$\ell_{2,1}$ norm regularized multi-kernel based joint nonlinear feature selection and over-sampling for imbalanced data classification

Peng Cao^a, Xiaoli Liu^a, Jian Zhang^c, Dazhe Zhao^d, Min Huang^b, Osmar Zaiane^e

^aCollege of Computer Science and Engineering, Northeastern University, Shenyang, China

^bCollege of Information Science and Engineering, Northeastern University, Shenyang, China

^cSchool of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, China

^dKey Laboratory of Medical Image Computing of Ministry of Education, Northeastern University, Shenyang, China

^eComputing Science, University of Alberta, Edmonton, Alberta, Canada

Abstract

High dimensionality and classification of imbalanced data sets are two of the most interesting machine learning challenges. Both issues have been independently studied in the literature. In order to simultaneously explore the both issues of feature selection and oversampling, we efficiently combine two different methodological approaches in an unified kernel framework. Specifically, we proposed a novel $\ell_{2,1}$ norm balanced multiple kernel feature selection ($\ell_{2,1}$ MKFS), and designed a proximal based optimization algorithm for efficiently learning the model. Moreover, multiple kernel oversampling (MKOS) was developed to generate synthetic instances in the optimal kernel space induced by $\ell_{2,1}$ MKFS, so as to compensate for the class imbalanced distribution. Our experimental results on multiple UCI data and two real medical application demonstrate that jointly operating nonlinear feature selection and oversampling with $\ell_{2,1}$ norm multi-kernel learning framework ($\ell_{2,1}$ MKFSOS) can lead to a promising classification performance.

Keywords: imbalanced data learning, feature selection, classification, multi-kernel learning, proximal method

1. Introduction

Class imbalance has been recognized as a crucial problem in machine learning [1, 2]. This problem occurs when the training data is not evenly distributed among classes, and is growing in importance and has been identified as one of the 10 main challenges of data mining [7]. Imbalanced data learning is also especially critical in many real world applications, such as anomaly detection in credit card transaction, fraud detection, medical diagnosis, equipment malfunction detection, etc. In these cases, standard classifiers usually tend to be overwhelmed by the majority class and ignore the minority class examples, since most classifiers assume an even distribution of examples among classes. Much work has been done in addressing the class imbalance problem, which can be grouped in two categories: the algorithm perspective and the data perspective[48, 9, 26, 3, 5]. Cost-sensitive learning tries to learn more characteristics of samples with the minority class from algorithm perspective by setting a high cost to the misclassification of samples of the minority class [11, 35, 4]. The methods with the data perspective re-balance the class distribution by re-sampling the data space either randomly or deterministically, so that the minority class can be well represented in the training set[1]. The re-sampling technique is the most straightforward and effective method for dealing with imbalance, since it is not dependent on the classifier and is simple to implement.

Feature selection is also an important topic in machine learning, especially in high-dimensional applications. Feature selection methods select a small subset of the original feature set to reduce the dimensionality of the data set and facilitate better generalization of training samples. Moreover, in some applications, features exhibit certain intrinsic group structures (e.g. multiple specific geometric features corresponds to a region-of-interest (ROI) in the brain; heterogeneous feature subsets from different view are extracted to describe an object). Hence the group effect in the features need to be taken into account while doing feature selection.

The high dimensionality and imbalanced class distribution occur at the same time. Class-imbalanced prediction of high-dimensional data presents an additional challenge since the irrelevant and redundant features influence the performance of the re-sampling algorithms for balancing the skewed class distribution. A number of solutions have been proposed to deal with class imbalance and feature reduction respectively. A low-dimensional representation of the data not only reduces the risk of overfitting of classifier model, but also can be very helpful to oversampling methods when facing imbalanced data sets. However, to the best

of our knowledge, jointly exploring feature selection and oversampling for imbalanced data classification has not been studied in the literature so far. Besides, most of the existing feature selection or oversampling methods are based linear model. Sparse models, such as lasso and other linear norm with considering certain intrinsic structures (spatial groups or temporal smoothness) in data [59] select an informative feature subset from the original feature set, and have achieved great success. However, they cannot capture the complex patterns when the inherent structure of data represents underlying nonlinear patterns in many real applications. Meanwhile, many oversampling algorithms generate new examples for the minority class via linear interpolation between existing minority class examples. However, it tends to result in distortion and inconsistencies when combined with kernel classifiers (e.g. SVM), since kernel classifiers work in the feature space while oversampling operates in the input space.

To address the above challenges, we propose an unified multi-kernel framework to efficiently combine nonlinear feature selection and oversampling algorithm. There has been much recent progress in nonlinear feature selection for multi-label classification [13], bioinformatics [15] and object categorization [16] where the predictor is a nonlinear function of the input features. Multiple kernel learning techniques approach the problem of feature selection as solving a task of learning the optimal weights for each feature, and have been shown to be amongst the most effective for nonlinear feature selection[37]. To perform nonlinear feature selection, a base kernel is usually computed for each feature in [44, 37], which may result in important features removed if the kernel parameter is not set appropriately. In order to improve over it, we associate multiple nonlinear base kernel with each individual feature, and regularize a multi-kernel learning model with an $\ell_{2,1}$ -norm to find the relevant features. The $\ell_{2,1}$ regularizer would enforce the sparsity among the features by ℓ_1 norm, while nonsparsity can be preserved for base kernels within the same feature by ℓ_2 norm. Then, we adapts a proximal gradient method for solving the $\ell_{2,1}$ norm of kernel weights, and use an accelerated method based on FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) [23]. On the other hand, in order to achieve the nonlinear oversampling in the kernel space, the most challenges are needed to be solved: how to generate new instances in the kernel space and how to exactly estimate the pre-image of the synthetic instances since the inverse of nonlinear map ϕ typically does not exist. In our work, we extend the original SMOTE algorithm into feature space optimized by $\ell_{2,1}$ MKL, and then use this relationship between feature-space distance and input-space distance, to embed the pre-image of new generated instances back to the input space.

The main contributions of our work are:

1. We proposed a novel multi-kernel framework to address the both issues of feature selection and imbalanced data learning simultaneously. The new multiple kernel based nonlinear feature selection not only alleviates the curse of dimensionality to derive a lower dimensional representation, but also facilitates the subsequent oversampling and classification tasks.

2. We regularize multiple kernel learning with $\ell_{2,1}$ norm and utilize a balanced loss as objective function to nonlinear feature selection for imbalanced data, called $\ell_{2,1}$ MKFS. Moreover, our $\ell_{2,1}$ MKFS can automatically determine the number of features when performing the feature selection procedure. Furthermore, an accelerated proximal gradient optimization algorithm is also introduced to optimize the parameters of kernel weights by solving the non-smooth objective of $\ell_{2,1}$ regularized multiple kernel learning with rigorously proved global convergence.

3. Besides the multi-kernel based feature selection (MKFS), the multi-kernel over-sampling (MKOS) method is proposed to balance the data distribution. The over-sampling is conducted on the reduced feature set and operate in the kernel space induced by $\ell_{2,1}$ MKFS for minority instances. Then the pre-images of new instances in the input space are estimated, so as to address the inconsistency issue created by the fact that the oversampling is carried out in the input space while the model is constructed in the kernel space. The new oversampling method in the optimal kernel space can generate accurate instances and address the problem of inconsistency.

4. To highlight the advantages of our proposed $\ell_{2,1}$ MKFSOS, we performed feature selection and oversampling on the various class imbalanced datasets from UCI, and applied our method to two real medical clinical applications: detection of lung nodule and diagnosis of Alzheimer’s disease. The results verify the effectiveness of nonlinear feature selection and oversampling in the multiple kernel framework for the complex data with heterogeneous feature subsets and feature groups, respectively.

The rest of the paper is organized as follows. In Section 2, we provide a summary of the various approaches taken to combat the issue of imbalanced data learning. A detailed mathematical formulation of $\ell_{2,1}$ MKFSOS is provided in Section 4. In Section 5, we conducted extensive experiments to verify the advantage of our method for imbalanced data classification. The conclusion is drawn in Section 7.

2. Related Work

2.1. *The imbalanced data problem*

A common problem faced in data mining is dealing with class imbalance. A dataset is said to be imbalanced if one class (called the majority, or negative class) vastly out-numbers the other (called the minority, or positive class). The class imbalance problem is only said to exist when the positive class is the class of interest. This is due to the fact that if the positive, minority, class is not of interest (i.e., it has no effect on the choice made), then it can be safely ignored. In most practical applications (e.g., loan recommendation, fraud prevention, spam detection, intrusion detection, species modeling, long term epidemiological studies, climate data analysis, etc.), however, the minority class is the class of interest, and therefore the class imbalance problem must be addressed.

Most classifiers assume an even distribution of examples among classes and assume an equal misclassification cost. Moreover, classifiers are typically designed to maximize accuracy, which is not a good metric to evaluate effectiveness in the case of imbalanced training data. Therefore, we need to improve traditional algorithms so as to handle imbalanced data and choose other metrics to measure performance instead of accuracy. We focus our study on imbalanced datasets with binary classes. Although support vector machines (SVM) often works effectively with balanced datasets, it could produce suboptimal results with imbalanced datasets. SVM tries to minimize the regularized hinge loss; it is driven by an error based objective function. Hence, an SVM classifier trained on an imbalanced dataset often produces models which are biased towards the majority class and have low performance on the minority class.

Evaluation measures play a crucial role in both assessing the classification performance and guiding the classifier modeling. The purpose of cost-sensitive learning is usually to build a model with total minimum misclassification costs. However, it should be based on the known cost matrix condition. In this article, the purpose of our cost sensitive learning is to get a best AUC or G-mean evaluation metric. And we train the cost sensitive learning using performance measures as the objective functions directly. Through training the cost sensitive classifier with measure based objective functions, we can discover the best factors in terms of the different evaluation. For imbalanced datasets, the evaluation metric should take into account the imbalance. The average accuracy is not an appropriate evaluation metric. We used the G-mean and AUC to evaluate the cost sensitive classifiers. The evaluation metrics value is taken as the fitness function to adjust the position of a particle. These two different evaluations reflect different aspec-

t of the classifier. The AUC concerns the ranking ability more and the G-mean concerns the two accuracies of both classes at the same time. The G-mean is the geometric mean of specificity and sensitivity, which is commonly utilized when performance of both classes is concerned and expected to be high simultaneously [25]. It is a good indicator on overall performance, and has been used by several researchers for evaluating classifiers on imbalanced datasets [28, 24, 33]. ROC analysis (abbr. of Receiver Operating Characteristic) has been recently introduced to evaluate machine learning algorithm. ROC curves measure the separating ability of a classifier between two classes. It depicts all possible trade-off between TP rate and FP rate. Closely related to ROC, AUC represents a ROC curve as a single scalar value by estimating the area under the curve, varying between 0 and 1. The AUC measures the performance of ranking a randomly chosen positive example higher than a randomly chosen negative example. In this case, it represents the performance of ranking an instance from the minority class higher than instances in the majority class. Since AUC is believed to be a better performance measure than accuracy for imbalanced classification problems, and independent of class prevalence. It is commonly used as the measure criteria for evaluating performance of classification on the imbalanced dataset[27, 33].

2.2. The oversampling for imbalanced data learning

Significant works have been conducted to deal with the imbalanced learning problem. These methods can be grouped in two categories: the data perspective and the algorithm perspective. Re-sampling methods are attractive under most imbalanced circumstances. This is because re-sampling adjusts only the original training data set instead of modifying the learning algorithm; therefore it provides a convenient and effective way to deal with imbalanced learning problems using standard classifiers. In this paper, we are only interested in sampling based approaches and hence we provide a brief overview of the methods proposed in this category.

A popular and effective oversampling method is the synthetic minority oversampling technique (SMOTE) [48]. The algorithm of SMOTE, randomly selects one of the K-nearest neighbors of $\mathbf{x}_i$, then multiply the corresponding feature vector difference with a random number λ_{ij} , and finally, adds this vector to $\mathbf{x}_i$. The new instance $\mathbf{x}_{ij}^{new}$ is:

$$\mathbf{x}_{ij}^{new} = \mathbf{x}_i + \lambda_{ij} \times (\mathbf{x}_i - \mathbf{x}_j) \quad (1)$$

Experiments show that this technique to be more useful for SVM than under-sampling or random oversampling[28]. SMOTE has gained popularity in solving imbalance problem due to its performance, and researchers put forward a number of the new improved algorithms based on it, such as SMOTEBoost[19], RSMSMOTE[20].

However it manipulates the instances blindly without considering the data distribution, resulting in creating wrong instances falling inside majority regions when the class dispersion or class noise exist. Moreover, it generates the same number of synthetic data samples for each original minority example. There exist many methods based on SMOTE for generating more appropriate instances to overcome this limitation of SMOTE. Cluster Based Synthetic Oversampling (CB-SO) algorithm [24] integrates clustering with the SMOTE, so as to ensure that synthetic samples created always lie inside minority regions. Adaptive Synthetic Sampling (ADASYN) algorithm [18] use a density distribution as a criterion to automatically decide the number of synthetic samples that need to be generated for each minority example by adaptively changing the weights of different minority examples to compensate for the skewed distributions.

2.3. Feature selection for imbalanced data learning

The imbalanced data distribution is often accompanied by high dimensionality in real-world data sets such as text classification and bioinformatics. The main idea of feature selection is to choose a subset of input features by eliminating features with little or no predictive information according to some measure. Feature selection algorithms [6] can be organized into three categories: filter, wrapper and embedded methods. These three differ in how the learning algorithm is incorporated while evaluating and selecting features. For imbalanced data learning, the most commonly used is wrapper and filter methods. In wrapper approach the feature selection method works in conjunction with the learning algorithm, while the filter approach selects the features using a preprocessing step that ignores the learning algorithm and it does not take into account the effect of the subset of features in the learning algorithm. A number of feature selection metrics have been explored, notable among which are Information Gain (IG), RElevance In Estimating Features (RELIEF), Correlation Coefficient (CC), however they do not consider the imbalanced data distribution. Feature Assessment by Sliding Thresholds (FAST)[17], which is based on the area under a ROC curve generated by moving the decision boundary of a single feature classifier with thresholds placed using an even-bin distribution. The second approach is based on the wrapper scheme. MOCSSVM [35] is a wrapper feature selection paradigm oriented by the evaluation measure

of imbalanced dataset as objective function with respect to misclassification cost, feature subset and intrinsic parameters of SVM simultaneously for improving the performance of cost-sensitive SVM. Recursive Feature Elimination (RFE) [36] is wrapper feature selection by employing a backward algorithm for feature elimination with balanced loss function. The third class of feature selection techniques is embedded methods, which determine the respective feature subset during classifier construction, which can be seen as a search in the combined space of feature subsets and hypotheses at the same time. Similar to wrapper methods, embedded approaches are specific to a given classifier algorithm but computationally less intensive than wrapper methods. By now, no embedded methods have been developed for solving the imbalanced data learning.

3. Proposed Method: $\ell_{2,1}$ MKAOS for feature selection and over-sampling for imbalanced data learning

In this section, we introduce the notations and provide a brief overview of $\ell_{2,1}$ MKFSOS at first, then we subsequently describe the detailed mathematical formulation and algorithm procedure of the $\ell_{2,1}$ norm regularized multiple kernel feature selection (MKFS) and multiple kernel oversampling (MKOS), respectively.

3.1. Notation

We first introduce some notations here. Bold capital letters, e.g., $\mathbf{K}$ indicates kernel matrix, and $\mathbf{D}$ indicates the matrix of kernel weight. Bold small letters, e.g., $\mathbf{w}$ and $\mathbf{d}$, indicate vectors. Let $\mathbf{x}_i$ be a training instance, and the training set is denoted as $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times P}$, where N and P are the number of training instances and dimensionality of $\mathbf{x}_i$. Let $\mathbf{y} = (y_1, \dots, y_N)^T$ be the label vector, with $y_i \in \{+1, -1\}$ being the class label of the i -th training instance. The amounts of minority and majority class are N_+ and N_- , respectively.

3.2. Brief overview of $\ell_{2,1}$ MKFSOS

The goal of $\ell_{2,1}$ norm multiple kernel based feature selection (MKFS) and oversampling (MKOS) is to process the imbalanced data from both the sample level and feature level in the feature space, so as to solve the classification of class imbalanced data with irrelevant features. An overview of the proposed method is illustrated in Figure 1.

The procedure of our methods involves four steps: building $\ell_{2,1}$ multi-kernel model, feature selecting according to the kernel weights, multi-kernel oversampling and training a SVM classifier on the new training dataset. $\ell_{2,1}$ MKFS not only reduce the dimensionality so as to improve the performance of the re-sampling algorithm, but also makes the distribution of minority samples denser as well as more well-defined, and explore the borderline instance located at the decision boundary, which are critical to classification and oversampling for minority class. MKOS conducts the oversampling in the optimal feature space determined by the lower features selected and kernel weights optimized by $\ell_{2,1}$ MKFS, overcoming the inconsistency issue where oversampling work in the input space and kernel classifier work in the kernel space.

