2, pp.809–813, 2011.

[31] V. Balasubramanian, S. S. Ho, and V. Vovk, "Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications," Elsevier, 2014.

[32] K. Matsuda, M. Ohsaki, S. Katagiri, H. Yokoi, and K. Takabayashi, "Application of Kernel Logistic Regression to the Prediction of Liver Fibrosis Stages in Chronic Hepatitis C," Joint Int'l Conf. on Soft Computing and Intelligent Systems and Int'l Symposium on Advanced Intelligent Systems, SCIS-ISIS2012, pp.780–784, 2012.

[33] Y. Fong, S. Datta, I. S. Georgiev, P. D. Kwnong, and G. D. Tomaras, "Kernel-based Logistic Regression Model for Protein Sequence without Vectorialization," Biostatistics, vol.16, no.3, pp.480–492, 2015.

[34] X. Wang, E. P. Xing, and D. J. Schaid, "Kernel Methods for Large-scale Genomic Data Analysis," Briefings in Bioinformatics, vol.16, no.2, pp.183–192, 2015.

[35] Z. Liu and J. Hu, "Mislocalization-related Disease Gene Discovery Using Gene Expression Based Computational Protein Localization Prediction," Methods, vol.93, no.15, pp.119–127, 2016.

[36] A. Nath and S. Karthikeyan, "Enhanced Prediction and Characterization of CDK Inhibitors Using Optimal Class Distribution," Interdisciplinary Sciences: Computational Life Sciences, doi: 10.1007s12539-016-0151-1, 2016.

[37] C. Cortes and V. Vapnik, "Support-vector Networks," Machine Learning, vol.20, pp.273–297, 1995.

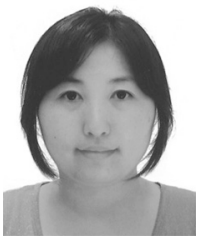
[38] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods," Cambridge University Press, 2000.

[39] S. S. Keerthi, V. Sindhwani, and O. Chapelle, "An Efficient Method for Gradient-based Adaptation of Hyperparameters in SVM Models," Neural Information Processing Systems Conf. NIPS-2006, pp.673–680, 2006.

[40] B. H. Juang, W. Chou, and C. H. Lee, "Minimum Classification Error Rate Methods for Speech Recognition," IEEE Trans. on Speech and Audio Processing, vol.5, no.3, pp.257–265, 1997.

[41] E. McDermott, T. J. Hazen, J. L. Roux, A. Nakamura, S. Katagiri, "Discriminative Training for Large-vocabulary Speech Recogni-

