Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 9911871, 11 pages
https://doi.org/10.1155/2021/9911871

Research Article

Multi-Nyström Method Based on Multiple Kernel Learning for Large Scale Imbalanced Classification

Ling Wang, Hongqiao Wang, and Guangyuan Fu

Department of Information Engineering, Rocket Force University of Engineering, Xi'an, 710025, China

Correspondence should be addressed to Hongqiao Wang; whq05@mails.tsinghua.edu.cn

Received 9 March 2021; Accepted 27 May 2021; Published 14 June 2021

Academic Editor: Cornelio Yáñez-Márquez

Extensions of kernel methods for the class imbalance problems have been extensively studied. Although they work well in coping with nonlinear problems, the high computation and memory costs severely limit their application to real-world imbalanced tasks. The Nyström method is an effective technique to scale kernel methods. However, the standard Nyström method needs to sample a sufficiently large number of landmark points to ensure an accurate approximation, which seriously affects its efficiency. In this study, we propose a multi-Nyström method based on mixtures of Nyström approximations to avoid the explosion of subkernel matrix, whereas the optimization to mixture weights is embedded into the model training process by multiple kernel learning (MKL) algorithms to yield more accurate low-rank approximation. Moreover, we select subsets of landmark points according to the imbalance distribution to reduce the model's sensitivity to skewness. We also provide a kernel stability analysis of our method and show that the model solution error is bounded by weighted approximate errors, which can help us improve the learning process. Extensive experiments on several large scale datasets show that our method can achieve a higher classification accuracy and a dramatical speedup of MKL algorithms.

1. Introduction

Real-world problems in computer vision [1], natural language processing [2, 3], and data mining [4, 5] present imbalanced traits in their data, which may be developed by the inherent properties of the data or some external factors such as sampling bias or measurement error. Unfortunately, most traditional learning algorithms are designed based on balanced data and target the overall classification accuracy, leading the minority class to be overwhelmed by the majority class. However, the minority class in these real-world problems is usually more important and expensive than the majority class.

In the past few decades, many algorithms have been proposed to solve the class imbalance problems [6–8]. The data-level methods artificially balance the skewed class distributions by data sampling [9, 10]. The algorithm-level methods lift the importance of minority instances via the modification of existing learners [11, 12]. However, there usually exist complex nonlinear structures in these real-world imbalanced data. In this case, the extensions of kernel methods for the class imbalance problems have been proven very effective [13–15]. In [16], Mathew et al. overcome the limitations of the synthetic minority oversampling technique (SMOTE) for nonlinear problems by oversampling in the feature space of the support vector machine. In [17], a kernel boundary alignment algorithm is proposed to adjust the class boundary by modifying the kernel matrix according to the imbalanced data distribution. The kernel-based adaptive synthetic data generation (KernelADASYN) for imbalanced learning is proposed in [18], which uses kernel density estimation (KDE) to estimate the adaptive oversampling density. However, with the development of data storage and data acquisition equipment, the scale of data continues to grow. The existing kernel-based class imbalanced learning (kernel CIL) methods suffer from serious challenges that the cost of calculating and storing a vast kernel matrix is very expensive.

A general technique for making kernel methods scalable is kernel approximation, of which the Nyström method is the most popular one [19]. The Nyström method constructs a low-rank approximation of the original kernel matrix from a subset of $l \ll n$ landmark points, where n is the data size. Computationally, it only needs to decompose a smaller matrix (denoted as $W \in \mathbb{R}^{l \times l}$). However, according to the approximation error bound $O(n/\sqrt{l}\,)$ for the Nyström method in [20], there is a trade-off between accuracy and efficiency. The more landmark points sampled provide improved approximation accuracy but require more computing resources, which results in the rapid expansion of the subkernel matrix W as the data size increases and seriously affects the efficiency of the Nyström method.

Some works study the efficacy of a variety of fixed and adaptive sampling schemes for the Nyström method. For example, Musco et al. presented a new Nyström algorithm based on recursive leverage score sampling, which runs in linear time in the number of training points [21]. An ensemble Nyström method has been proposed to yield more accurate low-rank approximations by running mixtures of the Nyström method based on several subsets of landmark points randomly sampled [22]. However, the mixture weights of the ensemble Nyström method are defined according to the approximation error of each Nyström approximation, which may lead to the performance not as expected when applied to practical classification or regression applications. Recently, there emerges a fast and accurate refined Nyström-based kernel classifier to improve the performance of the Nyström-based kernel classifier [23]. Although the Nyström method has been studied extensively, there still exists a potentially large gap between the performance of learner learned with the Nyström approximation and that learned with the original kernel.

In this study, we propose a novel method, multi-Nyström, for large scale imbalanced classification. We incorporate the multi-Nyström method and multiple kernel learning to learn an improved low-rank approximation kernel superior to any one of each multi-Nyström approximation, where each approximation is defined by different kernel functions and subsets of landmark points. Moreover, unlike existing sampling schemes for the multi-Nyström method, our method selects subsets of landmark points according to the imbalance distribution to deal with the problem of skewed data. Without computing and storing the full kernel matrix, our method can scale to large scale scenarios. The main contributions of this study are summarized as follows:

(1) We propose a multi-Nyström method to overcome the computational constraints of the Nyström method. Due to our method parallelized easily, it can generate more accurate approximates in large scale scenarios.

(2) We optimize the mixture weights according to the data and the problem at the hand, so that the combined approximation kernel matrix can produce better performance. Moreover, the low-rank approximation can significantly speed up the existing MKL algorithms process.

(3) We provide a stability analysis of our method, showing us the impact of kernel approximation error on the model solution and help determine the acceptable approximation error in the approximation of the kernel matrix.

The rest of this study is organized as follows. Section 2 introduces some related concepts. Section 3 then describes the proposed multi-Nyström approximation algorithm in detail. Experimental results and analysis compared with other algorithms are presented in Section 4. Finally, Section 5 summarizes the full work.

2. Related Work

2.1. Kernel Methods. Kernel methods such as support vector machines (SVMs) have become one of the most popular technologies of machine learning [24]. It can extend linear learners to nonlinear cases by introducing kernel trick. Consider a binary-class dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^s$ denotes an s-dimensional vector and $y_i \in \{+1, -1\}$ denotes its label. Define a nonlinear descriptor as

$$\Phi\colon \mathcal{X} \longrightarrow \mathcal{H} \quad \mathbf{x}_i \mapsto \Phi(\mathbf{x}_i). \tag{1}$$

The input data are mapped to a high-dimensional or even infinite-dimensional feature space, and the inner product in the feature space is calculated implicitly through the kernel function defined in the input space.

$$K(\mathbf{x}, x') = \langle \Phi(\mathbf{x}), \Phi(x') \rangle_{\mathcal{H}} = \Phi(\mathbf{x})^T \Phi(x'), \tag{2}$$

where $K\colon \mathbb{R}^s \times \mathbb{R}^s \mapsto \mathbb{R}$ is the kernel function that satisfies Mercer's theorem [25], and $\mathcal{H}$ is the corresponding reproducing kernel Hilbert space (RKHS). K can simply be a classical kernel like the radial basis function (RBF) kernel. Unfortunately, the kernel matrix $K \in \mathbb{R}^{n \times n}$ expands quadratically with the increase of data scale. The poor scalability limits the applicability of kernel methods in large scale scenarios.

