A multi-kernel based framework for heterogeneous feature selection and over-sampling for computer-aided detection of pulmonary nodules

Peng Cao¹,b, Xiaoli Liu¹,b, Jinzhu Yang¹, Dazhe Zhao¹,b, Wei Li³, Min Huang⁴,d, Osmar Zaiane⁴

¹ College of Computer Science and Engineering, Northeastern University, Shenyang, China
² Key Laboratory of Medical Image Computing of Ministry of Education, Northeastern University, Shenyang, China
³ College of Information Science and Engineering, Northeastern University, Shenyang, China
⁴ State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China
⁵ Computing Science, University of Alberta, Edmonton, Alberta, Canada

ARTICLE INFO

Keywords:
Lung nodule detection
False positive reduction
Classification
Imbalanced data learning
Multi-kernel learning
Feature selection

ABSTRACT

Classification plays a critical role in False Positive Reduction (FPR) in lung nodule Computer Aided Detection (CAD). To achieve effective recognition of nodule, many machine learning methods have been proposed. However, multiple heterogeneous feature subsets, high dimensional irrelevant features, as well as imbalanced distribution between the nodule and non-nodule classes typically makes this problem challenging. To solve these challenges, we proposed a multi-kernel based framework for feature selection and imbalanced data learning in Lung nodule CAD, involving multiple kernel learning with a \( \ell_1 \) norm regularizer for heterogeneous feature fusion and selection from the feature subset level, a multi-kernel feature selection based on pairwise similarities from the feature level, and a multi-kernel over-sampling for the imbalanced data learning. Experimental results demonstrate the effectiveness of the proposed method in terms of Geometric mean (G-mean) and Area under the ROC curve (AUC), and consistently outperform the competing methods.

1. Introduction

Lung cancer is one of the main public health issues in developed countries [1], and early detection of solitary pulmonary nodules (SPNs) is an important clinical indication for early-stage lung cancer diagnosis because SPNs have high probabilities to become malignant nodules [2]. SPNs refer to lung tissue abnormalities that are roughly spherical with round opacity and a diameter of up to 30 mm. Currently, nodules are mainly detected by one or multiple expert radiologists inspecting CT images of lungs. Recent research, however, shows that inter-reader variability in the detection of nodules by expert radiologists may exist [4]. In addition, since three-dimensional (3D) image processing and analysis techniques become applicable in thin-section CT images [5,7], a thin-section CT scan includes hundreds of sections and requires considerable time and effort in image interpretation by radiologists. For more than a decade, significant effort has been focused on developing automated systems that detect/recognize suspicious lesions in thoracic CT imagery as well as other types of imagery.

It is therefore an important task to develop Computer Aided Detection (CAD) systems that can aid/enhance radiologist workflow and potentially reduce false negative findings. CAD is a scheme that automatically detects suspicious lesions (i.e. nodule, polyps and masses) in medical images of certain body part, and provides their locations to radiologists. CAD has become one of the major research topics in medical imaging and diagnostic radiology. Generally, a typical CAD systems for cancer detection and diagnosis (i.e.: Breast, Lung, Polyp) cover four stages as depicted in Fig. 1: (a) image preprocessing, (b) image segmentation, (c) feature extraction and selection, and finally (d) classification. The stages of (a) and (b) belong to the initial lesion identification step, and the stages of (c) and (d) belong to the false-positive reduction step. Current CAD schemes for nodule characterization have achieved high sensitivity levels, whereas current schemes for nodule detection appear to report many false positives [3,8–10]. It is because detection algorithms have high sensitivity that some non-nodule structures (e.g., blood vessels) are labeled as nodules inevitably in the initial nodule identification step. 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 false-positive reduction step, or classification step, the aim of which is to learn a system capable of the prediction of the unknown output class of a previously unseen suspicious nodule with a good generalization ability, is a critical part in the lung nodule detection system [21–23,19,51,33]. However, there are three significant problems in the classification of the potential nodules:

Received 27 June 2016; Received in revised form 13 October 2016; Accepted 5 November 2016
Available online 14 November 2016
0031-3203/ © 2016 Elsevier Ltd. All rights reserved.
1. Heterogeneous feature subsets: As we know, 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 (FeaSets) 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 FeaSets can potentially exploit their complementary information and improve the prediction. Thus, how to combine the multiple complementary FeaSets effectively based on different aspects is a critical challenge.