Figure 1: Illustration of the proposed MKFSOS method: The original dataset $\mathbf{X}$ contains P features and N instances (N_- majority class instances, and N_+ minority class instances). The minority class is labeled by $+1$ and majority class is labeled by -1 . We build a $\ell_{2,1}$ MKL and calculate the kernel weight of each feature at first. Then, the dimensionality of feature set is reduced to P' by MKFS by removing some features with little contribution to classification, then the minority instances are oversampling to by $N_+ \times (R/100)$ instances on the remaining reduced feature set (R is oversampling ratio). Hence, the original training dataset T_{tr} is transformed into T'_{tr} by the two components in our proposed $\ell_{2,1}$ MKFSOS from the levels of features and instances simultaneously.

3.3. Multiple kernel learning

The primal objective function of Support Vector Machines (SVM) is:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y^n(\mathbf{w}^T \mathbf{x}_i + b) + \xi_i \geq 1, \\ & \xi_i \geq 0, \quad \forall i. \end{aligned} \quad (2)$$

where $\mathbf{w}$ is the normal of the separating hyperplane, b is the bias term, $\boldsymbol{\xi} = [\xi_1, \dots, \xi_N]$ is the vector of slack variables, and C is the misclassification penalty (The optimal C can be obtained by cross-validation).

After optimization, SVM learned weighting $\alpha_i \in \mathbb{R}^N$ on training example in kernel feature space $\phi(\mathbf{x}_i)$, where $\phi : \mathcal{X} \rightarrow \mathcal{H}$ defines a (possible nonlinear) feature mapping from the original input space to a Hilbert space $\mathcal{H}$. The decision function can then be represented by:

$$f(x) = \text{sign}\left(\sum_{i=1}^N y_i \alpha_i^* \mathbf{K}(\mathbf{x}, \mathbf{x}_i) + b^*\right) \quad (3)$$

where kernel function (kernel matrix) $\mathbf{K}(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}')$ is to assess the similarity of input samples, parameter α^* and b^* are obtained by solving the dual of the optimization in (2).

SVM can model the nonlinear distribution of the data by mapping the input data into a nonlinear feature space by kernel embedding[41, 42]. However, the most suitable types and parameters of the kernels for a particular task is often unknown, and the selection of the optimal kernel by exhaustive search on a pre-defined pool of kernels is usually time-consuming, and sometimes causes overfitting. Multiple kernel learning (MKL)[43], which learns the optimal kernel by a weighted, linear combination of predefined candidate kernels, has been introduced to handle the problem of kernel selection. Compared to a single kernel, such as SVM, MKL attempts to achieve better results by combining several base kernels instead of using only one specific kernel. MKL assumes that $\mathbf{x}_i$ can be mapped to Q different Hilbert spaces, $\mathbf{x}_i \rightarrow \phi_q(\mathbf{x}_i)$, $q = 1, \dots, Q$, implicitly with Q nonlinear mapping functions, and the objective of MKL is to seek the optimal kernel combination $\hat{\mathbf{K}}(x, x') = \sum_{q=1}^Q \mathbf{d}_q \mathbf{K}_q(x, x')$, $\mathbf{d}_q \geq 0$, $\sum_{q=1}^Q \mathbf{d}_q = 1$.

We obtain the primal objective function of multiple kernel learning:

$$\begin{aligned} \min_{\mathbf{w}, \mathbf{d}, b, \boldsymbol{\xi}} \quad & \frac{1}{2} \sum_{q=1}^Q \|\mathbf{w}_q\|^2 + C \sum_{i=1}^N \xi_i \\ \text{subject to} \quad & y_i \left(\sum_{q=1}^Q \langle \mathbf{w}_q, \sqrt{\mathbf{d}_q} \phi_q(\mathbf{x}_i) \rangle + b \right) \geq 1 - \xi_i, \quad i = 1, \dots, N \\ & \sum_{q=1}^Q \mathbf{d}_q = 1, \mathbf{d}_q \geq 0, q = 1, \dots, Q \end{aligned} \quad (4)$$

Since equation (4) is non-convex due to the products of $\mathbf{d}_q$ and $\mathbf{w}_q$, it can be resolved by applying the variable transformation $\mathbf{w}'_q = \sqrt{\mathbf{d}_q} \mathbf{w}_q$ as that in [45] and [46].

$$\begin{aligned}
& \min_{\mathbf{w}', \mathbf{d}, b, \xi} \quad \frac{1}{2} \sum_{q=1}^Q \frac{\|\mathbf{w}'_q\|^2}{\mathbf{d}_q} + C \sum_{i=1}^N \xi_i \\
& \text{subject to} \quad y_i \left(\sum_{q=1}^Q \langle \mathbf{w}'_q, \phi_k(\mathbf{x}_i) \rangle + b \right) \geq 1 - \xi_i, \quad i = 1, \dots, n \\
& \quad \sum_{q=1}^Q \mathbf{d}_q = 1, \mathbf{d}_q \geq 0, q = 1, \dots, Q
\end{aligned} \tag{5}$$

In order to obtain the best classification accuracy, MKL learns both the weights of the kernel combination and the parameters of the classifier by solving a single joint optimization problem. Accordingly, the decision function can be reformulated as:

$$f(x) = \text{sign} \left(\sum_{i=1}^N \alpha_i^* y_i \sum_{q=1}^Q \mathbf{d}_q^* \mathbf{K}_q(x, x_i) + b^* \right) \tag{6}$$

where $\mathbf{K}_q(\mathbf{x}, \mathbf{x}') = \phi_q(\mathbf{x}) \cdot \phi_q(\mathbf{x}')$, $\sum_{q=1}^Q \mathbf{d}_q^* = 1$

3.4. $\ell_{2,1}$ norm regularization

The regularization framework is common in machine learning, including Support Vector Machines, logistic regression and Lasso. We consider convex optimization problems of the form:

$$\min_{\mathbf{w}} J = \mathcal{L}(\mathbf{w}) + \lambda \mathcal{R}(\mathbf{w}) \tag{7}$$

where $\mathcal{L}(\cdot)$ denotes the loss function, $\mathcal{R}$ is a sparsity-inducing norm as regularization (typically non-smooth) and λ is a regularization parameter needed to be chosen by cross validation.

A typical regularization is ℓ_2 -norm regularization ($\|\mathbf{w}\|_2$), which is designed to solve ill-posed problems and avoid overfitting problems. Sparse methods have attracted a great amount of research efforts in the past decade due to their sparsity-inducing property, leading to interpretable models by effective feature selecting [59]. ℓ_1 -norm regularization ($\|\mathbf{w}\|_1$) introduces sparse solutions in the feature level.

The $\ell_{2,1}$ -norm, also called group lasso[56], extends the lasso model by introducing the ℓ_1 -norm on the ℓ_2 -norm of the function weights to select the explanatory factors in a group manner. The $\ell_{2,1}$ -norm regularization is $\|\mathbf{w}\|_{2,1} = \sum_{g=1}^G \|\mathbf{w}_g\|_2$, where $\{1, \dots, G\}$ are the groups of variables. It behaves like an ℓ_1 -norm on the vector $\|\mathbf{w}_g\|_2$, and therefore induces group sparsity. In other words, each $\|\mathbf{w}_g\|_2$, and equivalently each $\mathbf{w}_g$, is encouraged to be set to zero. On the other hand, within each group, the ℓ_2 -norm does not promote sparsity. The difference between ℓ_1 -norm (Lasso) and $\ell_{2,1}$ -norm (Group Lasso) are illustrated in Fig. 2.

Figure 2: The illustration of Lasso and Group lasso

3.5. $\ell_{2,1}$ norm balanced multiple kernel learning for feature selection

To perform nonlinear feature selection, a base kernel is computed for each feature (representation) in [44], which may result in one important feature then removed if the kernel function is not set appropriately. In the limit that only a single kernel is used per feature, we associate multiple nonlinear base kernel with each individual feature, hence there is a total of $P \times Q$ base kernels in our $\ell_{2,1}$ norm regularized MKL as shown in Fig. 3 (Q different kernel functions for each feature). The $\ell_{2,1}$ regularizer would enforce the sparsity among the features by ℓ_1 norm, while nonsparsity can be preserved for base kernels within the same feature by ℓ_2 norm. The $\ell_{2,1}$ norm of matrix $\mathbf{D}$ is defined as:

$$\|\mathbf{D}\|_{2,1} = \sum_{p=1}^P \sqrt{\sum_{q=1}^Q d_{pq}^2} = \sum_{p=1}^P \|\mathbf{d}_p\|_2 \quad (8)$$

Because $\ell_{2,1}$ enforces row sparsity on $\mathbf{D}$, the p -th row of $\mathbf{D}$ shrinks to zero if the corresponding feature is irrelevant to the output. The mixed $\ell_{2,1}$ constraint imposed on $\mathbf{D}$ will maintain sparsity between different feature so as to generate a compact set of feature, while the associated values in each group need not be

Figure 3: The $\ell_{2,1}$ norm regularized multi-kernel kernel learning

sparse, allowing our MKL algorithm to select more than one base kernel for each feature, avoiding removal of relevant features.

In order to make the errors from an imbalanced data set comparable and avoid the features selected are prone to ignore the accuracy of minority class, we use the balanced loss function, which takes the weighted average of the misclassification errors of two classes. We split the training set Tr into Tr_{maj} and Tr_{min} , containing the N_- majority and N_+ minority instances, respectively. The objective function of $\ell_{2,1}$ norm regularized balanced MKL is:

$$\begin{aligned}
 \min_{\mathbf{w}', \mathbf{D}, b, \xi} \quad & \frac{1}{2} \sum_{p=1, q=1}^{P, Q} \frac{\|\mathbf{w}'_{pq}\|^2}{d_{pq}} + C \left(\frac{1}{N_+} \sum_{i=1}^{N_+} \xi_i^+ + \frac{1}{N_-} \sum_{i=1}^{N_-} \xi_i^- \right) + C_\gamma \|\mathbf{D}\|_{2,1} \\
 \text{subject to} \quad & y_i \left(\sum_{p=1, q=1}^{P, Q} \langle \mathbf{w}'_{pq}, \phi_{pq}(\mathbf{x}_i^{(p)}) \rangle + b \right) \geq 1 - \xi_i, \quad i = 1, \dots, N \\
 & d_{pq} \geq 0, \quad \forall p, q \\
 & \xi_i \geq 0, \quad \forall i
 \end{aligned} \tag{9}$$

where $\mathbf{x}_i^{(p)}$ indexes the p -th feature of the instance of $\mathbf{x}_i$, C_γ is regularization parameter.

The primal formulation (9) can be seen as the composite objective optimization problem:

$$\min_{\mathbf{D} \geq 0} J(\mathbf{D}) + C_\gamma \|\mathbf{D}\|_{2,1} \quad (10)$$

where

$$J(\mathbf{D}) = \begin{cases} \min_{\mathbf{w}', b, \xi} \frac{1}{2} \sum_{p=1, q=1}^{P, Q} \frac{\|\mathbf{w}'_{pq}\|^2}{d_{pq}} + C \left(\frac{1}{N_+} \sum_{i=1}^{N_+} \xi_i^+ + \frac{1}{N_-} \sum_{i=1}^{N_-} \xi_i^- \right) \\ \text{subject to} & y_i \left(\sum_{p=1, q=1}^{P, Q} \langle \mathbf{w}'_{pq}, \phi_{pq}(\mathbf{x}_i^{(p)}) \rangle + b \right) \geq 1 - \xi_i \quad \forall i \\ & \xi_i \geq 0 \quad \forall i \end{cases} \quad (11)$$

The Lagrangian of problem (11):

$$\begin{aligned} L = & \frac{1}{2} \sum_{p=1, q=1}^{P, Q} \frac{\|\mathbf{w}'_{pq}\|^2}{d_{pq}} + C \left(\frac{1}{N_+} \sum_{i=1}^{N_+} \xi_i^+ + \frac{1}{N_-} \sum_{i=1}^{N_-} \xi_i^- \right) \\ & + \sum_i^N \alpha_i (1 - \xi_i - y_i \sum_{p=1, q=1}^{P, Q} \mathbf{w}'_{pq} \phi_{pq}(\mathbf{x}_i^{(p)}) - y_i b) - \sum_i^N \beta_i \xi_i \end{aligned} \quad (12)$$

where α_i and β_i are the Lagrange multipliers of the constraints.

To obtain the dual objective function we minimize L with respect to the variables $\mathbf{w}, b, \xi$ by setting the partial derivatives to 0,

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}'_{pq}} &= \mathbf{w}'_{pq} - d_{pq} \sum_i^N \alpha_i y_i \phi_{pq}(\mathbf{x}_i^{(p)}) = 0, \quad \forall g, q \\ \frac{\partial L}{\partial b} &= \sum_i^N \alpha_i y_i = 0 \\ \frac{\partial L}{\partial \xi_i^+} &= \frac{C}{N_+} - \alpha_i^+ - \beta_i^+ = 0, \quad \forall i, \quad y_i = 1 \\ \frac{\partial L}{\partial \xi_i^-} &= \frac{C}{N_-} - \alpha_i^- - \beta_i^- = 0, \quad \forall i, \quad y_i = -1 \end{aligned} \quad (13)$$

By setting to zero the derivatives of this Lagrangian according to the primal variables, we derive the following associated dual problem:

$$\max \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \sum_{p=1, q=1}^{P,Q} d_{pq} \mathbf{K}_{pq}(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(p)})$$

$$\frac{C}{N_+} \geq \alpha_i^+ \geq 0 \quad \forall i \quad (14)$$

$$\frac{C}{N_-} \geq \alpha_i^- \geq 0 \quad \forall i \quad (15)$$

$$\sum_i \alpha_i y_i = 0 \quad \forall i$$

Because of strong duality, $J(\mathbf{D})$ is also the objective value of the dual problem:

$$\begin{aligned} J(\mathbf{D}) &= \sum_{i=1}^N \alpha_i^* - \frac{1}{2} \sum_{i,j=1}^N \alpha_i^* \alpha_j^* y_i y_j \sum_{p=1, q=1}^{P,Q} d_{pq} \mathbf{K}_{pq}(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(p)}) \\ &= \sum_{i=1}^N \alpha_i^* - \frac{1}{2} \sum_{i,j=1}^N \alpha_i^* \alpha_j^* y_i y_j \text{tr}(\mathbf{K}^{ij} \mathbf{D}) \end{aligned} \quad (16)$$

where

$$\mathbf{K}^{ij} = \begin{bmatrix} \mathbf{K}_{11}(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(1)}) & \cdots & \mathbf{K}_{1Q}(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(1)}) \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{P1}(\mathbf{x}_i^{(P)}, \mathbf{x}_j^{(P)}) & \cdots & \mathbf{K}_{PQ}(\mathbf{x}_i^{(P)}, \mathbf{x}_j^{(P)}) \end{bmatrix} \quad (17)$$

Eq. (10) is a composite objective, where $J(\mathbf{D})$ is convex and differentiable, while $\|\mathbf{D}\|_{2,1}$ is convex but non-smooth. To optimize $\mathbf{D}$, we efficiently solve it by designing a new accelerated proximal gradient method with the idea of FISTA (fast iterative shrinkage-thresholding algorithm) [23] in this work. A well studied idea in efficient optimization of such composite objective functions is to start with a quadratic approximation of the form:

$$Q_L(\mathbf{D}, \mathbf{D}^{(t)}) := J(\mathbf{D}^{(t)}) + \langle \mathbf{D} - \mathbf{D}^{(t)}, \nabla J(\mathbf{D}^{(t)}) \rangle + \frac{L^{(t)}}{2} \|\mathbf{D} - \mathbf{D}^{(t)}\|^2 + C\gamma \|\mathbf{D}\|_{2,1} \quad (18)$$

where

$$\nabla J(\mathbf{D}^{(t)}) = [\nabla J(d_{11}^{(t)}), \dots, \nabla J(d_{pq}^{(t)}), \dots, \nabla J(d_{PQ}^{(t)})] \quad (19)$$

and

$$\nabla J(d_{pq}^{(t)}) = -\frac{1}{2} \sum_{i,j=1}^N \alpha_i^* \alpha_j^* y_i y_j \mathbf{K}_{pq}(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(p)}) \quad \forall p, q \quad (20)$$

The differentiability of $J(d_{pq}^{(t)})$ is ensured by the uniqueness of α^* and by the differentiability of the objective function. Ignoring constant terms in $\mathbf{D}^{(t)}$, the unique minimizer of the above expression can be written as Eq. (21), by which $\mathbf{D}$ is iteratively updated by proximal gradient method.