2.2. Multiple Kernel Learning. Due to different kernels corresponding to different similarity concepts or using features from different views, MKL can obtain more complete representations of the input data by combining multiple kernels. In MKL, each instance $(\mathbf{x}_i, y_i)$ is mapped into different feature spaces by a series of descriptors [26]:

$$\Phi_{\mathcal{H}}(\mathbf{x}_i) = \left[\sqrt{d_1}\,\Phi_1^T(\mathbf{x}_i^1), \ldots, \sqrt{d_M}\,\Phi_M^T(\mathbf{x}_i^M)\right]^T, \tag{3}$$

where $\mathbf{x}_i^m$ represents feature from the m^{th} view of instance $\mathbf{x}_i$, $d_m \geq 0$, $m = 1, \ldots, M$ is the corresponding weight, and M is the total number of predefined kernels. Then, substitute any dot product term with kernels:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}_i) &= \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_i) \rangle_{\mathscr{H}} \\ &= \sum_{m=1}^{M} d_m \langle \Phi_m(\mathbf{x}^m), \Phi_m(\mathbf{x}_i^m) \rangle_{\mathscr{H}_m} \\ &= \sum_{m=1}^{M} d_m K_m(\mathbf{x}^m, \mathbf{x}_i^m), \end{aligned} \tag{4}$$

where each base kernel function $K_m(\cdot,\cdot)\colon \mathbb{R}^s \times \mathbb{R}^s \longrightarrow \mathbb{R}$ is a positive definite kernel associated with an RKHS $\mathscr{H}_m$. The purpose of MKL is to learn a resulting discriminant function of the form $f(\mathbf{x}) = \sum_m f_m(\mathbf{x}^m) + b$ with $\mathscr{H}_m \coloneqq \{f_m | f_m(\mathbf{x}) = \sum_{i=1}^{\infty} \alpha_i K_m(\mathbf{x}^m, \mathbf{x}_i^m)\}$.

Based on the aforementioned definition, the seminal work in MKL proposes the following structural risk minimization framework as MKL primal problem with kernel weights on a simplex [27].

$$\begin{aligned} &\min_{\{f_m\},b,\xi,d} : \frac{1}{2} \sum_{m=1}^{M} \frac{1}{d_m} \|f_m\|_{\mathscr{H}_m}^2 + C \sum_{i=1}^{N} \xi_i \\ &\qquad y_i \left(\sum_{m=1}^{M} f_m(\mathbf{x}_i^m) + b \right) \geq 1 - \xi_i \\ &\text{s.t.} \quad \xi_i \geq 0,\ i = 1, \ldots, N \\ &\qquad \sum_{m=1}^{M} d_m = 1,\ d_m \geq 0, m = 1, \ldots, M, \end{aligned} \tag{5}$$

where C is the regularization parameter of the error term. ξ is the slack variable. The L1-norm constraint on the weight vector d enforces the kernel combination to be sparse. We assume $\|f_m\|_{\mathscr{H}_m}^2 = 0$ whenever $d_m = 0$ in order to reach a finite objective. That implies if the weight of a certain kernel reaches $d_m = 0$, stop the optimization of f_m since the solution is known $f_m = 0$ [28].

Although MKL is an ideal candidate for combining multiview data, scalability is a key issue for MKL: (1) the computation and memory costs for maintaining several kernel matrices are heavy and (2) the computational efficiency of MKL solvers is not high.

2.3. Standard Nyström Method. Let $L = \{\mathbf{c}_1, \ldots, \mathbf{c}_l\}$, where $\mathbf{c}_i \in \mathbb{R}^s$ denotes a set of l landmark points randomly selected from D uniformly without replacement, $C \in \mathbb{R}^{n \times l}$ denotes the subkernel matrix between all instances and the landmark points, and $W \in \mathbb{R}^{l \times l}$ be a symmetric positive semidefinite (SPSD) subkernel matrix among the points in L. Then, the Nyström method uses W and C to generate a rank-k approximation $\widetilde{\mathrm{K}}_k$ of kernel matrix K for $k \leq l$ [20]:

$$K \approx \widetilde{K}_k \coloneqq C W_k^+ C^T, \tag{6}$$

where $W_k \in \mathbb{R}^{l \times l}$ is the best rank-k approximation to W with respect to the Frobenius norm, that is, $W_k = \operatorname{argmin}_{\operatorname{rank}(V)=k} \|W - V\|_F$, and W_k^+ denotes the pseudoinverse of W_k. Given the matrix W_k, the feature of each instance $\mathbf{x}_i$ can be evaluated as

$$\phi(\mathbf{x}_i) = \sqrt{W_k^+} \left(K(\mathbf{x}_i, \mathbf{c}_1), \ldots, K(\mathbf{x}_i, \mathbf{c}_l) \right)^T. \tag{7}$$

Calculate the singular value decomposition (SVD) of W as $W = U \Lambda U^T$, where U is the orthonormal and $\Lambda = \operatorname{diag}(\sigma_1, \ldots, \sigma_m)$ is the diagonal with $\sigma_1 \geq \cdots \geq \sigma_m \geq 0$. Then, the final approximate decomposition of K is denoted as the following form:

$$K \approx \widetilde{U}_k \widetilde{\Lambda}_k \widetilde{U}_k^T, \quad \text{with}\, \widetilde{U}_k = \sqrt{\frac{l}{n}} C U_k \Lambda_k^{-1}, \widetilde{\Lambda}_k = \frac{n}{l} \Lambda_k, \tag{8}$$

where $\Lambda_k \in \mathbb{R}^{k \times k}$ is the diagonal formed by the top k singular values of Λ, and $U_k \in \mathbb{R}^{l \times k}$ is formed by the associated singular vectors.

The total time complexity of the Nyström method is $O(l^3 + nlk)$ including $O(l^3)$ for SVD on W and $O(nlk)$ for matrix multiplication with C [29]. For $l \ll n$, it is much lower than the $O(n^3)$ complexity taken by SVD on K.