2. High dimensional irrelevant features: For the feature descriptor of nodule data, not all features are equally useful in removing false positives, some of which from the original feature space (input space) are irrelevant to the tasks at hand. It is usually not clear what the best feature descriptor to discriminate the nodule and non-nodule is.

3. Imbalanced data distribution between TPs and FPs: The two classes (nodule and non-nodule) are skewed and have extremely unequal misclassification costs, which is a typical class imbalance problem [11,29]. The imbalanced data issue usually occurs in computer-aided detection systems since the healthy class is far better represented than the diseased class in the collected data [30,32,12]. Class imbalanced data has detrimental effects on the performance of conventional classifiers, since. Typically classifiers attempt to reduce global error rate without taking the data distribution into consideration. As a result, standard classifiers usually tend to be overwhelmed by the majority class and ignore the minority class examples, resulting in poor performance.

In order to solve the issues above, we propose in this paper a multiple kernel framework to accurately classify potential nodule objects. The framework is named Multiple Kernel based feature Fusion and Selection and over-Sampling (MKFSS), which involves three key techniques: MKFF (Multiple Kernel based Feature Fusion), MKFS (Multiple Kernel based Feature Selection) and MKOS (Multiple Kernel based Over-Sampling) to overcome the issues introduced previously. More specifically, our main contributions to the lung CAD research are summarized as follows:

1. Heterogeneous feature fusion and selection from feature set level: Multiple kernel learning methods [15,42] not only learn an optimal linear combination of given base kernels, but also can be used to combine heterogeneous sources of data. In order to effectively fuse and select the heterogeneous FeaSets, we utilize multiple kernel learning with imposing the mixed $\ell_1$ norm constraint on the kernel weights, explicitly incorporating auxiliary information about structures in parameters, so as to exploit complementary heterogeneous FeaSets with different kernels and recognize the contribution of classification of different multiple FeaSets at the same time.

2. Feature selection from individual feature level based on feature weighting: Feature selection is an important preprocessing step in machine learning. Traditional feature selection algorithms work on a single data source only. In order to integrate multiple kernel learning and feature selection, we propose two multi-kernel based feature ranking techniques to select discriminative features according to their predictive powers (scores) in the kernel space induced by MKF, in order to decrease the size of the feature vectors and improve classification performance.

3. Imbalanced data learning based on over-sampling scheme: To solve the issue of imbalanced nodule data learning, a multi-kernel over-sampling method is proposed to balance the data distribution. The new over-sampling method in the optimal kernel space can generate accurate instances and address the problem of inconsistency which the over-sampling is carried out in the input space while the kernel classifier model is constructed in the kernel space.

All three methods are conducted in the kernel space based on the multiple kernel framework. Fig. 2 shows the schematic flowchart of the proposed MKFSS framework. From the preprocessed CT images, G heterogeneous FeaSets are extracted from the view of intensity, texture, shape and gradient. At first, the heterogeneous FeaSets are fused by multi-kernel learning in the kernel space. Then, feature selection and over-sampling are conducted sequentially based on the combined kernel induced by the proposed multi-kernel learning. Therefore, besides the optimal weight (kernel matrix) of different FeaSets, the optimal data representation (data matrix) with respect to features (columns) and samples (rows) is obtained. At last, a supervised kernel classifier SVM (Support vector machine) with the resulting optimal
The kernel is finally trained on the new training data to discriminate the true nodules versus false positives. An overview of the proposed classification pipeline including the procedure of training and predicting is illustrated in Fig. 3. To perform a validation with our system, we empirically investigate and compare the proposed method with the state-of-the-art approaches in the multi-source fusion, multiple kernel classification, class imbalance learning and the false positive reduction in lung nodule CAD. The experimental results show the unique properties of the proposed method for overcoming the challenges in the lung nodule CAD and demonstrate the promising effectiveness of this method compared with other competitive learning algorithms.