tion Using Minimum Classification Error," IEEE Trans. on Audio, Speech, and Language Processing, vol.15, no.1, pp.203–223, 2007.
[42] X. He, L. Deng, and W. Chou, "Discriminative Learning in Sequential Pattern Recognition," IEEE Signal Processing Magazine, vol.25, no.5, pp.14–36, 2008.
[43] R. Salakhutdinov and G. Hinton, "Deep Boltzmann Machines," Int'l Conf. on Artificial Intelligence and Statistics AISTATS-2009, vol.5, pp.448–455, 2009.
[44] D. G. Altman and J. M. Bland, "Diagnostic Tests 1: Sensitivity and Specificity," British Medical J., vol.308, no.6943, p.1552, 1994.
[45] D. G. Altman and J. M. Bland, "Diagnostic Tests 2: Predictive Values," British Medical J., vol.309, no.6947, p.102, 1994.
[46] D. G. Altman and J. M. Bland, "Diagnostic Tests 3: Receiver Operating Characteristic Plots," British Medical J., vol.309, no.6948, p.188, 1994.
[47] US National Library of Medicine, National Institutes of Health, "PubMed," https://www.ncbi.nlm.nih.gov/pubmed, 2016.
[48] C. J. van Rijsbergen, "Information Retrieval," Butterworths, 1979.
[49] M. Ohsaki, K. Matsuda, P. Wang, S. Katagiri, and H. Watanabe, "Formulation of the Kernel Logistic Regression based on the Confusion Matrix," IEEE Congress on Evolutionary Computation CEC-2015, pp.2327–2334, 2015.
[50] M. Jansche, "Maximum Expected F-Measure Training of Logistic Regression Models," Conf. on Human Language Technology and Empirical Methods in Natural Language Processing HLT/EMNLP-2005, pp.692–699, 2005.
[51] T. Jaakkola and D. Haussler, "Probabilistic Kernel Regression Models," Conf. on Artificial Intelligence and Statistics AISTATS-1999, vol.126, pp.00–04, 1999.
[52] G. C. Cawley and N. L. C. Talbot, "Efficient Model Selection for Kernel Logistic Regression," IEEE Int'l Conf. on Pattern Recognition ICPR-2004, vol. 2, pp.439–442, 2004.
[53] K. Tanaka, T. Kurita, and T. Kawabe, "Selection of Import Vectors via Binary Particle Swarm Optimization and Cross-Validation for Kernel Logistic Regression," IEEE Int'l Joint Conf. on Neural Networks IJCNN-2007, pp.1037–1042, 2007.
[54] R. Memisevic, "Dual Optimization Conditional Probability Models," NIPS Workshop on Kernel Methods and Structured Domains, Technical Report, 2006.
[55] M. Lichman, UCI Machine Learning Repository, http://archive.ics.uci.edu/ml. University of California, School of Information and Computer Science.
[56] J. Alcala-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. Garcia, L. Sanchez, F. Herrera. "KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework," J. of Multiple-Valued Logic and Soft Computing, vol.17, no.2–3, pp.255–287, 2011.



Miho Ohsaki received B.E., M.E., and Dr. Eng. degrees from Kyushu Institute of Design (currently, Kyushu University) in 1994, 1996, and 1999, respectively. From 1999 to 2004, she was an assistant professor at Shizuoka University. From 2004, she started working at Doshisha University, and is now a professor there. Her research interests are machine learning and its application to biomedical data analysis. She is a member of IEEE, IPSJ, and JSAI.

Peng Wang received B.E. from Xidian University in 2011, and M.E. from both Xidian University and Doshisha University by the double degree program in 2014.

Kenji Matsuda received B.E. and M.E. from Doshisha University in 2011 and 2013, respectively. He works at Yahoo! Japan Corporation. He is a member of IEEE.

Shigeru Katagiri received his Dr. Eng. degree in information engineering from Tohoku University in 1982. From 1982 to 1986, he worked at the Electrical Communication Laboratories, Nippon Telegraph and Telephone Public Corporation (currently NTT). In 1986, he moved to the Advanced Telecommunications Research Institute International (ATR), and in 1999 he returned to the NTT Communication Science Laboratories. Since 2006, he has been with Doshisha University, where he is a professor of the Graduate School of Science and Engineering. He has played several roles in academic communities, including the Chair of IEEE James L. Flanagan Speech and Audio Processing Award committee, the Chair of IEEE Kansai section, and a member of the Science Council of Japan. Dr. Katagiri is an IEEE Fellow and an NTT R&D Fellow.

Hideyuki Watanabe received his Ph.D. degree from Hokkaido University in 1993. From 1993 to 2009, he worked for Advanced Telecommunications Research Institute International (ATR). From 2009 to 2016, he worked for National Institute of Information and Communications Technology (NICT). From 2016, he works for ATR again. His current research interests include studies on pattern recognition theory, discriminative training, and speech signal processing. He is a member of the Acoustic Society of Japan, IEICE, and IEEE.

Anca Ralescu received a bachelor degree in mathematics from University of Bucharest in 1972, and received MA and PhD in mathematics from Indiana University in 1981 and 1983, respectively. She is currently a professor of Computer Science in the Department of Electrical Engineering and Computing Systems, University of Cincinnati. Her research interests are in the area of intelligent systems, including machine learning, knowledge representation, brain computer interface, management of uncertainty using probabilistic and fuzzy sets approaches. She is a senior member of IEEE.