$$\begin{aligned} \mathbf{D}^{(t+1)} &= \arg \min_{\mathbf{D}} Q_L(\mathbf{D}, \mathbf{D}^{(t)}) \\ &= \arg \min_{\mathbf{D}} J(\mathbf{D}^{(t)}) + \langle \mathbf{D} - \mathbf{D}^{(t)}, \nabla J(\mathbf{D}^{(t)}) \rangle + \frac{L^{(t)}}{2} \|\mathbf{D} - \mathbf{D}^{(t)}\|^2 + C\gamma \|\mathbf{D}\|_{2,1} \\ &= \arg \min_{\mathbf{D}} \left\| \mathbf{D} - \left(\mathbf{D}^{(t)} - \frac{1}{L^{(t)}} \nabla J(\mathbf{D}^{(t)}) \right) \right\|^2 + \frac{C\gamma}{L^{(t)}} \|\mathbf{D}\|_{2,1} \end{aligned} \quad (21)$$

The minimizer of the above expression can be written as:

$$\mathbf{D}^{(t+1)} = \mathbf{prog}(\mathbf{D}^{(t)}) = \pi_L^{C\gamma}(\mathbf{U}^{(t)}) = \arg \min_{\mathbf{D}} \frac{1}{2} \|\mathbf{D} - \mathbf{U}^{(t)}\|_2^2 + \frac{C\gamma}{L^{(t)}} \|\mathbf{D}\|_{2,1} \quad (22)$$

where $\mathbf{prog}(\mathbf{D}^{(t)})$ can be viewed as a proximal operator corresponding to the non-smooth penalty of $\|\mathbf{D}\|_{2,1}$, $\mathbf{U}^{(t)} = \mathbf{D}^{(t)} - \frac{1}{L^{(t)}} \nabla J(\mathbf{D}^{(t)})$, $L^{(t)}$ plays the role of a stepsize and is determined by the line search. Meanwhile, in order to accelerate the proximal gradient method in Eq. (22), we consider the proximal operator $\mathbf{prog}(\mathbf{D}^{(t)})$ at a specific linear combination of the previous two iterates $\{\mathbf{D}^{(t-1)}, \mathbf{D}^{(t-2)}\}$, instead of at just the previous iterate $\mathbf{D}^{(t-1)}$ as follows:

$$\mathbf{V}^{(t)} = \mathbf{D}^{(t-1)} + \frac{a_{(t)} - 1}{a_{t+1}} (\mathbf{D}^{(t-1)} - \mathbf{D}^{(t-2)}) \quad (23)$$

Thanks to the separability of $\mathbf{D}^{(t+1)}$ on each row, we can update the weights for each row $\mathbf{d}_p^{(t+1)}$ individually:

$$\mathbf{d}_p^{(t+1)} = \arg \min_{\mathbf{d}_p} \frac{1}{2} \|\mathbf{d}_p - \mathbf{u}_p^{(t)}\|_2^2 + \frac{C_\gamma}{L^{(t)}} \|\mathbf{d}_p\|_2 \quad (24)$$

where $\mathbf{d}_p$ and $\mathbf{u}_p$ are the p -th rows of matrix $\mathbf{D}$ and $\mathbf{U}$ respectively.

Following [10], the row-wise updates can be done by soft-thresholding as

$$\mathbf{d}_p^{(t+1)} = \frac{\max \left\{ \|\mathbf{u}_p^{(t)}\|_2 - \frac{C_\gamma}{L^{(t)}}, 0 \right\}}{\|\mathbf{u}_p^{(t)}\|_2} \mathbf{u}_p^{(t)} \quad (25)$$

Moreover, a projection operator $\mathbf{P}$ is introduced ensuring that the non-negativity constraints on $\mathbf{d}_p^{(t+1)}$ ($\mathbf{d}_p \geq 0$) are satisfied.

$$\mathbf{P}(\mathbf{d}_p^{(t+1)}) = \max(\mathbf{0}, \mathbf{d}_p^{(t+1)}) \quad (26)$$

The algorithm for updating kernel weighting $\mathbf{D}$ by the FISTA method is given in Algorithm 1. Because the problem in Eq. (10) is a convex problem, the global optimum solution to the problem can be obtained. Moreover, the Algorithm 1 will monotonically decrease the objective of the problem Eq. (10) in each iteration and converge to the global optimum of the problem. As shown in [23], such an algorithm is guaranteed to converge at a rate $O(1/t^2)$. Through the optimization, the weights $\mathbf{D}$ can be learned to reflect the relative importance of different features. The final discriminant function is:

$$f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^N \alpha_i^* y_i \sum_{p=1, q=1}^{P, Q} d_{pq}^* \mathbf{K}_{pq}(\mathbf{x}^{(p)}, \mathbf{x}_i^{(p)}) + b^* \right) \quad (27)$$

The proposed $\ell_{2,1}$ MKFS actually belongs to embedded feature selection if oversampling is not conducted since the model determines the feature subset during multi-kernel classifier construction. Since the regularization parameters C and C_γ need to be selected during the training, we use 5-fold cross validation on the training data to select the parameters.

Algorithm 1 FISTA Method ($\ell_{2,1}$ MKL) for optimizing $\mathbf{D}$ in Eq. (10)

Input: Training Data X , class label Y , $L_0 \geq 0$, $a^{(1)} = 1$, C and C_γ

Output: Weights of kernel $\mathbf{D}$

- 1: Initialize $\mathbf{D}^0 = [d_{11}, \dots, d_{PQ}]$ with $d_{pq} \in [0, 1]$
 - 2: $\mathbf{V}^0 = \mathbf{D}^0$
 - 3: **for** $t = 1$ to $\dots$ **do**
 - 4: Find the smallest nonnegative integers i_t such that with $L^{(t)} = 2^{i_t} L^{(t-1)}$
 $J(\pi_L^{C_\gamma}(\mathbf{V}^{(t)})) \leq Q_L(\mathbf{V}, \mathbf{V}^{(t)})$, and set $L^{(t)} = 2^{i_t} L^{(t-1)}$
 - 5: Calculate $J(\mathbf{D}^{(t)})$ by using an SVM solver with X, Y and combined kernel
 $\sum_{p=1, q=1}^{P, Q} d_{pq} \mathbf{K}_{pq}$ according to Eq. (16), and $\nabla J(\mathbf{D}^{(t)})$ according to Eq.
(19)
 - 6: **for** $p = 1$ to P **do**
 - 7: $\mathbf{u}_p^{(t)} = \mathbf{v}_p^{(t)} - \frac{1}{L^{(t)}} \nabla J(\mathbf{v}_p^{(t)})$
 - 8: Compute $\mathbf{d}_p^{(t)} = \arg \min_{\mathbf{d}_p} \frac{1}{2} \left\| \mathbf{d}_p - \mathbf{u}_p^{(t)} \right\|_2^2 + \frac{C_\gamma}{L^{(t)}} \|\mathbf{d}_p\|_2$ by solving Eq.
(25)
 - 9: $\mathbf{d}_p^{(t)} = \mathbf{P}(\mathbf{d}_p^{(t)})$
 - 10: $a^{(t+1)} = \frac{(1 + \sqrt{1 + 4(a^{(t)})^2})}{2}$
 - 11: $\mathbf{v}_p^{(t+1)} = \mathbf{d}^{(t)} + \frac{a^{(t)} - 1}{a^{(t+1)}} (\mathbf{d}_p^{(t)} - \mathbf{d}_p^{(t-1)})$
 - 12: **end for**
 - 13: **if** convergence **then**
 - 14: $\mathbf{D} = \mathbf{D}^{(t+1)}$
 - 15: **end if**
 - 16: **end for**
-

3.6. Multi-kernel oversampling

The methods with the data perspective re-balance the class distribution by re-sampling the data space, either oversampling instances of the minority class or undersampling instances of the majority class. While many re-sampling methods have been proposed in the literature, which target only the characteristic of the imbalanced distribution between classes, there still exist two unsolved key issues:

(1) The common re-sampling for kernel methods (e.g. SVM) is not appropriate since the preprocessing methods operate in the original input space, introducing distortions and inconsistency when combined with kernel classifiers that operate in the feature space induced by a kernel function.

(2) The existing re-sampling is conducted on the whole feature set. However,

the irrelevant and redundant features influence the mechanism of oversampling.

To address the issue of traditional oversampling, we propose a multi-kernel over-sampling (MKOS), which is carried out in the feature space after feature selecting, hence it can alleviate the negative influence due to the irrelevant and redundant features. More, the proposed oversampling can work in the same feature space as kernel classifier. Specifically, we extend the original SMOTE algorithm into feature space by using the optimal kernel matrix optimized by $\ell_{2,1}$ MKFS. Utilizing the strategy of SMOTE procedure, MKOS generates artificial example $\mathbf{x}_{ij}^{new}$ located on the path connecting selected minority example $\phi(\mathbf{x}_i)$ and one of its closest neighbor $\phi(\mathbf{x}_j)$ in the feature space rather than the original input space:

$$\phi(\mathbf{x}_{ij}^{new}) = \phi(\mathbf{x}_i) + \lambda_{ij} \times (\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)) \quad (28)$$

The idea of the proposed multi kernel oversampling is shown in Fig. 4.

Figure 4: An illustrative toy example of the proposed MKOS algorithm. In (a), x_{ij}^{new1} is the synthetic instances based on x_i and x_j by SMOTE in the original input space, resulting in invading the majority distribution and generating inaccurate instances since the data has less clearly separable property in the original input space. In (b), all the samples are mapped into a feature space by a appropriate kernel mapping ϕ^* optimized by $\ell_{2,1}$ MKFS. Then, the new instance $\phi^*(x_{ij}^{new2})$ is generated between $\phi^*(x_i)$ and $\phi^*(x_j)$. Finally, we find the pre-image instance of $\phi^*(x_{ij}^{new2})$ based on distance constraints. Moreover, the $\phi^*(x_{ij}^{new2})$ mapping from x_{ij}^{new2} is located in the opposite side, resulting in that the instance generated in original input space is not necessarily the appropriate one in feature space. Oversampling in the feature space not only generate more accurate samples due to more separable distribution provided by optimal kernel mapping, but also be consistent with the building of SVM classifier.

After creating artificial data based on the feature space similarities between existing minority examples according to Eq. (28) we need to find the pre-image of the artificial data in the input space. Since we choose the RBF kernel: $K(\mathbf{x}, \mathbf{x}') = \exp(\frac{-\|\mathbf{x}-\mathbf{x}'\|}{2\sigma^2})$ in our framework, the relationship between feature-space distance $d_{fea}(\phi(\mathbf{x}_{ij}^{new}), \phi(\mathbf{x}'))$ and input-space distance $d_{in}(\mathbf{x}_{ij}^{new}, \mathbf{x}')$ is:

$$\begin{aligned}
 d_{fea}^2(\phi(\mathbf{x}_{ij}^{new}), \phi(\mathbf{x}')) &= \|\phi(\mathbf{x}_{ij}^{new}) - \phi(\mathbf{x}')\|^2 \\
 &= \hat{K}^*(\mathbf{x}_{ij}^{new}, \mathbf{x}_{ij}^{new}) - 2\hat{K}^*(\mathbf{x}_{ij}^{new}, \mathbf{x}') + \hat{K}^*(\mathbf{x}', \mathbf{x}') \\
 &= 2 - 2\exp(-\|\mathbf{x}_{ij}^{new} - \mathbf{x}'\|/(2\sigma^2)) \\
 &= 2 - 2\exp(-d_{in}^2(\mathbf{x}_{ij}^{new}, \mathbf{x}')/(2\sigma^2))
 \end{aligned} \tag{29}$$

According to equation (28), the distance between the new instance $\mathbf{x}_{ij}^{new}$ and any instance $\mathbf{x}'$.

$$\begin{aligned}
d_{fea}^2(\phi(\mathbf{x}_{ij}^{new}), \phi(\mathbf{x}')) &= \|\phi(\mathbf{x}_{ij}^{new}) - \phi(\mathbf{x}')\|^2 \\
&= d_{fea}^2(\phi(\mathbf{x}_i) + \lambda_{ij} \times (\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)), \mathbf{x}') \\
&= \|\phi(\mathbf{x}_i) + \lambda \times (\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)), \mathbf{x}'\|^2 \\
&= \mathbf{K}^*(\mathbf{x}', \mathbf{x}') - 2(\lambda_{ij} - 1)\mathbf{K}^*(\mathbf{x}', \mathbf{x}_i) - 2\lambda_{ij}\mathbf{K}^*(\mathbf{x}', \mathbf{x}_j) \\
&\quad + (\lambda_{ij} - 1)^2\mathbf{K}^*(\mathbf{x}_i, \mathbf{x}_i) + 2\lambda_{ij}(1 - \lambda_{ij})\mathbf{K}^*(\mathbf{x}_i, \mathbf{x}_j) \\
&\quad + \lambda_{ij}^2\mathbf{K}^*(\mathbf{x}_j, \mathbf{x}_j)
\end{aligned} \tag{30}$$

where $\mathbf{K}^*$ is the combined kernel matrix optimized by $\ell_{2,1}$ MKFS.

A unique pre-image $\hat{\mathbf{x}}_{ij}^{new}$ can be estimated with its K_2 nearest neighbors in the feature space according to [40] as follows:

$$\begin{aligned}
\hat{\mathbf{x}}_{ij}^{new} &= \frac{\sum_{g=1}^{K_2} \gamma_g \exp(-\frac{\|\mathbf{x}_{ij}^{new} - \mathbf{x}'_g\|^2}{2\sigma^2}) \mathbf{x}'_g}{\sum_{g=1}^{K_2} \gamma_g \exp(-\frac{\|\mathbf{x}_{ij}^{new} - \mathbf{x}'_g\|^2}{2\sigma^2})} \\
&= \frac{\sum_{g=1}^{K_2} \gamma_g \exp(-\frac{d_{in}^2(\mathbf{x}_{ij}^{new}, \mathbf{x}'_g)}{2\sigma^2}) \mathbf{x}'_g}{\sum_{g=1}^{K_2} \gamma_g \exp(-\frac{d_{in}^2(\mathbf{x}_{ij}^{new}, \mathbf{x}'_g)}{2\sigma^2})} \\
&= \frac{\sum_{g=1}^{K_2} \gamma_g (\frac{1}{2}(2 - d_{fea}^2(\phi(\mathbf{x}_{ij}^{new}), \phi(\mathbf{x}'_g)))) \mathbf{x}'_g}{\sum_{g=1}^{K_2} \gamma_g (\frac{1}{2}(2 - d_{fea}^2(\phi(\mathbf{x}_{ij}^{new}), \phi(\mathbf{x}'_g))))}
\end{aligned} \tag{31}$$

where $d_{fea}^2(\phi(\mathbf{x}_{ij}^{new}), \phi(\mathbf{x}'_g))$ can be calculated according to Eq. (30), γ_g is a normalized value of distance between $\phi(\mathbf{x}_{ij}^{new})$ and $\phi(\mathbf{x}'_g)$ according to 28. The MKOS algorithm is detailed in Algorithm 2.

3.7. Time complexity

For MKFS, in each iteration of algorithm 1, we need to solve a canonical SVM problem with the $\ell_{2,1}$ kernel weights optimized by a gradient descent method. The time complexity of the gradient calculation is ignorable compared to the SVM solver. Each SVM training scales in $O(n_{sv}^3)$ with n_{sv} being the number of support vectors, while the complexity of the gradient computation of $\mathbf{d}$ is of the order

Algorithm 2 Multi kernel Over-Sampling

Input:

Training data set Tr , Oversampling ratio R , Number of nearest neighbors K_1 for oversampling, Number of nearest neighbors K_2 for finding pre-image, Number of new instances for seed instance M

Output:

The new balanced dataset Tr'

- 1: $Tr_{new} = NULL$
 - 2: $M = (int)(R/100)$
 - 3: Obtain the minority class data set Tr_{min} from Tr
 - 4: **for** $i = 1; i \leq |Tr_{min}|; i++$ **do**
 - 5: $S = NULL$
 - 6: $NN_1 = findKNN(\phi(\mathbf{x}_i), K_1, Tr_{min})$ *Function to find the K nearest neighbors*
 - 7: **while** $M > 0$ **do**
 - 8: Randomly choose one minority data example from NN_1 , $\phi(\mathbf{x}_j)$
 - 9: $NN_2 = findKNN(\phi(\mathbf{x}_{ij}^{new}), K_2, Tr_{min})$ according to Eq. (30)
 - 10: Obtain the pre-image $\mathbf{x}_{ij}^{new}$ with nearest neighbors NN_2 according to Eq. (31)
 - 11: $S = S \cup \mathbf{x}_{ij}^{new}$
 - 12: $M = M - 1$
 - 13: **end while**
 - 14: $Tr_{new} = Tr_{new} \cup S_i$
 - 15: **end for**
 - 16: $Tr' = Tr \cup Tr_{new}$
-

Algorithm 3 findKNN

Input:

Seed instance $\phi(\mathbf{x})$, Number of nearest neighbors K , Data set Tr

Output:

The set of nearest neighbors KNN

- 1: **for** each instance $\phi(\mathbf{x}_r)$ in Tr **do**
 - 2: Calculate the distance dis_r between $\phi(\mathbf{x}_r)$ and seed $\phi(\mathbf{x})$ in the feature space
 - 3: **end for**
 - 4: Sort DIS in descending order
 - 5: $KNN = getFirstInstances(DIS, K)$
-

of $O(P \times Q \times n_{sv}^2)$, where P is feature size and Q is the amount of the kernel functions for each feature. Hence, the total time complexity of MKOS is $O(n_{sv}^3)$. However, in the optimization procedure of $\ell_{2,1}$ MKFS, warm-start technique is used for successive SVM retrainings to accelerate the training process. The gradient based approach produces estimates of $\mathbf{D}$ on a smooth trajectory, so that the previous SVM solution provides a good guess (i.e., providing previous as input) for the current SVM training.