The structure of the paper is as follows: in Section 2 we review current state-of-the-art techniques for tackling the nodule classification problem. In Section 3 we present the multiple kernel learning. In Section 4, we introduce the proposed method MKFSS. In Section 5, the experimental results on a real dataset are given, which show excellent performance of the proposed method in comparison with other competitive methods. Finally, Section 6 concludes the paper.

2. Related work

Regarding classification of lung nodule, a number of techniques have been presented using machine learning approaches to classify samples as true positive nodule and false nodule, and this is also the focus of this paper; thus we only review the existing lung nodule classification methods from the machine learning perspective.

2.1. Linear classifier

Classifiers are designed to generate models from sample data and the models are desired to best predict the future input data. Various classifier models have been applied for reducing the false positive nodules. One of the most frequently employed and simplest classifiers is the rule-based classifier [18]; however it is hard to manually determine the selection of cut-off thresholds to classify abnormal and normal. Since linear discriminant analysis (LDA) offers simplicity in computation and effectiveness in classification, it is commonly used to discriminate the potential nodule [19].

2.2. Nonlinear classifier

Neural networks are often employed in nodule recognition tasks [20,22] as nonlinear model, which have the ability to automatically learn complex input–output relationships, and have low dependence on domain specific knowledge. However, in neural networks, it was difficult to determine the number of units in the hidden layer, and its gradient-based algorithm might be trapped in local minimal. In addition to neural networks, SVM [23–27] is the most commonly employed for reduction of false positives. Compared with neural networks, SVM tends to find a global solution during the training as the model complexity has been taken into consideration as a structural risk in SVM training, hence SVM has a strong generalization classification ability. SVM utilizes nonlinear mapping to make the data linear separable, hence the kernel function is the key. An important new trend is the appearance of ensemble learners which combine the decisions of multiple classifiers to form an integrated output, so as to enhance the generalization ability of a single model. Suzuki et al. have proposed a pixel-based massive training artificial neural network (MTANN) for distinction between nodule and FPs [22]. Lee et al. developed a random forest ensemble classification to improve the nodule classification performance [28].

2.3. Feature selection

Ge et al. [19] employed a stepwise process to select important features iteratively by adding new features to or removing features from the subset of currently selected features. They then used the selected features as the input of a linear discriminant analysis (LDA) technique for removing false positives. Boroczky et al. [23] used genetic algorithms to determine automatically the optimal size of the selected feature subset, and to choose the most relevant features from the entire feature set.
2.4. Imbalanced data learning methods for nodule classification

There is an important problem in the classification of potential nodule data. The dataset is typically imbalanced, and the costs of misclassification are different. Class imbalanced data has detrimental effects on the performance of conventional classifiers, resulting in lowering the performance of discrimination in the candidate nodule. However, in nodule classification, the problem has attracted less attention. The methods for solving imbalanced data can be grouped into two categories: the data perspective and the algorithm perspective [29]. The methods with the data perspective re-balance the class distribution by re-sampling the data space either randomly or deterministically. The authors in [23] use Tomek links to remove borderline false nodule cases in order to achieve 100% sensitivity. The authors in [33] proposed an ensemble based hybrid probabilistic sampling method, to solve the between-class and within-class imbalance simultaneously. Working with the algorithm perspective to adapt data sets is another way to deal with the imbalanced data problem. Cost-sensitive learning tries to learn more characteristics of samples with the minority class by assigning distinct costs to the training instances. Campadelli et al. prove that cost-sensitive SVM trained with imbalanced data sets achieves promising results in terms of sensitivity and specificity, by means of adjusting the misclassification cost of false positives versus false negatives [26]. Dolejsi et al. introduced an asymmetric Adaboost learning method to improve the sensitivity by setting different weights for two classes [34]. Peng et al. presented an effective wrapper incorporating the evaluation measure of imbalanced data into the objective function of cost-sensitive SVM [35]. Peng et al. extended the random subspace method to a novel cost sensitive adaptive random subspace ensemble, to increase the diversity among the components and to overcome imbalanced data classification [36].