3. Proposed Algorithms

3.1. Multi-Nyström Method. We divide the imbalance dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ into the minority class set $D^+ = \{(\mathbf{x}_i, +1)\}_{i=1}^{n_+}$ and the majority class set $D^- = \{(\mathbf{x}_i, -1)\}_{i=1}^{n_-}$. When there are irregularities in the imbalanced data (such as small disjuncts, overlapping, and noise [30]) and the data scale is large, applying a single kernel may make the model biased, skew, or misleading. Inspired by the MKL algorithm [31], we construct a low rank approximate multiple kernel framework as follows:

$$K(\mathbf{x}, \mathbf{x}_i) \approx \sum_{m=1}^{M} d_m \widetilde{K}_{m,k}(\mathbf{x}^m, \mathbf{x}_i^m), \quad \text{with}\, d_m \geq 0, \tag{9}$$

where $\widetilde{K}_{m,k}$ corresponds to the rank-k approximation of each base kernel matrix K_m, and d_m is the corresponding mixture weight. As for the Nyström method, a key aspect is the sampling scheme [32]. For reducing the sensitivity to skewness in data, we adopt the stratified undersampling of the majority class to select M subsets of landmark points written as $L = \{L_m\}_{m=1}^M$ with each $L_m = \{\mathbf{c}_{m,1}, \ldots, \mathbf{c}_{m,l}\}$. The subkernel matrix between all instances and the landmark points can be expressed as

$$C = [C_1, \ldots, C_M] \in \mathbb{R}^{n \times Ml}, \tag{10}$$

where $C_m \in \mathbb{R}^{n \times l}$. Then, we perform the standard Nyström method on each C_m independently to get a rank-k approximation $\widetilde{K}_{m,k} = C_m W_{m,k}^+ C_m^T$ of each base kernel matrix K_m. Finally, by linearly combining these approximations, we can get the general form of approximation multiple kernel $\widetilde{K}$:

$$\widetilde{K} = [C_1, \ldots, C_M] \begin{bmatrix} d_1 W_{1,k}^+ & & \\ & \ddots & \\ & & d_M W_{M,k}^+ \end{bmatrix} \begin{bmatrix} C_1^T \\ \vdots \\ C_M^T \end{bmatrix}. \tag{11}$$

Given the mixture weight d_m, the feature of each instance $\mathbf{x}_i$ can be evaluated as

$$\tilde{\phi}(x_i) = \begin{bmatrix} \sqrt{d_1 W_{1,k}^{+}}\left(K_1(\mathbf{x}_i, \mathbf{c}_{1,1}), \ldots, K_1(\mathbf{x}_i, \mathbf{c}_{1,l})\right)^T \\ \vdots \\ \sqrt{d_M W_{M,k}^{+}}\left(K_M(\mathbf{x}_i, \mathbf{c}_{M,1}), \ldots, K_M(\mathbf{x}_i, \mathbf{c}_{M,l})\right)^T \end{bmatrix}. \tag{12}$$

Similarly, for the convenience of subsequent calculations, formula (11) can be rewritten as

$$\tilde{K} = \left[\tilde{U}_{1,k}, \ldots, \tilde{U}_{M,k}\right] \begin{bmatrix} d_1 \tilde{\Lambda}_{1,k} & & \\ & \ddots & \\ & & d_M \tilde{\Lambda}_{M,k} \end{bmatrix} \begin{bmatrix} \tilde{U}_{1,k}^T \\ \vdots \\ \tilde{U}_{M,k}^T \end{bmatrix}. \tag{13}$$

where $\tilde{U}_{m,k} \in \mathbb{R}^{n\times k}$, and $\tilde{\Lambda}_{m,k} \in \mathbb{R}^{k\times k}$ denotes the approximate decomposition of K_m obtained by (8). Figure 1 shows the proposed multi-Nyström method and includes an optimization process of the mixture weights detailed futher in next subsection.

When the mixture weight d_m is fixed or known, the total time complexity of the multi-Nyström method is $O(Ml^3 + Mnlk)$. Although our method requires M times more CPU resources than the standard Nyström method, $M \ll n$ is typically $O(1)$ for large scale data, and our method can compute in parallel in the distributed computing environment. Moreover, the SVD on the subkernel matrix W is decomposed into that on M much smaller matrices would also accelerate the calculation process.

3.2. Optimization to Mixture Weights. The purpose of MKL is to learn an optimal convex combination of a series of kernels during training. Based on the aforementioned definition, we propose an approximate multiple kernel learning framework for large scale imbalanced classification by modifying the original MKL framework in [26]

$$\min J(\mathbf{d}) \text{ such that } \|\mathbf{d}\|_1^2 = 1, \quad d_m \geq 0, m = 1, \ldots, M, \tag{14}$$

where

$$J(\mathbf{d}) = \begin{cases} \min\limits_{\boldsymbol{\alpha}} & \dfrac{1}{2}\boldsymbol{\alpha}^T Y \tilde{K} Y \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha} \\ & \mathbf{y}^T \boldsymbol{\alpha} = 0 \\ s.\text{t.} & \\ & 0 \leq \alpha \leq C, \end{cases} \tag{15}$$

where α is the Lagrange multipliers vector, and $Y = \operatorname{diag}(y_1, \ldots, y_n)$. To avoid numerical instability caused by ill-conditioning [19], we substitute $\tilde{K}_{m,k} \longleftarrow \tilde{K}_{m,k} + \sigma I$, where σ is a small positive constant called jitter factor. Moreover, to calculate the inverse of the approximate matrix $\tilde{K}^{-1}$ and avoid storing the complete $n \times n$ matrix $\tilde{K}$, we iteratively perform the following series of operations:

$$\begin{aligned} T_0^{-1} &= \frac{1}{\sigma} I, \\ T_1^{-1} &= \left(T_0 + d_1 \tilde{K}_{1,k}\right)^{-1} \\ &= \left(T_0 + d_1 \tilde{U}_{1,k} \tilde{\Lambda}_{1,k} \tilde{U}_{1,k}^T\right)^{-1} \\ &\cdots \\ T_M^{-1} &= \left(T_{M-1} + d_M \tilde{K}_{M,k}\right)^{-1} \\ &= \left(T_{M-1} + d_M \tilde{U}_{M,k} \tilde{\Lambda}_{M,k} \tilde{U}_{M,k}^T\right)^{-1}, \end{aligned} \tag{16}$$

where T_m^{-1} is calculated using the SMW formula according to the last result T_{m-1}^{-1}. After performing the series of $M+1$ operations, we can obtain $\tilde{K}^{-1} = T_M^{-1}$.

Lemma 1 (see [33]). *Let A and C both be invertible; then, Sherman–Morrison–Woodbury (SMW) formula gives an explicit formula for the inverse of matrices $A + UCV$ if $C^{-1} + VA^{-1}U$ is invertible.*

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1} + VA^{-1}U\right)^{-1} VA^{-1}. \tag{17}$$

We can find that when the mixture weight is known, formula (15) is same as the dual problem of SVM. Hence, we have

$$J = \frac{1}{2}\boldsymbol{\alpha}^{*T} Y \tilde{K} Y \boldsymbol{\alpha}^* - \mathbf{e}^T \boldsymbol{\alpha}^*, \tag{18}$$

where α^* is the optimal solution minimizing (15). With α^* considered a constant in J, J can be regarded as a function of $\mathbf{d}$, and we calculate the gradient of the objective J with respect to d_m.

$$\begin{aligned} \frac{\partial J}{\partial d_m} &= \frac{1}{2}\boldsymbol{\alpha}^{*T} Y \left(\tilde{U}_{m,k} \tilde{\Lambda}_{m,k} \tilde{U}_{m,k}^T + \sigma I\right) Y \boldsymbol{\alpha}^* \\ &= \frac{1}{2}\boldsymbol{\alpha}^{*T} Y \left(\tilde{U}_{m,k} \tilde{\Lambda}_{m,k} \tilde{U}_{m,k}^T\right) Y \boldsymbol{\alpha}^* + \frac{\sigma}{2}\boldsymbol{\alpha}^{*T}\boldsymbol{\alpha}^*. \end{aligned} \tag{19}$$