For MKOS, the overall procedure consists of three parts: 1) Searching the nearest neighbors of all minority instances is $O(N_-^2)$, where N_- is the minority class size. 2) Obtaining the nearest neighbors of new generated instances is $O(N_-^2 \times M)$. 3) Estimating the pre-imaging is $O(N_- \times M \times K_2)$. Thus the total time complexity of MKOS is $O(N_-^2) + O(N_-^2 \times M) + O(N_- \times M \times K_2) = O(N_-^2)$.

4. Experimental study

In this section, we present experimental results to validate and understand the performance of $\ell_{2,1}$ MKFSOS on imbalanced datasets. At first, we evaluated the effectiveness of our method on the classification of different datasets from the UCI database by comparing with other methods specifically devised for imbalanced data. Moreover, we used real data sets selected from two different application domains including: lung nodule detection and Alzheimer’s disease diagnosis. These two real world applications nicely illustrates the effectiveness of $\ell_{2,1}$ MKFSOS, the combination of feature selection and oversampling schemes.

4.1. Evaluating the effectiveness of feature selection for imbalanced data on UCI datasets

4.1.1. Dataset and Parameter Settings

We chosen 20 different UCI datasets with over 25 dimensionality and different degrees of class imbalance. We used all available datasets from the combined sets used in [28, 35, 47], where kernel classifier is also researched for imbalanced data learning. This ensures that we did not choose only the datasets on which our method performs better. Moreover, we added 8 additional datasets with different properties for a more comprehensive evaluation. Table 1 summarizes the relevant information for each data set.

All the experiments are carried out by means of a 10-fold stratified cross-validation. That is, the dataset was split into ten folds, each one containing 10% of the patterns of the dataset. For each fold, the algorithm is trained with the examples contained in the remaining folds and then tested with the current fold.

Table 1: Description of data set characteristics used in experiments

Dataset	Instances	Features	Minority Class ratio
Abalone	4177	8	1%
Ecoli	281	7	2%
Oil	937	49	4%
Car	1728	6	4%
Letter	20000	16	4%
Hypothyroid	3163	25	5%
Sick	3772	29	6%
Soybean	683	35	6%
Anneal	898	38	8%
Satimage	6435	36	10%
Zernike	2000	47	10%
Glass	214	9	14%
Segment	2310	19	14%
Hepatitis	155	19	20%
Wdbc	198	33	24%
Haberman	306	3	27%
Ionosphere	351	33	36%
Wdbc	569	32	37%
Spambase	4601	57	40%
Sonar	208	60	47%

For the $\ell_{2,1}$ MKFS, the regularization parameters C and C_γ in Eq. (9) are chosen by nested cross-validation strategy on the training data (trying values 0.01, 0.1, 1, 10, 100). The commonly used kernel functions are BRF kernel, and we use BRF kernels with six different kernel bandwidths ($2^{-2}, 2^{-1}, \dots, 2^3$) for each feature, which totally yields $Q \times P$ kernels (P is the number of dimension, and $Q = 6$). **Each base kernel matrix is normalized to unit trace.** The stopping criterion we have used are the following. For our alternate optimization methods, we stop when $\max(|\mathbf{D}^{(t+1)} - \mathbf{D}^{(t)}|) < 0.001$. For MKOS, there are three parameters need to be set:

K_1 : number of nearest neighbors used to generate the synthetic data instances

K_2 : number of nearest neighbors in estimating the pre-image of the new synthetic instances in the feature space.

R : Over-sampling ratio.

In our experiments, the parameters are: $K_1 = 10, K_2 = 10, R = 200\%$. All these values are chosen after some preliminary runs and they show the best results but not to be optimal for each dataset. It is known that the use of overall accuracy

is not an appropriate evaluation measure for imbalanced data, therefore the metric for representing the performance of each classifier is chosen to be G-mean and AUC.

The experimentation to our proposed $\ell_{2,1}$ MKFSOS involves four stages, which are designed to explore the performance of our $\ell_{2,1}$ MKFSOS method:

Experiment I: The proposed algorithm is an extension of multi-kernel, feature selection, and over-sampling. To exhibit the performance of each component of our proposed approach, the comparison is conducted between our method and these baseline methods as well as basic SVM in order to experimentally validate the effectiveness;

Experiment II: Our proposed method can be extended to other forms with different regularization and kernel construction of MKL. To examine the performance of different strategies of our method for imbalanced datasets, we compared these variants. This can empirically show which strategy is a more appropriate strategy for imbalanced data with different characteristic;

Experiment III: In order to verify the benefits of feature selection to oversampling, we apply our $\ell_{2,1}$ MKFS to other commonly used oversampling algorithms, and compare them on the whole feature set and reduced feature subset, respectively;

Experiment IV: We investigate the performance of $\ell_{2,1}$ MKFSOS compared to seven state-of-the-art methods. The purpose of this experiment is to show that the $\ell_{2,1}$ MKFSOS method is a competitive approach when compared to other state-of-the-art re-sampling or cost sensitive learning methods specifically designed for dealing with imbalanced datasets.

4.1.2. Experiment I: The comparison with the baseline methods

The goal of this experiment is to analyze how the $\ell_{2,1}$ MKFSOS improves the classification for SVM on the imbalanced data, and show the contribution of each component (kernel based feature selection and kernel based oversampling) in the multi-kernel framework. We compare our $\ell_{2,1}$ MKFSOS with the basic SVM, basic MKL, kernel based oversampling (KOS) and kernel based feature selection (MKFS). The intrinsic parameters of basic SVM is optimized by nested cross validation. For basic MKL and MKFS, we use the implementation of SimpleMKL[14] with the same kernel setting as $\ell_{2,1}$ MKFSOS. The KOS and MKFS are conducted after the training of the basic SVM and MKL, respectively.

Table 2: The performance of the comparable baseline methods on multiple data sets with respect to G-mean and AUC

Dataset	Metric	SVM	MKL	KOS	MKFS	$\ell_{2,1}$ MKFSOS
Abalone	G-mean	0.616 $\pm$ 0.025	0.667 $\pm$ 0.031	0.849 $\pm$ 0.054	0.672 $\pm$ 0.038	0.871$\pm$0.019
	AUC	0.638 $\pm$ 0.016	0.682 $\pm$ 0.020	0.847 $\pm$ 0.042	0.691 $\pm$ 0.042	0.915$\pm$0.033
Ecoli	G-mean	0.829 $\pm$ 0.011	0.829 $\pm$ 0.008	0.871 $\pm$ 0.033	0.855 $\pm$ 0.028	0.879$\pm$0.025
	AUC	0.848 $\pm$ 0.025	0.856 $\pm$ 0.018	0.882 $\pm$ 0.030	0.871 $\pm$ 0.024	0.890$\pm$0.022
Oil	G-mean	0.714 $\pm$ 0.059	0.722 $\pm$ 0.042	0.787 $\pm$ 0.064	0.751 $\pm$ 0.057	0.829$\pm$0.066
	AUC	0.727 $\pm$ 0.032	0.751 $\pm$ 0.028	0.804 $\pm$ 0.047	0.782 $\pm$ 0.041	0.951$\pm$0.033
Car	G-mean	0.929 $\pm$ 0.008	0.945 $\pm$ 0.011	0.961 $\pm$ 0.018	0.977 $\pm$ 0.028	0.992$\pm$0.015
	AUC	0.990 $\pm$ 0.005	0.992 $\pm$ 0.004	0.992 $\pm$ 0.005	0.995 $\pm$ 0.002	1.000$\pm$0.000
Letter	G-mean	0.934 $\pm$ 0.051	0.940 $\pm$ 0.033	0.951 $\pm$ 0.056	0.947 $\pm$ 0.037	0.988$\pm$0.008
	AUC	0.898 $\pm$ 0.027	0.904 $\pm$ 0.009	0.942 $\pm$ 0.025	0.938 $\pm$ 0.014	0.997$\pm$0.003
Hypothyroid	G-mean	0.341 $\pm$ 0.105	0.329 $\pm$ 0.089	0.872 $\pm$ 0.079	0.446 $\pm$ 0.077	0.978$\pm$0.013
	AUC	0.855 $\pm$ 0.046	0.851 $\pm$ 0.032	0.909 $\pm$ 0.026	0.860 $\pm$ 0.044	0.962$\pm$0.017
Sick	G-mean	0.525 $\pm$ 0.071	0.550 $\pm$ 0.066	0.617 $\pm$ 0.082	0.577 $\pm$ 0.074	0.753$\pm$0.055
	AUC	0.743 $\pm$ 0.052	0.770 $\pm$ 0.029	0.822 $\pm$ 0.037	0.781 $\pm$ 0.025	0.911$\pm$0.020
Soybean	G-mean	0.942 $\pm$ 0.022	0.944 $\pm$ 0.019	0.978 $\pm$ 0.008	0.969 $\pm$ 0.025	0.998$\pm$0.018
	AUC	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000
Anneal	G-mean	0.662 $\pm$ 0.058	0.681 $\pm$ 0.043	0.780 $\pm$ 0.063	0.721 $\pm$ 0.024	0.991$\pm$0.005
	AUC	0.876 $\pm$ 0.055	0.910 $\pm$ 0.024	0.943 $\pm$ 0.031	0.955 $\pm$ 0.010	1.000$\pm$0.000
Satimage	G-mean	0.751 $\pm$ 0.031	0.772 $\pm$ 0.046	0.767 $\pm$ 0.055	0.788 $\pm$ 0.035	0.879$\pm$0.029
	AUC	0.860 $\pm$ 0.011	0.866 $\pm$ 0.008	0.866 $\pm$ 0.012	0.884 $\pm$ 0.021	0.944$\pm$0.014
Zernike	G-mean	0.495 $\pm$ 0.081	0.535 $\pm$ 0.075	0.725 $\pm$ 0.061	0.591 $\pm$ 0.075	0.865$\pm$0.059
	AUC	0.733 $\pm$ 0.041	0.767 $\pm$ 0.034	0.830 $\pm$ 0.028	0.772 $\pm$ 0.033	0.915$\pm$0.032
Glass	G-mean	0.745 $\pm$ 0.033	0.766 $\pm$ 0.019	0.828 $\pm$ 0.044	0.791 $\pm$ 0.027	0.919$\pm$0.014
	AUC	0.952 $\pm$ 0.018	0.953 $\pm$ 0.022	0.969 $\pm$ 0.017	0.958 $\pm$ 0.023	0.998$\pm$0.001
Segment	G-mean	0.995 $\pm$ 0.003	1.000$\pm$0.000	0.999 $\pm$ 0.001	1.000$\pm$0.000	1.000$\pm$0.000
	AUC	0.998 $\pm$ 0.001	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000
Hepatitis	G-mean	0.531 $\pm$ 0.057	0.528 $\pm$ 0.022	0.608 $\pm$ 0.037	0.559 $\pm$ 0.034	0.828$\pm$0.021
	AUC	0.632 $\pm$ 0.039	0.629 $\pm$ 0.034	0.677 $\pm$ 0.020	0.717 $\pm$ 0.028	0.887$\pm$0.013
Wdbc	G-mean	0.552 $\pm$ 0.061	0.559 $\pm$ 0.040	0.549 $\pm$ 0.052	0.611 $\pm$ 0.028	0.687$\pm$0.029
	AUC	0.621 $\pm$ 0.030	0.635 $\pm$ 0.029	0.616 $\pm$ 0.025	0.681 $\pm$ 0.032	0.778$\pm$0.022
Haberman	G-mean	0.711 $\pm$ 0.013	0.742 $\pm$ 0.008	0.806 $\pm$ 0.039	0.745 $\pm$ 0.016	0.840$\pm$0.035
	AUC	0.776 $\pm$ 0.009	0.794 $\pm$ 0.011	0.797 $\pm$ 0.017	0.794 $\pm$ 0.008	0.820$\pm$0.012
Ionosphere	G-mean	0.714 $\pm$ 0.044	0.750 $\pm$ 0.032	0.751 $\pm$ 0.044	0.786 $\pm$ 0.018	0.822$\pm$0.019
	AUC	0.787 $\pm$ 0.024	0.799 $\pm$ 0.017	0.793 $\pm$ 0.016	0.830 $\pm$ 0.023	0.929$\pm$0.011
Wdbc	G-mean	0.921 $\pm$ 0.013	0.936 $\pm$ 0.022	0.929 $\pm$ 0.024	0.949 $\pm$ 0.011	0.975$\pm$0.013
	AUC	0.942 $\pm$ 0.010	0.952 $\pm$ 0.008	0.944 $\pm$ 0.015	0.961 $\pm$ 0.007	0.997$\pm$0.002
Spambase	G-mean	0.842 $\pm$ 0.021	0.838 $\pm$ 0.034	0.833 $\pm$ 0.025	0.896 $\pm$ 0.039	0.929$\pm$0.020
	AUC	0.968 $\pm$ 0.019	0.954 $\pm$ 0.012	0.855 $\pm$ 0.017	0.937 $\pm$ 0.025	0.994$\pm$0.005
Sonar	G-mean	0.740 $\pm$ 0.035	0.725 $\pm$ 0.028	0.767 $\pm$ 0.022	0.761 $\pm$ 0.011	0.784$\pm$0.019
	AUC	0.749 $\pm$ 0.031	0.731 $\pm$ 0.027	0.749 $\pm$ 0.034	0.855 $\pm$ 0.022	0.873$\pm$0.014

From Table 2, we can see that $\ell_{2,1}$ MKFSOS, KOS and MKFS improve basic SVM 30.67% (G-mean) and 14.73% (AUC), 14.47% (G-mean) and 4.56% (AUC), and 7.53% (G-mean) and 4.55% (AUC), respectively. In particular, a big improvement is made on the datasets where the imbalance ratios and dimensionality are large, such as Hypothyroid and Zernike. The observation proves that SVM is ineffectual for the class imbalance problem. In some datasets (Abalone, Hypothyroid, Sick, Anneal, and Zernike), SVM performs quite poorly for handling imbalance. In addition, MKL improves over SVM 2.07% (G-mean) and 1.4% (AUC), and we found SVM outperformed MKL over four data sets (Hypothyroid, Hepatitis, Spambase, Sonar)

It is noteworthy that the proposed method consistently showed the best performance over the baseline methods. The proposed method improved on average by 28.26% (G-mean) and 13.04% (AUC) compared to MKL without feature selection and oversampling, by 26.65% (G-mean) and 11.58% (AUC) compared to KOS with only oversampling, and by 20.02% (G-mean) and 9.54% (AUC) compared to MKFS with only feature selection.

As expected, feature selection or oversampling has a positive impact on the performance of SVM built on the imbalanced data from the results. The results also imply that $\ell_{2,1}$ MKOSFS is able to improve the classification performance of SVM on the imbalanced data learning with jointly considering two aspects of feature selection and oversampling. Besides SVM, MKL also suffers from the imbalanced data distribution, and MKL is not always better than SVM. Although it seems a little bit surprising, the similar observation was reported in some previous empirical study [29], which also found that regular MKL does not always outperform SVM with an RBF kernel which kernel parameter was tuned by cross validation. This observation validates our motivation for overcoming the shallow limitation of the regular MKL methods.

4.1.3. Experiment II: The comparison with the variants of proposed method

There are four variants of our proposed ℓ_1 MKFSOS:

- ℓ_1 MKFSOS: $\ell_{2,1}$ norm is replaced by ℓ_1 norm to incorporated into MKFSOS and each feature corresponds one linear kernel.
- $\ell_{2,1}$ MKFS: only feature selection without oversampling.
- MKOS-Single: only oversampling on the whole features without feature selection, and each feature corresponds multiple kernel.

Table 3: The difference among the comparable multi-kernel learning methods with respect to the regularization of kernel weights, kernel function and features for each kernel matrix

Method	Regularization of kernel weights	Kernel function	Features for constructing each kernel matrix
MKOS-Single	$\sum_{p=1}^{P \times Q} \ \mathbf{d}_p\ _2$	RBF	each feature for Q matrices, totally $P \times Q$ matrices
MKOS-All	$\sum_{q=1}^Q \ \mathbf{d}_q\ _2$	RBF	P features for each matrix, totally Q matrices
$\ell_{2,1}$ MKFSOS($\ell_{2,1}$ MKFS)	$\sum_{p=1}^P \sqrt{\sum_{q=1}^Q d_{pq}^2}$	RBF	each feature for Q matrices, totally $P \times Q$ matrices
ℓ_1 MKFSOS	$\sum_{p=1}^P \ \mathbf{d}_p\ $	Linear	each feature for each matrix, totally P matrices

- MKOS-ALL: only oversampling on the whole features without feature selection, and all the features share the same multiple kernels.