3. Multiple kernel learning

Support vector machines (SVM) can model the nonlinear data distribution by mapping the input data into a nonlinear kernel space by kernel embedding [13,14]. However, the most suitable types and parameters of the kernels for a particular task are often unknown, and the selection of the optimal kernel by exhaustive search is usually time-consuming, and sometimes causes over-fitting. Multiple kernel learning (MKL) [15], which learns the optimal kernel by a weighted, linear combination of predefined candidate kernels, has been introduced to handle the problem of kernel selection. Moreover, MKL attempts to achieve better results by combining several base kernels instead of using only one specific kernel. MKL assumes that $x_i$ can be mapped to $Q$ different Hilbert spaces, $x_i \rightarrow \phi_q(x_i), q = 1, ..., Q$, implicitly with $Q$ nonlinear mapping functions $\phi_q: \mathbb{R} \rightarrow \mathcal{H}_q$ defines a nonlinear feature mapping from the original input space to a Hilbert space $\mathcal{H}_q$, and the objective of MKL is to seek the optimal kernel combination $K^*(x, x') = \sum_{q=1}^{Q} d_q K_q(x, x')$. $d_q \geq 0, \sum_{q=1}^{Q} d_q = 1$. We obtain the primal objective function of multiple kernel learning:

$$\min_{w, d, b, z} \frac{1}{2} \sum_{q=1}^{Q} \|w_q\|^2 + C \sum_{i=1}^{N} \xi_i, \text{ s.t. } \sum_{q=1}^{Q} \langle w_q, \phi_q(x_i) \rangle + b \geq 1 - \xi_i, \quad i = 1, ..., N \quad (1)$$

$$\sum_{q=1}^{Q} d_q = 1, \quad d_q \geq 0, \quad q = 1, ..., Q$$

where $w_q$ is the normal of the separating hyperplane for the feature mapping $\phi_q$, $b$ is the bias term, $\xi = [\xi_1, ..., \xi_N]$ is the vector of slack variables, and $C$ is the misclassification penalty.

Since Eq. (1) is non-convex due to the products of $d_q$ and $w_q$, it can be resolved by applying the variable transformation $w'_q = \sqrt{d_q} w_q$ as that in [37,38].
zero if the corresponding feature subset is irrelevant to the output. The mixed \( \ell_2,1 \) constraint imposed on \( D \) will maintain sparsity between different \( \text{FeaSets} \) so as to generate a compact set of features, while the associated values in each group need not be sparse, allowing our MKL algorithm to select more than one base kernel for each feature subset.

The objective function of \( \ell_2,1 \) norm regularized MKL is:

\[
\min_{w,D,b} \frac{1}{2} \sum_{q=1}^{G,Q} \frac{||w_q||^2}{d_q} + C \sum_{i=1}^{N} \xi_i \|D\|_{l,1} \quad \text{s.t.} \quad \sum_{q=1}^{G,Q} d_{iq} S_{iq} \geq 0, \quad \forall \ p, \ q; \quad \xi_i \geq 0, \quad \forall \ i
\]

where \( x_{i}^{(g)} \) indexes the \( g \)-th feature subset of the instance of \( x_i \), \( C \) and \( C_{ij} \) are regularization parameters.

The primal formulation (4) can be seen as the composite objective optimization problem:

\[
\min_{D \geq 0} J(D) + C ||D||_{l,1}
\]

where

\[
J(D) = \left\{ \begin{array}{ll}
\min_{w,D,b} & \frac{1}{2} \sum_{q=1}^{G,Q} \|w_q\|^2 \frac{1}{d_q} + C \sum_{i=1}^{N} \xi_i \\
\text{s.t.} & \sum_{q=1}^{G,Q} \frac{w_q}{d_q} (x_{i}^{(g)}, y_i) + b \geq 1 - \xi_i, \quad i = 1, \ldots, N
\end{array} \right.
\]

The Lagrangian of problem (6):

\[
L = \frac{1}{2} \sum_{q=1}^{G,Q} \|w_q\|^2 \frac{1}{d_q} + C \sum_{i=1}^{N} \xi_i + \sum_{i=1}^{N} \sum_{q=1}^{G,Q} (w_q \phi_q(x_{i}^{(g)}))' y_i b - \sum_{i=1}^{N} \xi_i b_i
\]

where \( \alpha_i \) and \( \beta_i \) are the Lagrange multipliers of the constraints.