We use the reduce gradient method in [27] to deal with problem (14). First, for satisfying the $L1$-norm constraint on the weight vector $\mathbf{d}$ in (14), we calculate the reduced gradient of $\mathbf{d}$:

$$\begin{aligned} \left[\nabla_{\text{red}} J\right]_m &= \frac{\partial J}{\partial d_m} - \frac{\partial J}{\partial d_\mu}, \forall m \neq \mu, \mu = \arg\max_m d_m, \\ \left[\nabla_{\text{red}} J\right]_\mu &= \sum_{m \neq \mu} \left(\frac{\partial J}{\partial d_\mu} - \frac{\partial J}{\partial d_m}\right), \quad \mu = \arg\max_m d_m, \end{aligned} \tag{20}$$

where $\nabla_{\text{red}} J$ denotes the reduced gradient of $J(\mathbf{d})$. Let d_μ be the largest element of the vector $\mathbf{d}$, and μ be the

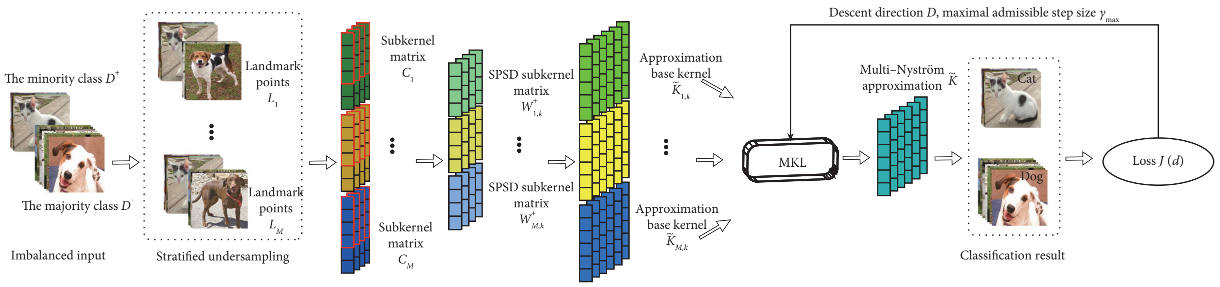

Descent direction D, maximal admissible step size $\gamma_{\max}$
The minority class D^+
The majority class D^-
Imbalanced input
Landmark points L_1
Landmark points L_M
Stratified undersampling
Subkernel matrix C_1
Subkernel matrix C_M
SPSD subkernel matrix $W^+_{1,k}$
SPSD subkernel matrix $W^+_{M,k}$
Approximation base kernel $\widetilde{K}_{1,k}$
Approximation base kernel $\widetilde{K}_{M,k}$
MKL
Multi–Nyström approximation $\widetilde{K}$
Cat
Dog
Classification result
Loss $J(d)$

Figure 1: Architecture of the proposed multi-Nyström method. M subsets from the majority class are sampled to construct balanced landmark points and then the Nyström method is used to obtain the approximate base kernel matrices and the multiple kernel learning (MKL) algorithm is applied to optimize the mixture weights and train classifier. Finally, the trained kernel classifier based on multi-Nystrom is obtained.

corresponding index. Obviously, $-\nabla_{\text{red}}J$ would be a descent direction. However, if $\exists m$ that makes $d_m = 0$ with $-[\nabla_{\text{red}}J]_m < 0$, then $d_m \longrightarrow 0^-$, which does not meet the nonnegative restriction. Therefore, $-[\nabla_{\text{red}}J]_m$ needs to be set to 0. Update descent direction is as follows:

$$D_m = \begin{cases} 0, & d_m = 0, [\nabla_{\text{red}}J]_m > 0, \\ -[\nabla_{\text{red}}J]_m, & d_m > 0, m \neq \mu, \\ -[\nabla_{\text{red}}J]_\mu, & m = \mu. \end{cases} \quad (21)$$

In general, MKL uses a two-step training method. It requires frequent calls to support vector machine solvers, which is prohibitive for large scale problems. Therefore, after each update on $\mathbf{d}$, we are not eager to substitute it into support vector machine solvers to update α^*, but continue to look for the maximum allowable step length in this descent direction until the objective function value stops declining. Finally, we get the optimal step length by the line search method. The complete algorithm of the multi-Nyström method with MKL is summarized in Algorithm 1.

3.3. Kernel Stability Analysis.

3.3. Kernel Stability Analysis. In some previous related works, Nyström is usually considered as a preprocessing method and mostly only study the approximate error bounds without considering the impact of the approximate on the performance of the kernel machine. In the following, we analyze the kernel stability of our method, bounding the relative performance based on the weighted kernel approximation error. It provides performance guarantees for our multi-Nyström approximate method in the context of large scale imbalanced classification.

Proposition 1. *Let α^* be the optimal solution for kernel SVM with kernel K and $\tilde{\alpha}$ be the solution of kernel SVM with kernel $\widetilde{K}$ obtained by Nyström approximation. Then,*

$$\left\|\tilde{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*\right\|_2 \le \theta^2 \frac{1 + \|\widetilde{K}\|_2}{\lambda_{\min}} \Delta \quad \text{with } \Delta = \sum_{m=1}^{M} d_m \left\|(\widetilde{K}_m - K_m)\right\|_2 \left\|\boldsymbol{\alpha}^*\right\|_2, \quad (22)$$

where $\lambda_{\min}$ is the smallest eigenvalue of $\widetilde{K}$, and θ is the constant from Hoffman's bound independent on α^* and $\tilde{\alpha}$.

Proof. Define $\nabla^+ f(\mathbf{x}) \equiv \mathbf{x} - [\mathbf{x} - \nabla f(\mathbf{x})]^+_{\mathscr{X}}$ be the projected gradient, where $\mathscr{X}$ is the bounded constraint and $[\mathbf{x}]^+_{\mathscr{X}} \equiv \operatorname{argmin}_{\mathbf{y}\in\mathscr{X}}\|\mathbf{x} - \mathbf{y}\|$ is the convex projection operator. It can be used to define an error bound according to the following theorem: □

Theorem 1 (see [34]). *Let $\tilde{\mathbf{x}}$ be the nearest optimal solution of the convex optimization problem:*

$$\min_{\mathbf{x}\in\mathscr{X}} f(\mathbf{x}) = g(E\mathbf{x}) + \mathbf{b}^T\mathbf{x}, \quad (23)$$

with $g(\mathbf{t})$ being σ_g strongly convex, $\nabla f(\mathbf{x})$ being ρ Lipschitz continuous, and $\mathscr{X} = \{\mathbf{x} | A\mathbf{x} \le d\}$ is a polyhedral set. The optimization problem admits a global error bound:

$$\|\mathbf{x} - \tilde{\mathbf{x}}\| \le \theta^2 \frac{1+\rho}{\sigma_g} \left\|\nabla^+ f(\mathbf{x})\right\|, \quad \forall \mathbf{x} \in \mathscr{X}, \quad (24)$$

where θ is the constant from Hoffman's bound.