The regularization of kernel weights, kernel function and features for each kernel matrix for the variants of our proposed method are shown in Table 3. Actually, we tested the effectiveness of our kernel oversampling operated in four different feature space. Prediction performance results, measured by G-mean and AUC of $\ell_{2,1}$ MKFSOS and variant methods are shown in Table 4.

In order to evaluate the significance of the results, we performed a statistical analysis of our results. Following Demsars recommendation[52], we first tested if there is any significant difference among the classification methods studied with Friedman test. By applying Friedman test, we concluded that there is a significant difference among the methods. Since the null hypothesis is rejected we have to proceed with further analysis to better understand the behavior of the classification algorithms. A series of Wilcoxon signed-ranks tests from Table 4 are listed in Table 5. It also shows the count of wins, ties, losses and corresponding p-value for pairwise post-hoc comparisons.

From Table 4 and Table 5, it can be seen that the performance of $\ell_{2,1}$ MKFSOS is significantly better than the other four variants. For the two MKOS algorithms, no significantly difference exist, but MKOS-single is better than MKOS-ALL in 13(8) of 20 datasets in terms of G-mean(AUC). From results, we can also notice that our method is performing well in the case of a balanced dataset such as Sonar. Moreover, $\ell_{2,1}$ MKFSOS cannot perform well on the datasets with lower dimensionality, such as Haberman with 3 attributes and Ecoli with 7 attributes.

From these results, the following conclusions about the performance of each algorithm may be drawn:

1. Both ℓ_1 MKFSOS and $\ell_{2,1}$ MKFSOS carry out the feature selection and oversampling at the same time based on the multiple kernel framework. The results demonstrates that the features selected by $\ell_{2,1}$ -norm is more relevant to the classification performance than ℓ_1 -norm. The performance of feature selection in ℓ_1

Table 4: The performance of the multiple variant methods on multiple data sets with respect to G-mean and AUC

Dataset	Metric	$\ell_{2,1}$ MKFS	ℓ_1 MKFSOS	MKOS-Single	MKOS-All	$\ell_{2,1}$ MKFSOS
Abalone	G-mean	0.679 $\pm$ 0.029	0.857 $\pm$ 0.017	0.865 $\pm$ 0.013	0.861 $\pm$ 0.023	0.871$\pm$0.019
	AUC	0.705 $\pm$ 0.018	0.882 $\pm$ 0.025	0.891 $\pm$ 0.017	0.898 $\pm$ 0.024	0.915$\pm$0.033
Ecoli	G-mean	0.865 $\pm$ 0.033	0.877 $\pm$ 0.029	0.887$\pm$0.019	0.875 $\pm$ 0.026	0.879 $\pm$ 0.025
	AUC	0.875 $\pm$ 0.025	0.886 $\pm$ 0.036	0.902$\pm$0.019	0.893 $\pm$ 0.022	0.890 $\pm$ 0.022
Oil	G-mean	0.738 $\pm$ 0.053	0.815 $\pm$ 0.049	0.780 $\pm$ 0.051	0.784 $\pm$ 0.033	0.829$\pm$0.066
	AUC	0.853 $\pm$ 0.042	0.944 $\pm$ 0.033	0.916 $\pm$ 0.035	0.933 $\pm$ 0.029	0.951$\pm$0.033
Car	G-mean	0.969 $\pm$ 0.023	0.995$\pm$0.016	0.979 $\pm$ 0.022	0.979 $\pm$ 0.015	0.992 $\pm$ 0.015
	AUC	0.995 $\pm$ 0.002	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000
Letter	G-mean	0.961 $\pm$ 0.011	0.977 $\pm$ 0.011	0.975 $\pm$ 0.005	0.968 $\pm$ 0.012	0.988$\pm$0.008
	AUC	0.948 $\pm$ 0.017	0.985 $\pm$ 0.009	0.973 $\pm$ 0.011	0.977 $\pm$ 0.015	0.997$\pm$0.003
hypothyroid	G-mean	0.369 $\pm$ 0.036	0.955 $\pm$ 0.021	0.977 $\pm$ 0.017	0.962 $\pm$ 0.021	0.978$\pm$0.013
	AUC	0.940 $\pm$ 0.027	0.951 $\pm$ 0.031	0.942 $\pm$ 0.029	0.929 $\pm$ 0.022	0.962$\pm$0.017
Sick	G-mean	0.569 $\pm$ 0.029	0.750 $\pm$ 0.040	0.786 $\pm$ 0.035	0.792$\pm$0.044	0.753 $\pm$ 0.055
	AUC	0.827 $\pm$ 0.015	0.903 $\pm$ 0.033	0.896 $\pm$ 0.029	0.921$\pm$0.023	0.911 $\pm$ 0.020
Soybean	G-mean	0.978 $\pm$ 0.005	0.988 $\pm$ 0.002	0.981 $\pm$ 0.005	0.981 $\pm$ 0.004	0.998$\pm$0.018
	AUC	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000
Anneal	G-mean	0.774 $\pm$ 0.024	0.978 $\pm$ 0.008	0.966 $\pm$ 0.011	0.935 $\pm$ 0.020	0.991$\pm$0.005
	AUC	0.978 $\pm$ 0.007	1.000$\pm$0.000	1.000$\pm$0.000	0.982 $\pm$ 0.002	1.000$\pm$0.000
Satimage	G-mean	0.839 $\pm$ 0.009	0.862 $\pm$ 0.016	0.847 $\pm$ 0.025	0.851 $\pm$ 0.022	0.879$\pm$0.029
	AUC	0.891 $\pm$ 0.007	0.936 $\pm$ 0.007	0.917 $\pm$ 0.015	0.923 $\pm$ 0.019	0.944$\pm$0.014
Zernike	G-mean	0.556 $\pm$ 0.033	0.811 $\pm$ 0.028	0.852 $\pm$ 0.025	0.834 $\pm$ 0.031	0.865$\pm$0.059
	AUC	0.829 $\pm$ 0.033	0.855 $\pm$ 0.027	0.876 $\pm$ 0.035	0.868 $\pm$ 0.033	0.915$\pm$0.032
Glass	G-mean	0.817 $\pm$ 0.019	0.955 $\pm$ 0.023	0.907 $\pm$ 0.019	0.878 $\pm$ 0.028	0.919$\pm$0.014
	AUC	0.988 $\pm$ 0.003	0.991 $\pm$ 0.003	0.983 $\pm$ 0.009	0.983 $\pm$ 0.011	0.998$\pm$0.001
Segment	G-mean	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	0.999 $\pm$ 0.001	1.000$\pm$0.000
	AUC	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000
Hepatitis	G-mean	0.581 $\pm$ 0.033	0.811 $\pm$ 0.024	0.775 $\pm$ 0.025	0.771 $\pm$ 0.014	0.828$\pm$0.021
	AUC	0.835 $\pm$ 0.017	0.829 $\pm$ 0.009	0.811 $\pm$ 0.016	0.815 $\pm$ 0.024	0.887$\pm$0.013
Wpbc	G-mean	0.572 $\pm$ 0.016	0.634 $\pm$ 0.031	0.662 $\pm$ 0.028	0.639 $\pm$ 0.022	0.687$\pm$0.029
	AUC	0.683 $\pm$ 0.015	0.744 $\pm$ 0.019	0.765 $\pm$ 0.022	0.756 $\pm$ 0.027	0.778$\pm$0.022
Haberman	G-mean	0.747 $\pm$ 0.019	0.811 $\pm$ 0.023	0.849$\pm$0.029	0.848 $\pm$ 0.033	0.840 $\pm$ 0.035
	AUC	0.799 $\pm$ 0.008	0.816 $\pm$ 0.007	0.835 $\pm$ 0.017	0.839$\pm$0.019	0.820 $\pm$ 0.012
Ionosphere	G-mean	0.759 $\pm$ 0.028	0.796 $\pm$ 0.025	0.787 $\pm$ 0.030	0.789 $\pm$ 0.024	0.822$\pm$0.019
	AUC	0.891 $\pm$ 0.015	0.907 $\pm$ 0.021	0.917 $\pm$ 0.015	0.905 $\pm$ 0.018	0.929$\pm$0.011
Wdbc	G-mean	0.942 $\pm$ 0.012	0.971 $\pm$ 0.008	0.958 $\pm$ 0.011	0.964 $\pm$ 0.020	0.975$\pm$0.013
	AUC	0.978 $\pm$ 0.008	0.986 $\pm$ 0.011	0.991 $\pm$ 0.005	0.991 $\pm$ 0.004	0.997$\pm$0.002
Spambase	G-mean	0.861 $\pm$ 0.015	0.903 $\pm$ 0.022	0.887 $\pm$ 0.031	0.875 $\pm$ 0.028	0.929$\pm$0.020
	AUC	0.981 $\pm$ 0.007	0.982 $\pm$ 0.005	0.994$\pm$0.003	0.982 $\pm$ 0.009	0.994$\pm$0.005
Sonar	G-mean	0.753 $\pm$ 0.007	0.772 $\pm$ 0.021	0.782 $\pm$ 0.023	0.769 $\pm$ 0.018	0.784$\pm$0.019
	AUC	0.823 $\pm$ 0.011	0.869 $\pm$ 0.010	0.859 $\pm$ 0.008	0.844 $\pm$ 0.014	0.873$\pm$0.014

Table 5: Statistical analysis of Table 4 in terms of G-mean(AUC). $\star$ indicates statistically significant difference with a p -value of 0.05.

comparisons	wins	ties	losses	p -value
$\ell_{2,1}$ MKFSOS $\star^{(*)}$ vs. ℓ_1 MKFSOS	17(16)	1(4)	2(0)	0.008(0.012)
$\ell_{2,1}$ MKFSOS $\star^{(*)}$ vs. $\ell_{2,1}$ MKFS	19(18)	1(2)	0(0)	0.001(0.002)
$\ell_{2,1}$ MKFSOS $\star^{(*)}$ vs. MKOS-Single	16(14)	1(5)	3(1)	0.023(0.007)
$\ell_{2,1}$ MKFSOS $\star^{(*)}$ vs. MKOS-ALL	18(14)	0(3)	2(3)	0.001(0.009)
MKOS-Single vs. MKOS-ALL	13(8)	2(5)	5(7)	0.151(0.244)

norm MKL performs worse since the information carried in the kernels that get zero weights is completely discarded, which tends to remove relevant features if the only kernel function is not appropriate; while our $\ell_{2,1}$ norm MKL selected more than one base kernel for each feature, so as to select features that can capture the high skew in the class distribution. Although the limitation of ℓ_1 MKFSOS, we also point out that feature selection through ℓ_1 norm multi-kernel is effective in improving the G-mean and AUC values over the base SVM classifier.

2. Although feature selection is not conducted in MKOS-single and MKOS-ALL, MKOS-single can obtain a more appropriate combined kernel and provide a better performance than MKOS-ALL.

3. Compared with $\ell_{2,1}$ MKFS with only feature selection and MKOS-Single and MKOS-All with only oversampling, the results imply that the promising advantage of joint feature selection and oversampling for imbalanced data learning. Another important conclusion is that the effect of oversampling is more significant than the component of feature selection for imbalanced data learning.

4. The new instances generated by our proposed oversampling are useful for modeling the classifier even the amounts of two classes are equal.

4.1.4. Experiment III: The comparison with the other oversampling methods

The goal of this experiment carried out in this section is to show the validity of the feature selection strategy for multiple oversampling methods. In this experiment, we empirically investigate the effectiveness of the data generation mechanism in MKOS compared with two popular oversampling algorithms: SMOTE [48] and ADASYN (Adaptive Synthetic Sampling Approach for Imbalanced Learning) [18] on all features and features selected by $\ell_{2,1}$ MKFS, respectively. For the three comparable oversampling methods, all the experiments are performed using 10-fold cross validation where prediction is performed on held-out data that is not oversampled. Table 6 shows the average G-mean and AUC

Figure 5: Significant test between $\ell_{2,1}$ MKFSOS and other oversampling algorithms ($\alpha=0.05$)

values for each oversampling algorithms on all features and reduced features. We performed a statistical analysis of our results among the comparable oversampling with Friedman test and Nemenyi post-hoc test. Figure 5 shows the results of the Nemenyi test for G-mean and AUC.

As can be seen, the values of G-mean and AUC produced through $\ell_{2,1}$ MKFSOS are quite compelling compared with the other two oversampling algorithms conducted on regardless of all features or reduced features, producing the best results on 18 of 20 datasets in terms of G-mean, and 16 of 20 datasets in terms of AUC. Moreover, $\ell_{2,1}$ MKFSOS is significantly better than All-SMOTE, ALL-ADASYN and FS-SMOTE with respect to G-mean, and significantly better than all the other comparable methods except MKOS with respect to AUC. Our proposed $\ell_{2,1}$ MKFSOS and MKOS exhibit good results across all the datasets except Ecoli and Spambase, where they are outperformed by several methods. In summary, compared to the widely used approaches, $\ell_{2,1}$ MKFSOS shows appealing robustness on minority class without compromising the prediction accuracies of majority class, hence keeps the whole performances much better. The oversampling in feature space indeed helps alleviate the issue of inconsistency between data generating and data learning for SVM.

Under the condition where the feature selection is not carried out, we found that the MKOS outperforms the two other oversampling algorithms SMOTE and ADASYN on 18 of 20 (16 of 20), and 16 of 20 (14 of 20) in terms of G-mean (AUC), respectively; and beat them on 17 of 20 (15 of 20), and 16 of 20 (14 of 20) in terms of G-mean (AUC) when combined with feature selection of $\ell_{2,1}$ MKFS. For SMOTE and ADASYN, the ones working on the reduced features selected by $\ell_{2,1}$ MKFS improve the ones working on the all features on 14 and 15 of 20

H

Table 6: Experimental comparison of three compared oversampling algorithms on the all features and reduced features with respect to G-mean and AUC