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

\[
\max \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i,j=1}^{N} a_i a_j y_i y_j \sum_{q=1}^{G,Q} \frac{d_{iq}}{d_j} \phi_q(x_{i}^{(g)})' \phi_q(x_{j}^{(g)})
\]

\[
C \geq a_i \geq 0; \quad a_i y_i = 0 \quad \forall \ i
\]

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

\[
J(D) = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i,j=1}^{N} a_i a_j y_i y_j \sum_{q=1}^{G,Q} \frac{d_{iq}}{d_j} \phi_q(x_{i}^{(g)})' \phi_q(x_{j}^{(g)})
\]

where
\[ K^i = \begin{bmatrix} K_{21}(x_1^{(1)}, x_1^{(1)}) & \cdots & K_{2i}(x_1^{(1)}, x_1^{(1)}) \\ \vdots & \ddots & \vdots \\ K_{Qi}(x_i^{(1)}, x_i^{(1)}) & \cdots & K_{Qi}(x_i^{(1)}, x_i^{(1)}) \end{bmatrix} \]  

Eq. (5) is a convex objective, where \( J(D) \) is convex and differentiable, while \( \|D\|_1 \) is convex but non-smooth. To optimize \( D \), we efficiently solve it by designing a new accelerated proximal gradient method with the idea of FISTA (fast iterative shrinkage-thresholding algorithm) [17] 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:

\[ Q_i(D, D^0) = \arg\min_{D} J(D^0) + \langle D - D^0, \nabla J(D^0) \rangle + \frac{L^0}{2} \|D - D^0\|^2 \]

+ \( C_f \|D\|_1 \)  

where

\[ \nabla J(D^0) = \{\nabla J(d^{(0)}_q), \ldots, \nabla J(d^{(0)}_{Qg})\} \]  

and

\[ \nabla J(d^{(0)}_q) = -\frac{1}{2} \sum_{i=1}^{N} a_i^q a_j^q y_j K_{q}(x_i^q, x_j^q) \quad \forall \ p, q \]  

The differentiability of \( J(d^{(0)}_q) \) is ensured by the uniqueness of \( a^q \) and by the differentiability of the objective function. Ignoring constant terms in \( D^0 \), the unique minimizer of the above expression can be written as Eq. (14), by which \( D \) is iteratively updated by proximal gradient method:

\[ D^{(i+1)} = \arg\min_{D} Q_i(D, D^0) = \arg\min_{D} J(D^0) + \langle D - D^0, \nabla J(D^0) \rangle + \frac{L^0}{2} \|D - D^0\|^2 \]

+ \( C_f \|D\|_1 \)  

\[ = \arg\min_{D} \|D\|_1 \|D - D^0\|^2 + \frac{L^0}{2} \|D - D^0\|^2 \]

The minimizer of the above expression can be written as:

\[ D^{(i+1)} = \pi_{C_f}(U^{(0)}) = \arg\min_{D} \|D - U^{(0)}\|^2 + \frac{L^0}{2} \|D\|_1 \]  

where \( \pi_{C_f}(\cdot) \) can be viewed as a proximal operator corresponding to the non-smooth penalty of \( \|D\|_1 \), \( U^{(0)} = D^{(0)} - \frac{1}{L^0} \nabla J(D^{(0)}) \). \( L^0 \) plays the role of a stepsize and is determined by the line search. Meanwhile, in order to accelerate the proximal gradient method in Eq. (15), we consider the proximal operator \( \pi_{C_f}(\cdot) \) at a specific linear combination of the previous two iterates \( D^{(i-1)}, D^{(i-2)} \), instead of just the previous iterate \( D^{(i-1)} \) as follows:

\[ V^{(i)} = D^{(i-1)} + \frac{d_{i+1}}{2} - \frac{1}{a_{i+1}} (D^{(i-1)} - D^{(i-2)}) \]  