Considering now the problem $\min_{\alpha\in\Omega}\tilde{f}(\alpha) = \tilde{g}(C^TY\alpha) - \mathbf{e}^T\alpha$ with $\tilde{g}(\mathbf{x}) = (1/2)\mathbf{x}^T W_k^+ \mathbf{x}$ and bounded constraint $\Omega = \{\alpha \,|\, \mathbf{y}^T\alpha = 0, 0 \le \alpha \le C\}$, then

$$\min_{\boldsymbol{\alpha}\in\Omega} \tilde{f}(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T Y C W_k^+ C^T Y \boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha}. \quad (25)$$

Note that the above problem is equivalent to problem (15) with the equality $\widetilde{K} = CW_k^+C^T$ (W_k^+ is SPSD), and we have

$$\begin{aligned} \lambda_{\min}(W_k^+)\|\mathbf{x} - \mathbf{y}\|^2 &\le (\nabla\tilde{g}(\mathbf{x}) - \nabla\tilde{g}(\mathbf{y}))^T(\mathbf{x} - \mathbf{y}) \\ &= (\mathbf{x} - \mathbf{y})^T W_k^+ (\mathbf{x} - \mathbf{y}) \\ &\Longrightarrow \sigma_{\tilde{g}} = \lambda_{\min}(W_k^+), \\ \|\nabla\tilde{f}(\mathbf{x}) - \nabla\tilde{f}(\mathbf{y})\| &= \left\|YCW_k^+C^TY(\mathbf{x} - \mathbf{y})\right\| \\ &\le \left\|CW_k^+C^T\right\|\|\mathbf{x} - \mathbf{y}\| \\ &= \|\widetilde{K}\|\|\mathbf{x} - \mathbf{y}\| \\ &\Longrightarrow \rho = \|\widetilde{K}\|_2. \end{aligned} \quad (26)$$

Let f be the dual objective function of multiple kernel learning problem (5) with the original kernel $K = \sum_m d_m K_m$, and $\tilde{f}$ be the objective function of approximate multiple kernel learning problem (9) with kernel $\widetilde{K} = \sum_m d_m \widetilde{K}_m$

```
Input. Dataset $D$; number of landmark points $l$; rank $k$; number of kernels $M$; predefined kernel function $k_m(\cdot,\cdot)$.
Output. Classification result for instance $\mathbf{x}$
Draw $M$ subsets of balanced landmark points $L = \{L_m\}_{m=1}^{M}$ with each $L_m = \{\mathbf{c}_{m,1}, \ldots, \mathbf{c}_{m,l}\}$
Calculate subkernel matrices $C_m \in \mathbb{R}^{n\times l}$ between the instances in $D$ and $L_m$ and $W_m \in \mathbb{R}^{l\times l}$ among the instances in $L_m$ with kernel $k_m(\cdot,\cdot)$
Calculate the singular value decomposition on $W_m = U_m\Lambda_m U_m^T$
Approximate $\tilde{K}_{m,k} = U_{m,k}\Lambda_{m,k}U_{m,k}^T$ according to (8)
Initialize mixture weights $d_m = (1/M)$ for $m = 1, 2, \ldots, M$
While stopping criterion not met do
    Calculate $J(\mathbf{d})$ by SVM solver with $\tilde{K} = \sum_m d_m \tilde{K}_{m,k}$ according to (15)
    Calculate descent direction $\mathbf{D}$ according to (19)–(21)
    Set $\mu = \text{argmax}_m d_m$, $J^* = 0$, $\mathbf{d}^* = \mathbf{d}$, $\mathbf{D}^* = \mathbf{D}$
    While $(J^* < J(\mathbf{d}))$ do
        Set $\mathbf{d} = \mathbf{d}^*$, $\mathbf{D} = \mathbf{D}^*$
        Set $\nu = \text{argmin}_{\{m|D_m<0\}} - (d_m/D_m)$, $\gamma_{\max} = -(d_\nu/D_\nu)$
        Set $\mathbf{d}^* = \mathbf{d} + \gamma_{\max}\mathbf{D}$, $D_\mu^* = D_\mu - D_\nu$, $D_\nu^* = 0$
        Calculate $J^*$ with $\tilde{K} = \sum_m d_m^* \tilde{K}_{m,k}$
    end while
    Line search along $\mathbf{D}$ for optimal $\gamma^* \in [0, \gamma_{\max}]$
    Assign $\mathbf{d} \longleftarrow \mathbf{d} + \gamma^*\mathbf{D}$
end while
```

Algorithm 1: The proposed MKLMO algorithm.

obtained by our multi-Nyström method (13). Consider now α^* and $\tilde{\alpha}$ as the optimal solutions of $f(\alpha)$ and $\tilde{f}(\alpha)$, respectively. We have

$$\begin{aligned}\nabla\tilde{f}(\boldsymbol{\alpha}^*) &= Y\tilde{K}Y\boldsymbol{\alpha}^* - YKY\boldsymbol{\alpha}^* + YKY\boldsymbol{\alpha}^* - \mathbf{e}\\ &= Y(\tilde{K} - K)Y\boldsymbol{\alpha}^* + \nabla f(\boldsymbol{\alpha}^*),\end{aligned}\tag{27}$$

where we use the fact that $\nabla f(\alpha^*) = 0$ and $\nabla\tilde{f}(\tilde{\alpha}) = 0$; therefore,

$$\begin{aligned}\|\nabla\tilde{f}(\boldsymbol{\alpha}^*)\|_2 &= \left\|\sum_{m=1}^{M} d_m Y(\tilde{K}_m - K_m)Y\boldsymbol{\alpha}^*\right\|_2\\ &\le \sum_{m=1}^{M} d_m\|(\tilde{K}_m - K_m)\boldsymbol{\alpha}^*\|_2\\ &\le \sum_{m=1}^{M} d_m\|(\tilde{K}_m - K_m)\|_2\|\boldsymbol{\alpha}^*\|_2,\end{aligned}\tag{28}$$

where $\|(\tilde{K}_m - K_m)\|_2$ is the spectral norm error of the m^{th} Nyström approximate based on the m^{th} subset of landmark points.

Furthermore, we use the inequality $\|\nabla^+\tilde{f}(\alpha^*)\|_2 \le \|\nabla\tilde{f}(\alpha^*)\|_2$ of the kernel SVM given by [35] (proof of Theorem 2) along with Theorem 1 to upper bound the norm difference between the optimal solutions of $f(\alpha)$ and $\tilde{f}(\alpha)$:

$$\begin{aligned}\|\boldsymbol{\alpha}^* - \tilde{\boldsymbol{\alpha}}\|_2 &\le \theta^2\frac{1+\rho}{\sigma_{\tilde{g}}}\|\nabla^+\tilde{f}(\boldsymbol{\alpha}^*)\|_2\\ &\le \theta^2\frac{1+\rho}{\sigma_{\tilde{g}}}\|\nabla\tilde{f}(\boldsymbol{\alpha}^*)\|_2\\ &\le \theta^2\frac{1+\rho}{\sigma_{\tilde{g}}}\sum_{m=1}^{M} d_m\|(\tilde{K}_m - K_m)\|_2\|\boldsymbol{\alpha}^*\|_2.\end{aligned}\tag{29}$$

The proposition shows us the norm difference $\|\alpha^* - \tilde{\alpha}\|_2$ is controlled by a weighted Nyström approximate error. And it guides us to focus on approximating the kernel matrices with greater weights for getting a better learning performance.