Dataset	Metric	All features			Feature selection($\ell_{2,1}$ MKFS)		
		SMOTE	ADASYN	MKOS	SMOTE	ADASYN	MKOS
Abalone	G-mean	0.847 $\pm$ 0.016	0.833 $\pm$ 0.011	0.861 $\pm$ 0.023	0.854 $\pm$ 0.027	0.839 $\pm$ 0.016	0.871$\pm$0.019
	AUC	0.855 $\pm$ 0.018	0.882 $\pm$ 0.025	0.898 $\pm$ 0.024	0.861 $\pm$ 0.012	0.885 $\pm$ 0.017	0.915$\pm$0.033
Ecoli	G-mean	0.872 $\pm$ 0.011	0.872 $\pm$ 0.018	0.875 $\pm$ 0.026	0.866 $\pm$ 0.019	0.871 $\pm$ 0.010	0.879$\pm$0.025
	AUC	0.879 $\pm$ 0.015	0.881 $\pm$ 0.019	0.893 $\pm$ 0.022	0.865 $\pm$ 0.019	0.908$\pm$0.013	0.890 $\pm$ 0.022
Oil	G-mean	0.765 $\pm$ 0.039	0.784 $\pm$ 0.045	0.784 $\pm$ 0.033	0.801 $\pm$ 0.052	0.835$\pm$0.049	0.829 $\pm$ 0.066
	AUC	0.898 $\pm$ 0.028	0.901 $\pm$ 0.016	0.933 $\pm$ 0.029	0.907 $\pm$ 0.034	0.912 $\pm$ 0.028	0.951$\pm$0.033
Car	G-mean	0.927 $\pm$ 0.019	0.954 $\pm$ 0.015	0.979 $\pm$ 0.015	0.946 $\pm$ 0.018	0.949 $\pm$ 0.022	0.992$\pm$0.015
	AUC	0.997 $\pm$ 0.001	0.996 $\pm$ 0.001	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000
Letter	G-mean	0.943 $\pm$ 0.015	0.928 $\pm$ 0.010	0.968 $\pm$ 0.012	0.962 $\pm$ 0.007	0.955 $\pm$ 0.018	0.988$\pm$0.008
	AUC	0.959 $\pm$ 0.013	0.962 $\pm$ 0.021	0.977 $\pm$ 0.015	0.969 $\pm$ 0.022	0.981 $\pm$ 0.009	0.997$\pm$0.003
hypothyroid	G-mean	0.945 $\pm$ 0.024	0.933 $\pm$ 0.015	0.962 $\pm$ 0.021	0.968 $\pm$ 0.019	0.961 $\pm$ 0.015	0.978$\pm$0.013
	AUC	0.924 $\pm$ 0.011	0.913 $\pm$ 0.013	0.929 $\pm$ 0.022	0.931 $\pm$ 0.021	0.922 $\pm$ 0.015	0.962$\pm$0.017
Sick	G-mean	0.762 $\pm$ 0.039	0.750 $\pm$ 0.037	0.792$\pm$0.044	0.754 $\pm$ 0.033	0.731 $\pm$ 0.046	0.753 $\pm$ 0.055
	AUC	0.893 $\pm$ 0.016	0.917 $\pm$ 0.021	0.921$\pm$0.023	0.875 $\pm$ 0.018	0.859 $\pm$ 0.015	0.911 $\pm$ 0.020
Soybean	G-mean	0.952 $\pm$ 0.010	0.969 $\pm$ 0.005	0.981 $\pm$ 0.004	0.972 $\pm$ 0.004	0.998$\pm$0.004	0.998$\pm$0.018
	AUC	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000
Anneal	G-mean	0.918 $\pm$ 0.018	0.922 $\pm$ 0.013	0.935 $\pm$ 0.020	0.977 $\pm$ 0.008	0.977 $\pm$ 0.005	0.991$\pm$0.005
	AUC	0.983 $\pm$ 0.002	0.989 $\pm$ 0.005	0.982 $\pm$ 0.002	0.993 $\pm$ 0.005	1.000$\pm$0.000	1.000$\pm$0.000
Satimage	G-mean	0.754 $\pm$ 0.031	0.771 $\pm$ 0.028	0.851 $\pm$ 0.022	0.812 $\pm$ 0.039	0.857 $\pm$ 0.033	0.879$\pm$0.029
	AUC	0.871 $\pm$ 0.026	0.883 $\pm$ 0.020	0.923 $\pm$ 0.019	0.873 $\pm$ 0.018	0.894 $\pm$ 0.011	0.944$\pm$0.014
Zernike	G-mean	0.811 $\pm$ 0.041	0.808 $\pm$ 0.025	0.834 $\pm$ 0.031	0.807 $\pm$ 0.044	0.833 $\pm$ 0.050	0.865$\pm$0.059
	AUC	0.832 $\pm$ 0.044	0.851 $\pm$ 0.038	0.868 $\pm$ 0.033	0.835 $\pm$ 0.039	0.840 $\pm$ 0.028	0.915$\pm$0.032
Glass	G-mean	0.849 $\pm$ 0.019	0.833 $\pm$ 0.022	0.878 $\pm$ 0.028	0.922 $\pm$ 0.012	0.908 $\pm$ 0.009	0.919$\pm$0.014
	AUC	0.944 $\pm$ 0.014	0.949 $\pm$ 0.016	0.983 $\pm$ 0.011	0.979 $\pm$ 0.010	0.976 $\pm$ 0.007	0.998$\pm$0.001
Segment	G-mean	0.999 $\pm$ 0.001	1.000$\pm$0.000	1.000$\pm$0.000	0.999 $\pm$ 0.001	1.000$\pm$0.000	1.000$\pm$0.000
	AUC	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000	1.000$\pm$0.000
Hepatitis	G-mean	0.753 $\pm$ 0.031	0.753 $\pm$ 0.027	0.771 $\pm$ 0.014	0.788 $\pm$ 0.020	0.797 $\pm$ 0.016	0.828$\pm$0.021
	AUC	0.807 $\pm$ 0.021	0.801 $\pm$ 0.028	0.815 $\pm$ 0.024	0.849 $\pm$ 0.022	0.855 $\pm$ 0.031	0.887$\pm$0.013
Wpbc	G-mean	0.614 $\pm$ 0.022	0.651 $\pm$ 0.019	0.639 $\pm$ 0.022	0.653 $\pm$ 0.025	0.686 $\pm$ 0.016	0.687$\pm$0.029
	AUC	0.730 $\pm$ 0.009	0.721 $\pm$ 0.014	0.756 $\pm$ 0.027	0.749 $\pm$ 0.021	0.744 $\pm$ 0.018	0.778$\pm$0.022
Haberman	G-mean	0.845 $\pm$ 0.019	0.835 $\pm$ 0.023	0.848$\pm$0.033	0.833 $\pm$ 0.025	0.847 $\pm$ 0.021	0.840 $\pm$ 0.035
	AUC	0.827 $\pm$ 0.012	0.831 $\pm$ 0.013	0.839$\pm$0.019	0.822 $\pm$ 0.014	0.808 $\pm$ 0.017	0.820 $\pm$ 0.012
Ionosphere	G-mean	0.757 $\pm$ 0.029	0.712 $\pm$ 0.033	0.789 $\pm$ 0.024	0.796 $\pm$ 0.015	0.764 $\pm$ 0.010	0.822$\pm$0.019
	AUC	0.822 $\pm$ 0.020	0.798 $\pm$ 0.022	0.905 $\pm$ 0.018	0.883 $\pm$ 0.015	0.862 $\pm$ 0.011	0.929$\pm$0.011
Wdbc	G-mean	0.942 $\pm$ 0.009	0.946 $\pm$ 0.011	0.964 $\pm$ 0.020	0.967 $\pm$ 0.015	0.962 $\pm$ 0.008	0.975$\pm$0.013
	AUC	0.982 $\pm$ 0.005	0.989 $\pm$ 0.005	0.991 $\pm$ 0.004	0.991 $\pm$ 0.004	0.997$\pm$0.002	0.997$\pm$0.002
Spambase	G-mean	0.891 $\pm$ 0.024	0.889 $\pm$ 0.018	0.875 $\pm$ 0.028	0.959$\pm$0.022	0.935 $\pm$ 0.027	0.929 $\pm$ 0.020
	AUC	0.976 $\pm$ 0.011	0.982 $\pm$ 0.008	0.982 $\pm$ 0.009	0.996$\pm$0.003	0.977 $\pm$ 0.011	0.994 $\pm$ 0.005
Sonar	G-mean	0.772 $\pm$ 0.025	0.684 $\pm$ 0.033	0.769 $\pm$ 0.018	0.780 $\pm$ 0.026	0.732 $\pm$ 0.018	0.784$\pm$0.019
	AUC	0.851 $\pm$ 0.022	0.809 $\pm$ 0.019	0.844 $\pm$ 0.014	0.858 $\pm$ 0.017	0.846 $\pm$ 0.017	0.873$\pm$0.014

datasets.

An interesting phenomenon is that the original SMOTE does not succeed to be the best method in any of the datasets considered in the comparison (except two ties for first place). In many previous researches, SMOTE is outperformed by other more sophisticated oversampling[31, 33, 34, 32].

The analysis of these results leads to the following observations:

1. Due to this overgeneralization of minority region in original input space, SMOTE and ADASYN prone to misclassify many majority samples as minority, decreasing the classification performance of SVM, especially for some difficult classification problem induced by complex distribution such as overlapping of class exist, while the data distribution become more separable in the kernel space by mapping with an appropriate kernel matrix, ensuring that the instances generated are more accurate than the oversampling in the original space. This demonstrates that the assumption that feature space provides a more suitable space than the input space for performing oversampling. The data distribution leads to reduce the chance of class overlap in feature space by mapping with an appropriate kernel matrix, ensuring that the instances generated are more accurate than the oversampling in the original space since.

2. The results indicate that feature selection is as important as the oversampling in the imbalanced data classification. Oversampling on the high dimensional irrelevant feature leads to the creation of wrong instances when the class dispersion or the class noise exists. Our $\ell_{2,1}$ MKFS does feature selection in the embedding paradigm with balanced loss as objective function, hence improves the classification performance on the imbalanced datasets for three oversampling algorithms on the most datasets with the only exception of the Sick dataset, although the proposed feature selection of $\ell_{2,1}$ MKFS is conducted by kernel method while SMOTE and ADASYN work in the original input space. Removing the irrelevant features is useful to achieve an adequate classification performance for all three oversampling algorithms, especially with high dimensional datasets such as Spambase with 57 features and Oil with 49 features, where $\ell_{2,1}$ MKFS offers a great advantage over no feature selection for oversampling. This implies that high dimensionality of the imbalanced data sets is a significant issue to be solved, a lower dimensional representations after removing irrelevant and redundant features can improve the oversampling performance. However, feature selection is generally often ignored when learning from imbalanced data.

Finally, we examined the behavior of the oversampling methods by means of varying the oversampling ratio parameter, which plays a crucial role in the quality of the data generation. The optimal oversampling ratio is usually unknown. In

order to observe the influence of the re-sampling ratio in MKOS on the classification performance, we chose Oil and Sick datasets as examples to observe the performance with the ranging of the oversampling ratio. Figure 6 illustrates the AUC of MKOS, SMOTE and ADASYN with and without the feature selection under different over-sampling ratio parameters.

Figure 6: The performance of multiple comparative methods with and without feature selection while tuning oversampling ratio on Oil and Sick datasets in terms of AUC

From the Figure 6, we can found $\ell_{2,1}$ MKFSOS performs well compared to other common over-sampling methods with respect to robustness to over-sampling ratio. The results show that kernel oversampling is robust to choosing how many oversampled data points to add into the training set whereas the performance of other re-sampling methods can be highly dependent on the degree of re-sampling.

4.1.5. Experiment IV: The comparison with the state-of-the-art approaches for imbalanced data learning

This experiment presents a comprehensive empirical investigation comparing the performances of several techniques for improving classification performance when data are imbalanced. There are four main approaches to accomplish an imbalanced data classification task: oversampling, undersampling, cost sensitive learning and feature selection methods. Specially, we compare the performance of seven methods with our proposed method, including as follows:

- SMOTEBoost (SMB) [19] is an ensemble based **oversampling** using a combination of SMOTE and a standard boosting procedure. SMOTEBoost tries to better model the minority class by providing the learner not only

with the minority class examples that were misclassified in the previous boosting iteration but also with a broader representation of those instances achieved by SMOTE. The ensemble size is set to be 50.

- Edited Nearest Neighbor (ENN)[9] is an **undersampling** method that removes data examples whose class label differs from that of at least two of its three nearest neighbors.
- Cluster-based Over-Sampling (CBOS) [24] is an **oversampling** integrating clustering with the SMOTE. In CBSO, we use K-means rather than average-linkage agglomerative clustering, a hierarchical clustering algorithm to decompose the minority class space, since average-linkage agglomerative clustering causes longer training time. By using K-mean instead of agglomerative clustering, it is become simpler to implement and that results in faster model training times.
- MetaCost is **cost sensitive learning** based on wrapping a metalearning stage around the error-based classifier so as to improve the accuracy of minority class[11]. The cost-sensitive learning technique takes misclassification costs into account during the model construction, and does not modify the imbalanced data distribution directly. The ratio misclassification cost of the Metacost is set by the reverse of the size of each class.
- MOCSSVM [35] is a **wrapper feature selection** paradigm oriented by the evaluation measure of imbalanced dataset as objective function with respect to misclassification cost, feature subset and intrinsic parameters of SVM simultaneously for improving the performance of cost-sensitive SVM. We use AUC as the evaluation measure to guide the optimization of the ratio misclassification cost, intrinsic parameters and feature set.
- Recursive Feature Elimination (RFE) [36] is **wrapper feature selection** by employing a backward algorithm for feature elimination with balanced loss function. The goal of this approach is to find a subset of features, eliminating those features whose removal has less impact on the final solution. In addition, artificial target class instances are generated using the oversampling method SMOTE on the training set prior to feature selection.
- RELIEFF (REF) is a **filter feature selection**. RELIEF is a feature selection metric based on the nearest neighbor rule. It evaluates a feature based on how well its values differentiate themselves from nearby points in order

to filter out poorly informative. SMOTE is employed since the ability to estimate the quality of attributes by RELIEFF is highly dependent on the number of instances from different classes in the dataset.

For Recursive Feature Elimination and RELIEFF feature selection methods, SMOTE is integrated for fair comparisons with our hybrid method (named RFE-SM and REF-SM). Besides, one of the important issues for the feature selection of RFE and RELIEFF is to set the number of required features. In our experiments, in order to compare equally, we retain the the top ranked features by each selector with the same size as our proposed $\ell_{2,1}$ MKFS. Table 7 presents the experimental results of the multiple comparable algorithms over the data sets described above with respect to G-mean, AUC and corresponding average rank. For space limitations, for each classification algorithm, we only provide the average values of G-mean and AUC instead of the average $\pm$ std for each imbalanced data set.

Figure 7: Average rankings of competing methods

Figure 8: Significant test between $\ell_{2,1}$ MKFSOS and the state-of-the-art algorithms ($\alpha=0.05$)

Table 7: Average values of G-mean and AUC for the state-of-the-art imbalanced data methods

Dataset	Metric	RFE-SM	REF-SM	ENN	CBOS	SMB	Metacost	MOCSSVM	$\ell_{2,1}$ MKFSOS
Abalone	G-mean	0.859	0.712	0.687	0.730	0.744	0.691	0.855	0.871
	AUC	0.903	0.686	0.613	0.727	0.780	0.699	0.893	0.915
Ecoli	G-mean	0.851	0.833	0.820	0.855	0.858	0.833	0.871	0.879
	AUC	0.860	0.848	0.815	0.872	0.861	0.844	0.896	0.890
Oil	G-mean	0.801	0.772	0.728	0.755	0.744	0.769	0.841	0.829
	AUC	0.948	0.882	0.711	0.724	0.921	0.897	0.951	0.951
Car	G-mean	0.987	0.968	0.945	0.977	0.968	0.940	0.998	0.992
	AUC	1.000	1.000	0.993	0.997	1.000	1.000	1.000	1.000
Letter	G-mean	0.985	0.961	0.933	0.929	0.922	0.935	0.985	0.988
	AUC	0.990	0.914	0.896	0.957	0.988	0.895	0.995	0.997
hypothyroid	G-mean	0.973	0.973	0.659	0.922	0.931	0.753	0.968	0.978
	AUC	0.944	0.927	0.919	0.931	0.905	0.819	0.949	0.962
Sick	G-mean	0.792	0.714	0.544	0.742	0.757	0.669	0.750	0.753
	AUC	0.917	0.911	0.729	0.909	0.883	0.854	0.898	0.911
Soybean	G-mean	0.976	0.967	0.953	0.971	0.997	0.945	1.000	0.998
	AUC	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Anneal	G-mean	0.881	0.852	0.715	0.772	0.784	0.676	0.984	0.991
	AUC	0.973	0.955	0.879	0.933	0.889	0.857	0.995	1.000
Satimage	G-mean	0.854	0.829	0.736	0.770	0.757	0.762	0.833	0.879
	AUC	0.921	0.876	0.848	0.892	0.873	0.884	0.967	0.944
Zernike	G-mean	0.814	0.795	0.601	0.831	0.827	0.648	0.822	0.865
	AUC	0.848	0.826	0.805	0.831	0.848	0.820	0.902	0.915
Glass	G-mean	0.845	0.839	0.813	0.920	0.898	0.862	0.941	0.919
	AUC	0.978	0.961	0.966	0.995	0.966	0.949	0.999	0.998
Segment	G-mean	1.000	1.000	0.993	1.000	0.998	1.000	1.000	1.000
	AUC	1.000	1.000	0.997	1.000	1.000	1.000	1.000	1.000
Hepatitis	G-mean	0.646	0.628	0.597	0.659	0.667	0.616	0.801	0.828
	AUC	0.828	0.801	0.744	0.777	0.759	0.737	0.805	0.887
Wpbc	G-mean	0.655	0.637	0.571	0.606	0.629	0.643	0.669	0.687
	AUC	0.860	0.825	0.632	0.766	0.772	0.744	0.853	0.874
Haberman	G-mean	0.849	0.841	0.822	0.862	0.853	0.839	0.832	0.840
	AUC	0.829	0.831	0.787	0.849	0.849	0.822	0.808	0.820
Ionosphere	G-mean	0.807	0.799	0.688	0.749	0.776	0.735	0.782	0.822
	AUC	0.908	0.861	0.791	0.877	0.869	0.795	0.877	0.929
Wdbc	G-mean	0.962	0.946	0.952	0.909	0.931	0.922	0.949	0.975
	AUC	0.992	0.969	0.923	0.977	0.981	0.951	0.974	0.997
Spambase	G-mean	0.885	0.869	0.867	0.921	0.905	0.859	0.919	0.926
	AUC	0.971	0.973	0.967	0.963	0.988	0.968	0.976	0.994
Sonar	G-mean	0.793	0.744	0.674	0.765	0.724	0.728	0.760	0.784
	AUC	0.891	0.848	0.731	0.820	0.866	0.841	0.880	0.873
Average	G-mean	0.86	0.834	0.765	0.832	0.834	0.791	0.878	0.890
	AUC	0.928	0.895	0.837	0.890	0.899	0.869	0.931	0.943
Average Rank	G-mean	3.175	5	7.4	4.575	4.875	6.4	2.9	1.725
	AUC	2.9	5	7.4	4.775	4.475	6.35	3.05	2.05

A first glance at the results shows that our methods generally outperform all other compared methods in terms of G-mean and AUC in 16 and 18 of the 20 datasets, respectively. $\ell_{2,1}$ MKFSOS provides a relative improvement in terms of G-mean(AUC) of 3.80%(1.63%), 7.39%(5.87%), 18.61%(14.28%), 7.66%(6.66%) and 7.40%(5.14%) as compared to re-sampling methods (RFE-SM, REF-SM, ENN, CBOS and SMB), and 13.76%(9.19%) and 1.51%(1.39%) as compared to cost sensitive learning methods (Metacost, and MOCSSVM), respectively. $\ell_{2,1}$ MKFSOS (0.890/0.943) is followed by MOCSSVM (0.878/0.931) and RFE-SM (0.86/0.928) in terms of average G-mean/AUC. ENN is the worst method (0.765/0.837). Except $\ell_{2,1}$ MKFSOS, the best oversampling method is SMB (0.834/0.899).