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

\[ d^{(i+1)} = \arg\min_{d_t} \|d_t - u_t^{(0)}\|^2 + \frac{C_f}{L^0} \|d_t\|_1 \]  

where \( d_t \) and \( u_t^{(0)} \) are the \( q \)-th rows of matrix \( D \) and \( U \) respectively.

Following [43], the row-wise updates can be done by soft-thresholding as

\[ d_t^{(i+1)} = \max \left( \frac{|d_t^{(i)}| - \frac{C_f}{L^0}}{2}, 0 \right) \]  

Moreover, a projection operator \( P: P(d_t^{(i+1)}) = \max(0, d_t^{(i+1)}) \) is introduced ensuring that the non-negativity constraints on \( d_t^{(i+1)}(d_t \geq 0) \) are satisfied. The algorithm for updating kernel weighting \( D \) by the FISTA method is given in Algorithm 2. Because the problem in Eq. (5) is a convex problem, the global optimum solution to the problem can be obtained. Moreover, the Algorithm 2 will monotonically decrease the objective of the problem in Eq. (5) in each iteration and converge to the global optimum of the problem. As shown in [17], such an algorithm is guaranteed to converge at a rate \( O(1/t^2) \). Through the optimization, the weights \( D \) can be learned to reflect the relative importance of different FeaSets. The final discriminant function is:

\[ f(x) = \text{sign} \left( \sum_{i=1}^{N} a_i^q y_i \sum_{g=1}^{Q} d_{g}^{(0)} K_{g}(x^q, x^{(0)}) + b^q \right) \]  

Algorithm 2. FISTA method (\( f_{2,1} \) MKL) for optimizing \( D \) in Eq. (5).

Input: Training Data \( X \), class label \( Y \), \( L_0 \geq 0 \), \( d_{q0} \in [0, 1] \)

Output: Weights of kernel \( D \)

1: \( D^0 = [d_{10}, \ldots, d_{Q0}] \)

2: \( \psi^0 = D^0 \)

3: \( t = 1 \) to \( \ldots \) do

4: Find the smallest nonnegative integers \( i_t \) such that with

\[ L^0 = 2\|V^{(0)}\| \quad J(d^{(0)}_q(U^{(0)})) \leq Q_i(U, U^{(0)}) \quad \text{and} \quad J(D^{(0)}) = 2\|V^{(0)}\| \]

5: Calculate \( J(D^{(0)}) \) with \( X, Y \) according to Eq. (9), and \( V^{(0)} \) according to Eq. (12)

6: \( g = 1 \) to \( G \) do

7: \( \psi_t^{(g)} = \psi_t^{(g-1)} - \frac{1}{L^0} \nabla J(\psi_t^{(g-1)}) \)

8: Compute \( d_t^{(g)} = \arg\min_{d_t} \frac{1}{2} \|d_t - u_t^{(g)}\|^2 + \frac{C_f}{L^0} \|d_t\|_1 \) by solving Eq. (18)

9: \( d_t^{(g+1)} = \psi_t^{(g)} \)

10: \( \psi_t^{(g+1)} = d_t^{(g+1)} + \frac{1}{a_{g+1}} (d_t^{(g)} - d_t^{(g-1)}) \)

11: \( d_t^{(g+1)} = \psi_t^{(g+1)} \)

12: \( \text{end for} \)

13: if convergence then

14: \( D = D^{(i+1)} \)

15: \( \text{end if} \)

16: \( \text{end for} \)