4. Experiments

In this section, in order to validate the efficiency of the proposed method in solving large scale imbalanced problems, we compare our method against kernel methods including SVM and MKSVM (multiple kernel SVM), as well as the Nyström approximation method. All experiments are implemented on a PC with Intel quad-core i7-8565U CPU@ 1.80 GHz and 8 GB memory.

4.1. Implementation. We implement our experiments on five real-world imbalanced datasets from the KEEL data repository (https://keel.es/) and the LIBSVM archive (https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/) (Table 1). For a fair comparison, we perform 10 times stratified 5-fold cross-validation and report the average result. We use LIBSVM (https://www.csie.ntu.edu.tw/cjlin/libsvm/index.html) and SimpleMKL (https://asi.insa-rouen.fr/enseignants/arakoto/code/mklindex.html) to run kernel SVM and MKSVM, respectively. As the kernel type, all experiments use the Gaussian kernel with bandwidth σ in the range of $\log_{10}\sigma = \{-1, 0, 1, 2\}$. Because we are interested in relative performance, we empirically set the trade-off parameter $C = 100$. In this study, we adopt the following three evaluation measures of the classification performance on imbalanced datasets: $F1$ score, G-mean, and area under ROC curve (AUC).

Table 1: Datasets used in experiments

Dataset	# feature	# instance	IR
Poker-8-9_vs_5	10	2075	82
Abalone19	8	4174	129.44
Page-blocks0	10	5472	8.79
USPS (class 9 against all)	256	9298	12.13

$$\begin{aligned} \mathrm{PRE} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \\ \mathrm{REC} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \\ \mathrm{SPE} &= \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}, \\ F1\ \text{score} &= \frac{2\times \mathrm{PRE}\times \mathrm{REC}}{\mathrm{PRE}+\mathrm{REC}}, \\ G-\text{mean} &= \sqrt{\mathrm{REC}\times \mathrm{SPE}}, \end{aligned} \tag{30}$$

where TP, TN, FP, and FN represent the number of true-positive, true-negative, false-positive, and false-negative instances, respectively. $F1$ score measures the classification performance on the minority class. G-mean reflects the overall classification performance. AUC works well for comparing performance between algorithms [36].

4.2. Experimental Results. Table 2 provides the average experimental results of the proposed method and the other three algorithms on the four imbalanced datasets using the above three measures. We first compare SVM and the standard Nyström method. The Nyström method uses uniform sampling without replacement to approximate the kernel matrix, which relieves the model's sensitivity to class imbalance to a certain extent. For example, on the Poker-8-9_vs_5 dataset, in terms of G-mean, the Nyström method improves nearly 7 times more than SVM. However, we can also see that in terms of AUC and $F1$ score, there still exits a large gap in model accuracy as compared with SVM.

Next, we compare our multi-Nyström method with the standard Nyström method. The experimental results clearly demonstrate that our method outperforms the Nyström method, especially in the context of extreme imbalance. This mainly benefits from the use of undersampling of the majority class, which can effectively balance the class distribution. Moreover, it can be seen that multi-Nyström can improve the accuracy of the model. For example, with the same number of landmark points, the $F1$ score and AUC value of multi-Nyström on the USPS dataset are closer to that of SVM or even higher on Poker-8-9_vs_5 and Page-blocks0 datasets.

Note that our method is also a type of approximation of MKL, and finally, we also examine the performance of MKL-based MKSVM. From the results, we can see the effect of using MKL to represent input data, which also implicitly explains how our method achieves better accuracy at the expense of more computations.

Table 2: $F1$ score, G-mean, and AUC results of different algorithms on four datasets

Datasets	Measures	SVM	Nyström	Multi-Nyström	MKSVM
Poker-8-9_vs_5	$F1$	0.0571	0.0327	0.0585	0.1906
	G-mean	0.0399	0.3357	0.5140	0.1589
	AUC	0.8107	0.6106	0.7953	0.7942
Abalone19	$F1$	0.0611	0.0200	0.0334	0.0569
	G-mean	0.0661	0.2914	0.3500	0.0661
	AUC	0.7487	0.5203	0.6094	0.7263
Page-blocks0	$F1$	0.8061	0.7954	0.8171	0.8342
	G-mean	0.7018	0.7292	0.7115	0.7381
	AUC	0.9857	0.9753	0.9585	0.9904
USPS	$F1$	0.8991	0.6688	0.8853	0.9102
	G-mean	0.8788	0.8593	0.8408	0.8807
	AUC	0.9939	0.9608	0.9874	0.9963

4.3. Discussion. In this part, we further discuss the impact of different parameters on performance. In the first experiment, in order to study the impact of the number of sampling landmark points on the classification performance, we fix the approximate rank parameter and successively increase the number of sampling landmark points, and then train and test the SVM model on four datasets, with results as shown in Figure 2. We can see that as the number of sampling landmark points increases, although there are some fluctuations, the performance of our method and Nyström still presents a rising trend. Moreover, except for few cases, our method uses fewer landmark points and can still yield higher G-mean.

In the second experiment, we study the performance with the variance of the rank parameter. Figure 3 shows the G-mean on four datasets by varying the approximate rank. They show us that with the same approximate kernel rank, our method can achieve better classification performance than others.

Finally, we further compare the running time of our method and MKSVM. We report the results on two datasets USPS and Page-blocks in Figure 4. The results show that our method can significantly speedup the MKL process under guaranteed performance. For example, on the USPS dataset, our method can reduce the running time by more than one order of magnitude. The main reason is due to the low-rank attribute of the approximate kernel matrix that speeds up the MKL algorithm process.

For further analysis of the experimental results, we perform the Friedman test with respect to the $F1$ score. First, we calculate the average ranks of SVM, Nyström, multi-Nyström, and MKSVM as shown in Figure 5. It can be noticed that MKSVM gives the best performance. Meanwhile, the SVM and the proposed multi-Nyström rank similarly. In a comparison of k algorithms on N datasets, considering r_i as the average ranking of the i^{th} algorithm, the Friedman variable F_F can be calculated as follows:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}, \tag{31}$$