Fig.7 shows the average rankings of the competing algorithms, and Fig. 8 visualizes the Nemenyi post-hoc test at the significance level $\alpha = 0.05$. It can be found from Figure 8 that $\ell_{2,1}$ MKFSOS statistically outperforms all methods except RFE-SM and MOCSSVM in terms of G-mean and AUC at the significance level $\alpha = 0.05$, which demonstrate the good behavior of the proposed approach compared with other related methods found in the literature.

The results in these tables are summarized hereafter:

1. Among the re-sampling algorithms, ENN is the worst method since that it is hard to identify the noise when the distribution is complex and imbalanced. Some border points may be also removed as noise while they are useful for training classifiers, resulting in suffering from the loss of information. The sophisticated oversampling based on SMOTE, CBOS (cluster-based oversampling) and SMB (ensemble-based SMOTE) generally performed the undersampling.
2. Among the cost sensitive learning, MetaCost also perform poor, and worse than the other advanced over-sampling methods. It may be because the ratio misclassification cost set based on the amount ratio between two classes is not appropriate, resulting in obtaining a unexpected result. The misclassification cost vital for cost sensitive learning, and need to be searched by some heuristic methods [21, 35].
3. Among the hybrid feature selection methods, $\ell_{2,1}$ MKFSOS, RFE-SM and MOCSSVM integrating feature selection techniques except REF-SM perform better than ENN, CBOS, SMB and Metacost for almost all datasets/metrics. The results demonstrate that feature selection designed for imbalanced data helps us to retain only those features that are most important for the classification task. RELIEFF fail to account for imbalanced class distributions and are not compatible to data oversampling, resulting in not identify the relevant features for imbalanced data and poor predictions for minority class samples. It also prove that the RELIEFF is not a good metric for feature selection on imbalanced data.

4. When comparing to RFE-SMOTE and MOCSSVM, $\ell_{2,1}$ MKFSOS produces better generalization performance, although without statistical difference. MOCSSVM can optimize the features and cost parameter simultaneously, but for some small data (Wdbc and Ionosphere), is worse since MOCSSVM is only use the original minority class without new knowledge to be added, causing overfitting when the number of minority class examples is limited. Moreover, MOCSSVM and RFE-SMOTE are based on single kernel selected by the cross validation method, while our $\ell_{2,1}$ MKFSOS which is based on kernel combinational. Most studies show that multiple kernel strategy outperforms the best performing kernel [50, 51]. Furthermore, in RFE-SMOTE, the feature selection is consistent with SVM classifier by ranking features according to their predictive power, while SMOTE operates in the original input space, presenting distortions with SVM classifier.

5. In the comparison among four hybrid feature selection algorithms combined oversampling (REF-SM, RFE-SM and $\ell_{2,1}$ MKFSOS) or cost sensitive learning (MOCSSVM), we can find wrapper and embedded methods take the relationship between features, and therefore they should be more effective in identifying and eliminating redundancy than filter method. In order to explore the classification performance with varying the number of selected features, we plot their performance using different numbers of variables in Fig. 9. To make the comparisons more complete, we compare four feature selection methods with as well as without oversampling except MOCSSVM by exploring the whole feature set. For space considerations, we only present the results on the Spambase and Zernike with a higher dimensionality (57 and 47, respectively) in Figure 9(a) and 9(b).

6. Regarding the computational complexity, even though our analysis is mainly devoted to algorithm performance, we should highlight that MOCSSVM is more time-consuming than RFE-SM and $\ell_{2,1}$ MKFSOS, since both of MOCSSVM and RFE-SM run iteratively for estimating the intrinsic parameters by PSO. As an example of the computation time comparison, we also presented the results on the Spambase and Zernike. Although $\ell_{2,1}$ MKFSOS requires many more calls to the SVM solver (#SVM eval), as we stated in Section 3.7, a warm-start call to the SVM solver results in much less computation than a call from scratch.

Table 8: Average performance measures for three hybrid feature selection methods

Methods	Zernike			Spambase		
	AUC	Time (s)	# SVM eval	AUC	Time (s)	#SVM eval
RFE-SM	0.848	388	46	0.971	297	40
MOCSSVM	0.902	2056	78	0.976	1644	67
$\ell_{2,1}$ MKFSOS	0.915	1278	129	0.994	1089	95

Figure 9: Classification performance of multiple feature selection strategies on Spambase and Zernike datasets using different numbers of feature with respect to AUC

It can be seen that our method still outperforms other feature selection algorithms regardless of oversampling or not. Comparing with Recursive Feature Elimination (RFE) and RELIEFF, MOCSSVM and $\ell_{2,1}$ MKFS can automatically determine the number of features. Moreover, wrapper based feature selection is computationally demanding compared with filter and embedded methods.

4.2. Evaluating the effectiveness of heterogeneous feature fusion and selection for computer aided lung nodule detection

In computer aided detection of Lung nodule, the variance in the volumes, shapes, and appearances of the suspicious nodule objects is enormous. In order to achieve satisfactory results in nodule candidate classification, multiple heterogeneous feature subsets are extracted to describe the visual objects. Each representation of feature subset describes different aspects of the visual characteristics of nodule candidates, a joint analysis of these feature subsets can potentially exploit their complementary information and improve the prediction. A common practice is the concatenation of the whole feature subsets into a longer feature vector.

However, this may be not enough for effective combination of features from different views. Moreover, current CAD schemes for nodule characterization have achieved high performance levels and would be able to improve radiologists performance in the characterization of nodules in thin-section CT, whereas current schemes for nodule detection appear to report many false positives [53]. Since the radiologists must examine each identified object, it is highly desirable to eliminate these false positives (FPs) as much as possible while retaining the true positives (TPs). The purpose of false-positive reduction is to remove these false positives (FPs) as much as possible while retaining a relatively high sensitivity. It is a binary classification between the nodule and non-nodule, and the two classes are commonly skewed and have extremely unequal misclassification costs, which is a typical class imbalance problem [8, 1]. To address the above challenges, we apply our $\ell_{2,1}$ MKFSOS to classification of suspicious nodule object by integrating heterogeneous feature subsets and oversampling the true positive (minority) class. the two classes are skewed and have extremely unequal misclassification costs, which is a typical class imbalance problem [8, 1]

4.2.1. Dataset

We adopted the scans collected by the lung image database consortium (LIDC) [30] to evaluate our proposed method. LIDC is a publicly available database from the National Biomedical Imaging Archive (NBIA), and its nodules have been fully annotated by multiple radiologists. In this database, four expert chest radiologists drew outlines for nodules having effective sizes of 3 mm or greater. The ground truth was then established by a blind reading and a subsequent unblinded reading. The LIDC Database contains 1012 cases, each of which includes images from a clinical thoracic CT scan and an associated XML file that records the results of a two-phase image annotation process performed by four experienced thoracic radiologists. Due to the limitation that our detection methods only be effective on the solid nodule, we successfully detected 209 solid nodules along with 778 non-nodules with our detection and labeling methods using the provided ground truth information from 372 scans containing solid nodules. Then the same features were computed from the VOIs to characterize the nodule candidate data. After obtaining the candidate nodule instances, we calculate the features for each of the nodule candidates from intensity, texture, shape and gradient distribution aspects. The feature lists is shown in Table 9.

Table 9: Feature set for candidate nodule classification

ID	Category	Feature Set(<i>FeaSet</i>)	Description
F1-7	Intensity	Intensity statistical feature	The gray value within the objects was characterized by use of seven statistics (mean, variance, max, min, skew, kurt, entropy)
F8-12		Sub-volume distribution feature	The average intensity within each sub-volume along the radial directions
F13-60	Texture	Gabor feature	The mean value of each filtered image
F61-67	Shape	SI statistical feature	The volumetric shape index (SI) representing the local shape feature at each voxel was characterized by use of seven statistics.
F68-74		CV statistical feature	The volumetric curvedness (CV), which quantifies how highly curved a surface is, was characterized by use of seven statistics.
F75-77		Geometric feature	volume, surface area and compactness
F78-84	Gradient	GC statistical feature	The concentration characterizing the degree of convergence of the gradient vectors at each voxel, was characterized by use of seven statistics
F85-91		GS statistical feature	The gradient strength of the gradient vectors at each voxel, was characterized by use of seven statistics

4.2.2. Results

The mixed $\ell_{2,1}$ norm formulation enforces group sparsity among different feature subsets, which actually performs as a role of feature subset selection, while at the same time exploiting complementary information among the different kernels. In order to visualize the contribution of each feature type in such a MKL-based fusion scheme, we plot the kernel weights of the base kernels of our multiple kernel learning in Fig.10(a), Fig.10(c), Fig.10(b) and Fig.10(d), respectively. The weight values correspond to the contribution for each representation (feature subset). Kernels corresponding to the discriminative feature subset are assigned the highest weights. We found that the weights of the Gabor and Gradient strength feature are very low (norm proportion is less than the given threshold ratio 10%). Henceforth, we retain the feature subsets with high weight as the most discriminative feature representation and remove the Gabor and Gradient strength feature subsets with least importance, so as to obtain the heterogeneous feature subset fusion

and selection. Moreover, from Fig. 10, we see that our $\ell_{2,1}$ norm MKL selected more than one base kernel for each feature set, while the weighted MKL tends to select sparse base kernels for feature fusion, resulting in inaccurate feature subset selection.

Figure 10: Base kernel weights comparison of different MKL algorithms on the nodule dataset

We compare the four multi-kernel based fusion schemes comprehensively in Table 10. All the multi-kernel learning methods are combined with MKOS. The ℓ_1 norm MKL performs worst since the information carried in the kernels that get zero weights is completely discarded, which may be explained by the fact that the combined kernels carry complementary information. Moreover, the ℓ_1 norm MKL and ℓ_2 norm MKL do not take the structure of feature subset into consideration, resulting in the prediction performance being dominated by our $\ell_{2,1}$ norm MKL. The experimental results in 10 demonstrated that the $\ell_{2,1}$ norm framework achieves the best. We obtained a more generalized performance for heterogeneous potential nodule data.

Table 10: The comparison of performance using multi feature representation classification methods

Method	Sensitivity (%)	Specificity (%)	G-mean (%)	AUC (%)
ℓ_1 MKL	77.55 ± 3.26	86.11 ± 2.87	81.72 ± 3.88	81.13 ± 3.41
ℓ_2 MKL	82.47 ± 6.48	91.07 ± 5.57	86.83 ± 5.95	84.33 ± 4.74
weighted-MKL	75.36 ± 3.11	88.21 ± 5.57	81.53 ± 3.16	82.40 ± 2.78
$\ell_{2,1}$ MKL	85.23 ± 1.67	91.33 ± 2.08	88.23 ± 1.77	87.23 ± 2.83

Moreover, we investigate the three different types of data fusion methods, namely, early (feature-level fusion), late (decision-level fusion), and intermediate (kernel-level fusion) fusion for discriminating potential nodule data in our experiment. We compare our proposed method (kernel-level fusion) with SVM-All (feature-level fusion, concatenating all the feature subsets from the different sources into a single feature vector), and two state-of-the-art ensemble methods (decision-level fusion) for false positive reduction of Lung CAD: Asymmetric AdaBoost (AsyAdaboost) [54] and (HPS-DRS) [55], with cost sensitive learning strategy and hybrid sampling technique for imbalanced classes (TP and FP) learning, respectively. Ensemble classifiers construct multiple base classifiers, then combine the predicted results by different classifiers via voting to reach the final output. AsyAdaboost and HPS-DRS are based Adaboost and random subspace ensemble framework, and incorporated with cost sensitive learning and hybrid sampling, respectively. In addition, we also report the classification performances by SVM on each individual type of features (SVM-Inten, SVM-Text, SVM-Grad and SVM-Shape) as baselines. In SVM-All, the training data are normalized to have zero mean and unit variance, and the test data are then normalized using the mean and variance of the training data. For the two ensemble methods, the amount of base classifiers is fixed to 50.

From the results in Table 11, we can conclude that regardless of the fusion level (feature-level, kernel-level or decision-level), the methods combining features from multiple views consistently outperform the best single feature based classifier. It confirms different source of feature set provide complementary information, which may be useful for classification of nodule candidates when used together. Besides, the experiment results (Table 11) demonstrate the benefit of $\ell_{2,1}$ MKFSOS compared to other fusion methods and the commonly used methods for false positive reduction in Lung nodule CAD. The disadvantage of SVM-All is that it treats multiple features equally, being incapable of effectively exploring the complementary information; while for the decision-level fusion of ensemble classifiers, only the final decision classification outputs are used, resulting in in-

Table 11: The performance of different comparative methods

Methods	Sen	Spec	G-mean	AUC
SVM-Inten	0.576	0.872	0.709	0.743
SVM-Text	0.522	0.844	0.664	0.718
SVM-Grad	0.562	0.866	0.698	0.727
SVM-Shape	0.597	0.886	0.727	0.749
SVM-All	0.648	0.892	0.760	0.774
$\ell_{2,1}$ MKFS	0.772	0.905	0.836	0.827
$\ell_{2,1}$ MKFSOS	0.853	0.897	0.875	0.882
AsyAdaboost	0.734	0.855	0.792	0.833
HPS-DRS	0.832	0.862	0.847	0.857

formation loss.

We also showed the ROC analysis of the various approaches in Figure 11. $\ell_{2,1}$ MKFSOS shows appealing performances on imbalanced data through feature selection and oversampling by means of multi kernel method.

Figure 11: ROC analysis of the various approaches for suspicious nodules classification

4.3. *Evaluating the effectiveness of feature group selection for classification of Alzheimer’s disease and mild cognitive impairment*

4.3.1. *Dataset*

In this subsection, we demonstrate another application of $\ell_{2,1}$ MKFSOS for diagnosis of Alzheimer’s disease and mild cognitive impairment. The data used in this paper were obtained from the ADNI database (adni.loni.usc.edu) [58]. The primary goal of ADNI has been to test whether serial MRI, PET, other biological markers, and clinical and neuropsychological assessments can be combined to measure the progression of MCI and early AD. Approaches to characterize AD progression will help researchers and clinicians develop new treatments and monitor their effectiveness. Further, being able to understand disease progression will increase the safety and efficacy of drug development and potentially decrease the time and cost of clinical trials. Neuroimaging has been shown to be a powerful tool for the exploration of disease progression and therapeutic efficacy in AD and MCI. While FDG-PET scans may show a high sensitivity or specificity for the early detection of AD, the validation of structural MRI markers is the core project in ADNI due to its greater availability, faster data acquisition, and lower cost. Therefore, our current work focuses on MRI data. The MRI features used in our experiments are based on the imaging data from the ADNI database processed by a team from UCSF (University of California at San Francisco), who performed cortical reconstruction and volumetric segmentations with the FreeSurfer image analysis suite (<http://surfer.nmr.mgh.harvard.edu/>). There were $p = 327$ MRI features (covariates) in total, including the cortical thickness average (TA), standard deviation of thickness (TS), surface area (SA) cortical volume (CV) and subcortical volume (SV) for a variety of regions-of-interest (ROIs). In the context of AD, each group corresponds to a region-of-interest (ROI) in the brain, and $\ell_{2,1}$ norm for kernel weights is employed to group relevant MRI features in the same ROI.

The dataset contains 816 subjects, who are categorized into 3 baseline diagnostic groups: Cognitively Normal (228), Mild Cognitive Impairment (399), and Alzheimers Disease (189). To evaluate the performances of our proposed method in diagnosis of AD (Alzheimers Disease) and MCI (Mild Cognitive Impairment), three binary classification experiments were performed: Normal Control (NC) vs MCI (Mild Cognitive Impairment), AD (Alzheimers Disease) vs MCI, and AD vs NC. The imbalanced data issue exist between class of AD (minority) and MCI, as well as class of MCI and NC (minority) . The oversampling ratio parameter is set to 200% for minority class.

4.3.2. Results

In our study, we compare different methods in distinguishing AD/NC, AD/MCI and MCI/NC, respectively, all of which are two-class classification problems. For the classification between AD and NC, oversampling is not conducted due to the nearly balanced class distribution. In the classification of AD and MCI, linear methods are popular [59, 62]. In this experiment, we also conduct the comparison between our nonlinear kernel based method and linear based feature selection method (logistic regression with group lasso [61], named LRGL), in order to evaluate the effectiveness of our proposed nonlinear method. Specifically, we use logistic regression with group lasso [56] to select the important ROIs, and then operate oversampling on the selected ROIs. At last, a SVM classifier is trained on the preprocessed training set for a fair comparison. Table 12-14 shows the performance of each algorithm for three binary classification with respect to sensitivity, specificity, G-mean and AUC.