4.3. Multi-kernel feature selection

After completing the feature fusion and selection from the perspective of feature group, there still exist irrelevant features in the remaining ones. Feature selection can reduce dimensionality, making the sequential computation on the input data more efficient. Moreover, the noisy features are eliminated for a better data representation, resulting in a more accurate classification result. Feature selection algorithms can be roughly classified into two groups, i.e., supervised feature selection and unsupervised feature selection. Hence, we perform two different feature selection SPEC (Spectral Feature Selection, unsupervised) [44] and ReliefF (Relief Feature Selection, supervised) [45] combined with the multi-kernel induced by our multi-kernel learning method. The linear distance is commonly used as a similarity measure, but the data may lie on a nonlinear distribution. The combined kernel matrix optimized can extract the pattern of the data in the form of pairwise similarities, thus handle this problem by mapping the data to a high-dimensional. The optimal kernel matrix learned can represent much more appropriate relationships among instances. Therefore, the multi-kernel matrix can then be incorporate into generic feature selection algorithm as a similarity measure. The following subsections give a brief introduction to each of the two feature ranking methods.
Firstly, we employ the $\xi_f$ (the opposite side, resulting in that the instance generated in original input space is a distance measurement. In order to based on distance constraints. Moreover, the $\mathcal{F}$ mapping from and input-space distance is:

$$f$$ and $\gamma$ can be directly obtained from the learned combined kernel $\phi$ and $\alpha$ in the kernel space rather than the original $\phi$. Finally, we is MKFS. Then, the new instance is generated between $x_{\text{new}}$ and $x_i$ in the kernel space rather than the original $\phi$. The nearest point to $x_{\text{new}}$ and $x_i$ in the kernel space rather than the original $\phi$. Oversampling in the kernel 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.

4.3.1. Multi-kernel unsupervised spectral based feature selection

Spectral feature selection (SPEC) is to select features according to the structures of the graph induced from $\mathcal{F}$ [44]. Firstly, we employ the spectrum of the graph to measure feature relevance. The graph $G$ is constructed by a set of pairwise instance similarity $\mathcal{F}$. The similarity matrix $\mathcal{S}$ can be directly obtained from the learned combined kernel matrix by MKFF algorithm. Based on the $G$, SPEC algorithm selects features in terms of the smoothness on the manifold formed by the observed data instances. The smoothness of a feature means that the feature which is consistent with the graph structure assigns similar values to instances that are near each other on the graph. Given the normalized Laplacian matrix $L$ of $G$, we calculate its spectral decomposition $(\lambda_i, \xi_i)\ i = 1,...,m$ ($m$ is the number of eigenvalues of $L$) denotes the spectrum decomposition of the normalized Laplacian matrix $L$.

According to spectral clustering theories, the eigenvalues of normalized Laplacian matrix $L$ measure the separability of the clusters of the graph $G$ and the eigenvectors are the corresponding soft cluster indicators. The better the $f$ aligns closely to the nontrivial eigenvectors with small eigenvalues of $L$, the better the feature $f_i$ can separate the training data. Therefore, a robust smoothness measure function is defined as:

$$r(f) = \frac{\sum_{j=1}^{m} a_i^2 \gamma(J_i)}{\sum_{j=1}^{m} a_j^2} = 1 - \tilde{f}_i^T \gamma(L) \tilde{f}_i$$  \hspace{1cm} (20)

where $\tilde{f}_i$ normalized weighted feature vector, defined as: $\tilde{f}_i = \frac{a_i f_i}{\|a_i f_i\|}$, $a_i = \cos \theta_i$ where $\theta_i$ is the angle between $f_i$ and $\xi_i$. It measures the similarity between the feature vector and the eigenvectors. The first called $(\lambda_0, \xi_0)$ is the trivial eigenpair of the graph, which is ignored.