with

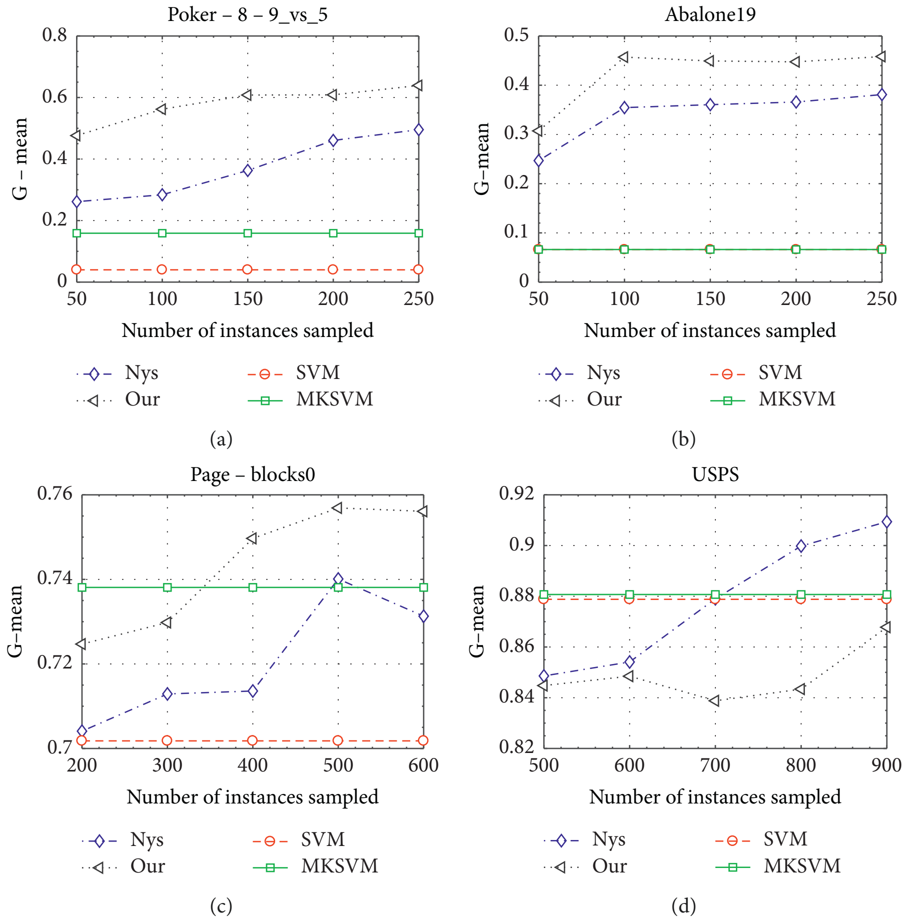


Figure 2: Classification performance with different numbers of instances sampled on four datasets.

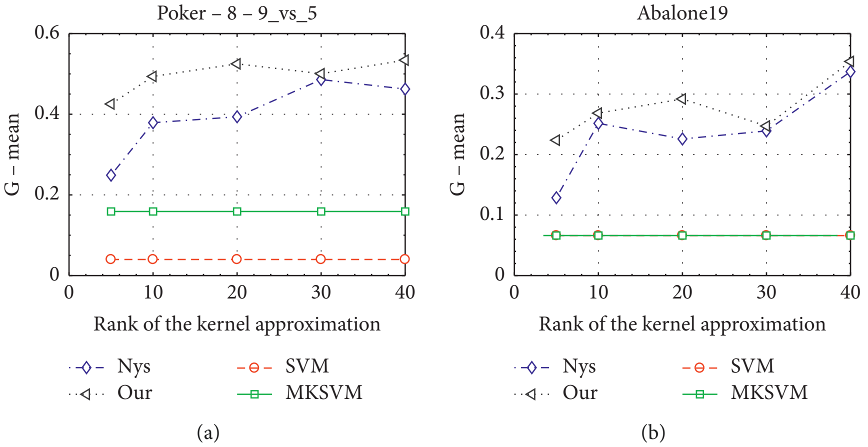
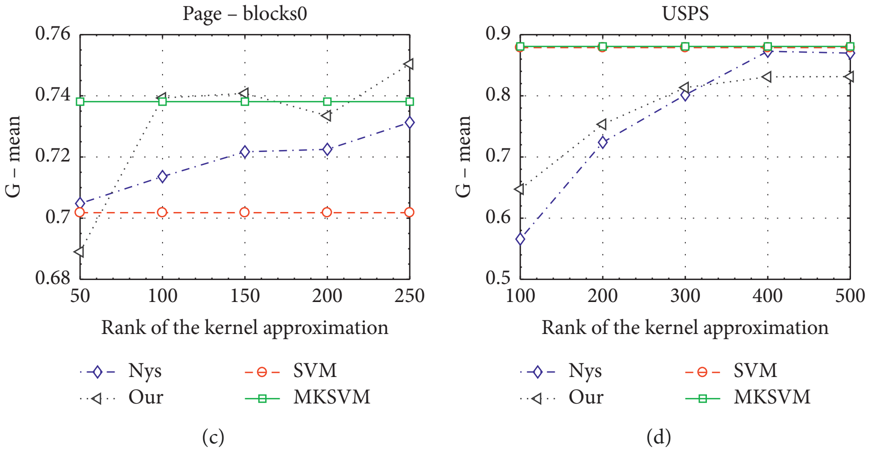


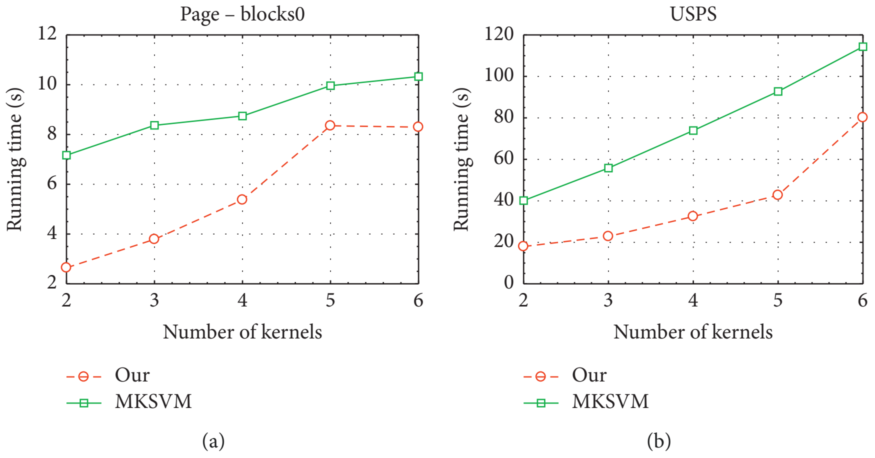
Figure 3: Continued.



(c)

(d)

Figure 3: Classification performance with different ranks of the kernel approximation on four datasets.



(a)

(b)

Figure 4: Timing performance with different numbers of kernels on two datasets.

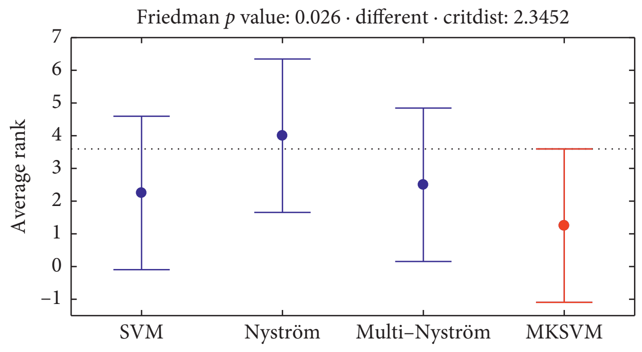


Figure 5: Average rank of the four algorithms for four datasets.

$$\chi_F^2 = \frac{12N}{k(k+1)}\left(\sum_{i=1}^{k} r_i^2 - \frac{k(k+1)^2}{4}\right), \tag{32}$$

where F_F is distributed to $(4-1)$ and $(4-1)(4-1)$ degrees of freedom. For our experiments, $F_F = 10.3333$. The critical value of $F(3,9)$ is 3.8625 for $\alpha = 0.05$. Since $F_F > F(3,9)$, we can reject the null hypothesis that all the algorithms have the same performance. Then, we perform the Nemenyi test to compare algorithms pairwise. The critical difference is calculated as follows:

$$\mathrm{CD} = q_a\sqrt{\frac{k(k+1)}{6N}}, \tag{33}$$

considering $\alpha = 0.05$ and $\mathrm{CD} = 2.3452$. The difference between the average ranking of the SVM, Nyström, and multi-Nyström with MKSVM is 1.0, 2.75, and 1.25, respectively. Hence, we can state that the best MKSVM is significantly better than Nyström at $\alpha = 0.05$. However, the difference between the best MKSVM and the proposed multi-Nyström is not significant, which indicates the proposed method achieves better performance than the standard Nyström kernel classifier and more efficiency than the best MKSVM.