Table 12: Experimental results of the different comparable classifiers for AD vs. NC

Methods	Sen	Spec	G-mean	AUC
SVM	0.787	0.880	0.832	0.850
LRGL	0.809	0.918	0.862	0.887
$\ell_{2,1}$ MKFS	0.849	0.922	0.885	0.907

Table 13: Experimental results of the different comparable classifiers for AD vs. MCI

Methods	Sen	Spec	G-mean	AUC
SVM	0.287	0.930	0.517	0.609
LRGL-SMOTE	0.319	0.938	0.547	0.681
ℓ_1 MKFSOS	0.618	0.856	0.727	0.729
$\ell_{2,1}$ MKFS	0.498	0.959	0.691	0.738
$\ell_{2,1}$ MKFSOS	0.664	0.835	0.745	0.772

Table 14: Experimental results of the different comparable classifiers for MCI vs. NC

Methods	Sen	Spec	G-mean	AUC
SVM	0.342	0.865	0.545	0.676
LRGL-SMOTE	0.746	0.573	0.654	0.729
ℓ_1 MKFSOS	0.818	0.782	0.800	0.776
$\ell_{2,1}$ MKFS	0.677	0.799	0.735	0.743
$\ell_{2,1}$ MKFSOS	0.794	0.822	0.808	0.801

Experiments indicate that our algorithm is able to attain higher sensitivity than other methods except ℓ_1 MKFSOS when classifying MCI and NC, while maintaining higher G-mean. Generally, feature selection methods are preferable in clinical applications due to their ease for interpretation. It is important to identify the critical brain regions that are most related to AD and MCI.

In our $\ell_{2,1}$ MKFSOS method, the weights of kernels for each ROI is used to select the potential predictors from MRI. The predictive ROIs are identified by 10-fold cross validation. In each fold, the ROIs are ranked based on the value of $\|d_g\|_2$, and the top-10 are identified as important. Figure 12 lists the top 10 feature identified by our $\ell_{2,1}$ MKFSOS method when discriminating AD and NC. The observation proves that the ROIs which appear in the top-10 in majority of the folds are deemed most predictive and important. The important regions identified in 12 such as Hippocampus, Entorhinal, Fusiform, ParsOpercularis are relevant according to existing AD domain knowledge and in accordance with prior studies [62, 60].

5. Conclusion

Feature selection and classification of imbalanced data sets are two of the most interesting machine learning challenges. We proposed a multi kernel framework to address the both issues of feature selection and imbalanced data learning simultaneously. The proposed method has been validated on some benchmark imbalanced data and two real applications. The experimental results presented in this study have demonstrated that the proposed framework provides a very competitive solution to other existing state-of-the-arts methods by integrating feature selection and oversampling in a unified kernel framework. In the future research, we will extend the framework to the imbalanced multi-class [3] and multi-label data [63] classification.

Figure 12: The top 10 ROI identified by $\ell_{2,1}$ MKFSOS in discriminating AD and NC

6. Acknowledgments

This research was supported by the National Science Foundation for Distinguished Young Scholars of China under Grant (No. 71325002, No. 61225012), the National Key Technology Research and Development Program of the Ministry of Science and Technology of China under grant (No.2014BAI17B01), the National High Technology Research and Development Program of China (No.2015AA020106), the National Natural Science Foundation of China (No.61502091), the Fundamental Research Funds for the Central Universities (No.N140403004), and the Postdoctoral Science Foundation of China (No.2015M570254).

- [1] He, Haibo and Garcia, Eduardo A, Learning from imbalanced data, *IEEE Transactions on Knowledge and Data Engineering*, **21**(9):1263–1284, 2009.
- [2] Jian, Chuanxia and Gao, Jian and Ao, Yinhui, A new sampling method for classifying imbalanced data based on support vector machine ensemble, *Neurocomputing*, **193**:115–122, 2016.
- [3] Fernández-Navarro, Francisco and Hervás-Martínez, César and Gutiérrez, Pedro Antonio, A dynamic over-sampling procedure based on sensitivity for multi-class problems, *Pattern Recognition*, **44**(8):1821–1833, 2011.
- [4] Li, Kuan and Kong, Xiangfei and Lu, Zhi and Wenyin, Liu and Yin, Jianping, Boosting weighted ELM for imbalanced learning, *Neurocomputing*, **128**:15–21, 2014.
- [5] Gao, Ming and Hong, Xia and Chen, Sheng and Harris, Chris J and Khalaf, Emad, PDFOS: PDF estimation based over-sampling for imbalanced two-class problems, *Neurocomputing*, **138**:248–259, 2014.
- [6] Guyon, Isabelle and Gunn, Steve and Nikravesh, Masoud and Zadeh, Lofti A, Feature extraction: foundations and applications, *Springer, Berlin*, 2008.
- [7] Yang, Qiang and Wu, Xindong, 10 challenging problems in data mining research, *International Journal of Information Technology & Decision Making*, **5**(4):597–604, 2006.
- [8] Chawla, Nitesh V and Japkowicz, Nathalie and Kotcz, Aleksander, Editorial: special issue on learning from imbalanced data sets, *ACM Sigkdd Explorations Newsletter*, **6**(1):1–6, 2004.
- [9] D. Wilson, Asymptotic properties of nearest neighbor rules using edited data, *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-2**(3) (1972), 408–421.
- [10] Liu, Jun and Ji, Shuiwang and Ye, Jieping, Multi-task feature learning via efficient l2, 1-norm minimization, *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, 339–348, 2009.
- [11] P. Domingos, Metacost: a general method for making classifiers cost-sensitive, In *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 99)*, ACM, New York, NY, USA, 1999, pp. 155–164.

- [12] Ye, Jieping and Liu, Jun, Sparse methods for biomedical data, *ACM Sigkdd Explorations Newsletter*, **14**(1):4–15, 2012.
- [13] Ji, Shuiwang and Sun, Liang and Jin, Rong and Ye, Jieping, Multi-label multiple kernel learning, *Advances in neural information processing systems*:777–784, 2009.
- [14] Rakotomamonjy, Alain and Bach, Francis R and Canu, Stéphane and Grandvalet, Yves, SimpleMKL, *Journal of Machine Learning Research*, **9**:2491–2521, 2008.
- [15] Chen, Jianhui and Ji, Shuiwang and Ceran, Betul and Li, Qi and Wu, Mingrui and Ye, Jieping, Learning subspace kernels for classification, *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*:106–114, 2008.
- [16] Hwang, Sung Ju and Sha, Fei and Grauman, Kristen, Sharing features between objects and their attributes, *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*:1761–1768, 2011.
- [17] Chen, Xue-wen and Wasikowski, Michael, Sharing features between objects and their attributes, *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*:124–132, 2008.
- [18] He, Haibo and Bai, Yang and Garcia, Edwardo A and Li, Shutao, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, *IEEE International Joint Conference on Neural Networks*:1322–1328, 2008.
- [19] N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer, SMOTEBoost: Improving prediction of the minority class in Boosting, In *Proceedings of 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2003)*, Springer Berlin Heidelberg, Cavtat-Dubrovnik, Croatia, 2003, pp. 107–119.
- [20] T. Hoens and N. Chawla, Generating diverse ensembles to counter the problem of class imbalance, In *Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2010)*, Hyderabad, India, 2010, pp. 488–499.

- [21] Sun, Yanmin and Kamel, Mohamed S and Wang, Yang, Boosting for learning multiple classes with imbalanced class distribution, In *Proceedings of Sixth International Conference on Data Mining*, pp. 592–602, 2006.
- [22] Jawanpuria, Pratik and Varma, Manik and Nath, Saketha, On p-norm path following in multiple kernel learning for nonlinear feature selection, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*:118–126, 2014.
- [23] Beck A, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM journal on imaging sciences*:183–202, 2009.
- [24] Barua, Sukarna and Islam, Md Monirul and Murase, Kazuyuki, A novel synthetic minority oversampling technique for imbalanced data set learning, *Neural Information Processing*:735–744, 2011.
- [25] Kubat, Miroslav and Matwin, Stan and others, Addressing the curse of imbalanced training sets: one-sided selection, *ICML*:179–186, 1997.
- [26] Das, Biswajit and Krishnan, Narayanan C and Cook, Diane J, wRACOG: A gibbs sampling-based oversampling technique, *13th International Conference on Data Mining (ICDM)*:111–120, 2013.
- [27] Exploratory undersampling for class-imbalance learning, 10 challenging problems in data mining research, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **39**(2):539–550, 2009.
- [28] Akbani, Rehan and Kwek, Stephen and Japkowicz, Nathalie, Applying support vector machines to imbalanced datasets, *Machine learning: ECML 2004*:39–50, 2004.
- [29] Zhuang, Jinfeng and Tsang, Ivor W and Hoi, Steven CH, Two-Layer Multiple Kernel Learning, *AISTATS*:909–917, 2011.
- [30] McNitt-Gray, Michael F and Armato, Samuel G and Meyer, Charles R and Reeves, Anthony P and McLennan, Geoffrey and Pais, Richie C and Freymann, John and Brown, Matthew S and Engelmann, Roger M and Bland, Peyton H and others, The Lung Image Database Consortium (LIDC) Data Collection Process for Nodule Detection and Annotation, *Academic radiology*, **14**:1464–74, 2007.

- [31] Das, Barnan and Krishnan, Narayanan C and Cook, Diane J, RACOG and wRACOG: Two Probabilistic Oversampling Techniques, *IEEE transactions on knowledge and data engineering*, **27**(1):222–234, 2015.
- [32] Yu, Hualong and Ni, Jun and Zhao, Jing, ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data, *Neurocomputing*, **101**:309–318, 2013.
- [33] Cao, Peng and Zhao, Dazhe and Zaiane, Osmar, Hybrid probabilistic sampling with random subspace for imbalanced data learning, *Intelligent Data Analysis*, **18**(6):1089–1108, 2014.
- [34] Barua, Sukarna and Islam, Md Monirul and Yao, Xin and Murase, Kazuyuki, MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning, *IEEE Transactions on Knowledge and Data Engineering*, **26**(2):405–425, 2014.
- [35] Cao, Peng and Zhao, Dazhe and Zaiane, Osmar, An optimized cost-sensitive SVM for imbalanced data learning, *Advances in Knowledge Discovery and Data Mining*:280–292, 2013.
- [36] Maldonado, Sebastián and Weber, Richard and Famili, Fazel, Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines, *Information Sciences*, **286**:228–246, 2014.
- [37] Jawanpuria, Pratik and Varma, Manik and Nath, Saketha, On p-norm path following in multiple kernel learning for nonlinear feature selection, *Proceedings of the 31st International Conference on Machine Learning*:118–126, 2014.
- [38] Tiwari, Deepika, Handling Class Imbalance Problem Using Feature Selection, *International Journal of Advanced Research in Computer Science & Technology*, **2**(2):516–520, 2014.
- [39] Kwok, James Tin-Yau and Tsang, Ivor Wai-Hung, The pre-image problem in kernel methods, *IEEE Transactions on Neural Networks*, **15**(6):1517–1525, 2004.
- [40] Rathi, Yogesh and Dambreville, Samuel and Tannenbaum, Allen, Statistical shape analysis using kernel PCA, *Electronic Imaging 2006*:60641B–60641B, 2006.

- [41] Gu, Bin and Sheng, Victor S and Wang, Zhijie and Ho, Derek and Osman, Said and Li, Shuo, Incremental learning for ν -support vector regression, *Neural Networks*, **67**:140–150, 2015.
- [42] Gu, Bin and Sheng, Victor S, A robust regularization path algorithm for ν -support vector classification, *IEEE Transactions on Neural Networks and Learning Systems*, 2016.
- [43] Gönen, Mehmet and Alpaydın, Ethem, Multiple Kernel Learning Algorithms, *Journal of Machine Learning Research*, **12**:2211–2268, 2011.
- [44] Dileep, Aroor Dinesh and Sekhar, C Chandra, Representation and feature selection using multiple kernel learning, *International Joint Conference on Neural Networks(IJCNN 2009)*:717–722, 2009.
- [45] Kloft, Marius and Brefeld, Ulf and Laskov, Pavel and Müller, Klaus-Robert and Zien, Alexander and Sonnenburg, Sören, Efficient and accurate l_p -norm multiple kernel learning, *Advances in neural information processing systems*:997–1005, 2009.
- [46] Zien, Alexander and Ong, Cheng Soon, Multiclass Multiple Kernel Learning, *Proceedings of the 24th international conference on Machine learning*:1191–1198, 2007.
- [47] Maratea, Antonio and Petrosino, Alfredo and Manzo, Mario, Adjusted F-measure and kernel scaling for imbalanced data learning, *Information Sciences*, **257**:331–341, 2014.
- [48] Chawla, Nitesh V. and Bowyer, Kevin W. and Hall, Lawrence O. and Kegelmeyer, W. Philip, SMOTE: synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research*, **16**(1):321–57, 2002.
- [49] Chen, Sheng and He, Haibo and Garcia, Edwardo A, RAMOBoost: ranked minority oversampling in boosting, *IEEE Transactions on Neural Networks*, **21**(10):1624–1642, 2010.
- [50] Yang, Jingjing and Li, Yuanning and Tian, Yonghong and Duan, Lingyu and Gao, Wen, Group-sensitive multiple kernel learning for object categorization, *IEEE 12th International Conference on Computer Vision*:436–443, 2009.

- [51] Bucak, Serhat S and Jin, Rong and Jain, Anubhav K, Multiple kernel learning for visual object recognition: A review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **36**(7):1354–1369, 2014.
- [52] Demšar, Janez, Statistical comparisons of classifiers over multiple data sets, *Journal of Machine learning research*, **7**:1–30, 2006.
- [53] Li Q, Recent progress in computer-aided diagnosis of lung nodules on thin-section CT, *Computerized medical imaging and graphics*, **4**:248–57, 2014.
- [54] Dolejsi, Martin and Kybic, Jan and Tuma, Stanislav and Polovincák, Michal, Reducing false positive responses in lung nodule detector system by asymmetric adaboost, *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*:656–659, 2008.
- [55] Cao, Peng and Yang, Jinzhu and Li, Wei and Zhao, Dazhe and Zaiane, Osmar, Ensemble-based hybrid probabilistic sampling for imbalanced data learning in lung nodule CAD, *Computerized Medical Imaging and Graphics*, **38**(3):137–150, 2014.
- [56] Yuan, Ming and Lin, Yi, Model selection and estimation in regression with grouped variables, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **68**(1): 49–67, 2006.
- [57] Dubey, Rashmi and Zhou, Jiayu and Wang, Yalin and Thompson, Paul M and Ye, Jieping and Alzheimer’s Disease Neuroimaging Initiative and others, Analysis of sampling techniques for imbalanced data: An n= 648 ADNI study, *NeuroImage*, **87**: 220–241, 2014.
- [58] Weiner, Michael W and Aisen, Paul S and Jack, Clifford R and Jagust, William J and Trojanowski, John Q and Shaw, Leslie and Saykin, Andrew J and Morris, John C and Cairns, Nigel and Beckett, Laurel A and others. The alzheimer’s disease neuroimaging initiative: progress report and future plans. *Alzheimers Dement*, **6**:202–211, 2010.
- [59] Ye, Jieping and Farnum, Michael and Yang, Eric and Verbeeck, Rudi and Lobanov, Victor and Raghavan, Nandini and Novak, Gerald and DiBernardo, Allitia and Narayan, Vaibhav A, Sparse learning and stability selection for predicting MCI to AD conversion using baseline ADNI data, *BMC neurology*, **12**(46): 1–12, 2012.

- [60] Wan, Jing and Zhang, Zhilin and Rao, Bhaskar D and Fang, Shiaofen and Yan, Jingwen and Saykin, Andrew J and Shen, Li, Identifying the neuroanatomical basis of cognitive impairment in Alzheimer’s disease by correlation-and nonlinearity-aware sparse Bayesian learning, *IEEE transactions on medical imaging*, **33**(7): 1475–1487, 2014.
- [61] Wang, Hua and Nie, Feiping and Huang, Heng and Risacher, Shannon L and Saykin, Andrew J and Shen, Li and Alzheimer’s Disease Neuroimaging Initiative and others, Identifying disease sensitive and quantitative trait-relevant biomarkers from multidimensional heterogeneous imaging genetics data via a sparse multimodal multitask learning, *Bioinformatics*, **28**(12): 127–136, 2012.
- [62] Zhu, Xiaofeng and Suk, Heung-Il and Lee, Seong-Whan and Shen, Ding-gang, Subspace regularized sparse multi-task learning for multi-class neurodegenerative disease identification, *IEEE Transactions on Biomedical Engineering*, **63**(3): 607–618, 2015.
- [63] Charte, Francisco and Rivera, Antonio J and del Jesus, María J and Herrera, Francisco, Addressing imbalance in multilabel classification: Measures and random re-sampling algorithms, *Neurocomputing*, **163**: 3–16, 2015.