5. Conclusions

In this study, we propose a novel method to overcome the time and memory limitations of the standard Nyström method and extend it to the case of large scale imbalanced classification. In general, kernel approximation and model training are carried out separately. To obtain more accurate results, our method mixes multiple Nyström approximations and embeds them in the model training process to learn the model parameters and mixture weights simultaneously. In particular, the approximate kernel matrix yielded by our method is low rank and balanced. We also provide an error bound of the model solution based on our approximate method to guide us in improving the learning process. Experimental results show that our method can achieve a higher classification accuracy. On the other hand, it can dramatically improve the efficiency of exiting MKL algorithms.

Potential improvements: there are still some caveats in our current solution. For example, due to the curse of kernelization, the number of support vectors grows in an unbounded manner when suffered the nonzero loss. This significantly increases the computational cost and can be infeasible for large scale problems. Future work will chiefly focus on more efficient variants of multi-Nyström involving budget kernel learning to address the issue.

Data Availability

The data used to support the findings of this study have been deposited in the KEEL repository (http://keel.es/) and the LIBSVM archive (https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was jointly supported by the National Natural Science Foundation of China (61403397) and Natural Science Basic Research Plan in Shaanxi Province of China (2020JM-358 and 2015JM6313).

References

[1] C. Huang, Y. Li, C. Change Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5375–5384, Las Vegas, NV, USA, June 2016.

[2] J. Zhu and E. Hovy, "Active learning for word sense disambiguation with methods for addressing the class imbalance problem," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 783–790, Prague, Czech Republic, June 2007.

[3] W. Sun, J. Sun, Y. Zhu, and Y. Zhang, "Video super-resolution via dense non-local spatial-temporal convolutional network," *Neurocomputing*, vol. 9, no. 403, pp. 1–12, 2020.

[4] N. V. Chawla, "Data mining for imbalanced datasets: an overview," in *Data Mining and Knowledge Discovery Handbook*, pp. 875–886, Springer, Berlin, Germany, 2009.

[5] X. Luo, J. Sun, L. Wang et al., "Short-term wind speed forecasting via stacked extreme learning machine with generalized correntropy," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4963–4971, 2018.

[6] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of imbalanced data: a review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687–719, 2009.

[7] 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 Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2011.

[8] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[10] H. He, Y. Bai, A. Edwardo Garcia, and S. Li, "Adasyn: adaptive synthetic sampling approach for imbalanced learning," in *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1322–1328, IEEE, Hong Kong, China, June 2008.

[11] R. Batuwita and V. Palade, "FSVM-CIL: fuzzy support vector machines for class imbalance learning," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 558–571, 2010.

[12] H. Yu, C. Sun, X. Yang, S. Zheng, and H. Zou, "Fuzzy support vector machine with relative density information for classifying imbalanced data," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 12, pp. 2353–2367, 2019.

[13] X. Hong, S. Chen, and C. J. Harris, "A kernel-based two-class classifier for imbalanced data sets," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 28–41, 2007.

[14] Y. Tang and Y.-Q. Zhang, N. V. Chawla and S. Krasser, "SVMS modeling for highly imbalanced classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 281–288, 2008.

[15] A. Maratea, A. Petrosino, and M. Manzo, "Adjusted F-measure and kernel scaling for imbalanced data learning," *Information Sciences*, vol. 257, pp. 331–341, 2014.

[16] J. Mathew, C. Khiang Pang, M. Luo, and W. H. Leong, "Classification of imbalanced data by oversampling in kernel space of support vector machines," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 4065–4076, 2017.

[17] G. Wu and E. Y. Chang, "KBA: kernel boundary alignment considering imbalanced data distribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 786–795, 2005.

[18] Bo Tang and H. He, "Kerneladasyn: kernel based adaptive synthetic data generation for imbalanced learning," in *Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC)*, pp. 664–671, IEEE, Sendai, Japan, May 2015.

[19] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," *Advances in Neural Information Processing Systems*, vol. 13, pp. 682–688, 2000.

[20] P. Drineas and M. W. Mahoney, "On the Nyström method for approximating a gram matrix for improved kernel-based learning," *Journal of Machine Learning Research*, vol. 6, pp. 2153–2175, 2005.

[21] C. Musco and C. Musco, "Recursive sampling for the Nyström method," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3836–3848, Long Beach, CA, USA, December 2017.

[22] S. Kumar, M. Mohri, and A. Talwalkar, "Ensemble Nyström method," *Advances in Neural Information Processing Systems*, vol. 22, pp. 1060–1068, 2009.

[23] Z. Li, T. Yang, L. Zhang, and R. Jin, "Fast and accurate refined Nyström-based kernel SVM," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, Phoenix, AZ, USA, February 2016.

[24] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, vol. 36, no. 3, pp. 1171–1220, 2008.

[25] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Science & Business Media, Berlin, Germany, 2013.

[26] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[27] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[28] M. Alioscha-Perez, M. Cédric Oveneke, and H. Sahli, "SVRG-MKL: a fast and scalable multiple kernel learning solution for features combination in multi-class classification problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1710–1723, 2019.

[29] S. Kumar, M. Mohri, and A. Talwalkar, "Sampling techniques for the Nyström method," in *Proceedings of the Artificial Intelligence and Statistics*, pp. 304–311, Clearwater Beach, FL, USA, April 2009.

[30] L. Wang, H. Wang, and G. Fu, "Multiple kernel learning with minority oversampling for classifying imbalanced data," *IEEE Access*, vol. 9, pp. 565–580, 2021.

[31] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *The Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[32] S. Kumar, M. Mohri, and A. Talwalkar, "Sampling methods for the Nyström method," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 981–1006, 2012.

[33] C. Y. Deng, "A generalization of the Sherman-Morrison-Woodbury formula," *Applied Mathematics Letters*, vol. 24, no. 9, pp. 1561–1564, 2011.

[34] P.-W. Wang and C.-J. Lin, "Iteration complexity of feasible descent methods for convex optimization," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1523–1548, 2014.

[35] C.-J. Hsieh, Si Si, and I. S. Dhillon, "Fast prediction for large-scale kernel machines," in *Proceedings of the Neural Information Processing Systems*, pp. 3689–3697, Citeseer, Montreal, Quebec, Canada, December 2014.

[36] S. Barua, Md Monirul Islam, X. Yao, and K. Murase, "Mwmote–majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, 2012.