4.3.2. Multi-kernel supervised margin based feature selection

Relief Feature Selection (ReliefF) is a margin based feature selection approach. It determines the relevance of a feature according to its contribution to the hypothesis margin of the instances with the given labels from the training data [45]. We incorporate the multi-kernel into the relevance criterion of features. Hence, the relevance of a feature $f_i$ is defined as:

$$r(f_i) = \frac{1}{2} \sum_{j=1}^{N} (\|x_{i,j} - \text{Nearhit}(x_i)\| - \|x_{i,j} - \text{Nearmiss}(x)\|)$$  \hspace{1cm} (21)

where $x_{i,j}$ denotes the value of instance $x_i$ on feature $f_i$, Nearhit($x$) and Nearmiss($x$) denote the nearest point to $x$ with the same and the different label, respectively, and $\|\|$ is a distance measurement. In order to obtain more exact neighborhoods of each instance, the learned optimal kernel matrix by MKFF algorithm is used to calculate the nearest instances.

4.4. Multi-kernel over-sampling

The methods with the data perspective re-balance the class distribution either by over-sampling instances of the minority class or by undersampling instances of the majority class. While many re-sampling methods have been proposed in the literature, 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 kernel 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 over-sampling.

To address the issue of traditional over-sampling, we propose a multi-kernel over-sampling (MKOS), which is carried out in the kernel space after feature selecting, hence it can alleviate the negative influence due to the irrelevant and redundant features. Moreover, the proposed over-sampling can work in the same kernel space as kernel classifier. Specifically, we extend SMOTE [47] into kernel space by using the optimal kernel matrix optimized by $\ell_2,1$ MKFS. Utilizing the strategy of SMOTE procedure, MKOS generates artificial example $x_{\text{new}}$ located on the path connecting selected minority example $\phi(x_i)$ and one of its closest neighbor $\phi(x_j)$ in the kernel space rather than the original input space:

$$\phi(x_{\text{new}}) = \phi(x_i) + \lambda_{ij} \times (\phi(x_i) - \phi(x_j))$$  \hspace{1cm} (22)

The idea of the proposed multi-kernel over-sampling is shown in Fig. 6. After creating artificial data based on the kernel space similarities between existing minority examples according to eq. (22), we need to find the pre-image of the artificial data in the input space. Since we choose the RBF kernel: $K(x, x') = \exp(-\|x - x'\|^2)$ in our framework, the relationship between feature-space distance $d_{\text{f}}(\phi(x_{\text{new}}), \phi(x'))$ and input-space distance $d_{\text{i}}(x_{\text{new}}^*, x')$ is:
\[ d_{\text{new}}^2(\phi(x_{\text{new}})), \phi(x')| = \| \phi(x_{\text{new}}) - \phi(x') \|^2 = K^*(x_{\text{new}}, x_{\text{new}}) - 2 \bar{K}^*(x_{\text{new}}, x') + \bar{K}^*(x', x') \\
= 2 - 2 \exp(-\| x_{\text{new}} - x' \|^2/(2\sigma^2)) \\
= 2 - 2 \exp(-d_{\text{new}}^2(\phi(x_{\text{new}}), x')/(2\sigma^2)) \]  

According to Eq. (22), the distance between the new instance \( x_{\text{new}} \) and any instance \( x' \):

\[ d_{\text{new}}^2(\phi(x_{\text{new}})), \phi(x')| = \| \phi(x_{\text{new}}) - \phi(x') \|^2 \\
= d_{\text{new}}^2(\phi(x) + \lambda \times (\phi(x) - \phi(x))), x') \\
= 2(\lambda - 1)K^*(x', x) - 2\lambda_0K^*(x, x') + (\lambda_0^2 - 1)^2K^*(x, x) + 2\lambda_0(1 - \lambda_0)K^*(x, x') + \lambda_0^2K^*(x, x) \]

where \( K^* \) is the combined kernel matrix optimized by \( C_{\text{MKFS}} \).

A unique pre-image \( \tilde{x}_{\text{new}} \) can be estimated with its \( K_2 \) nearest neighbors in the kernel space according to [31] as follows:

\[ \tilde{x}_{\text{new}} = \sum_{p=1}^{K_2} e_p \exp \left( \frac{-d_{\text{new}}^2(\phi(x_{\text{new}}), x_p)}{2\sigma^2} \right) x_p \]

\[ = \sum_{p=1}^{K_2} e_p \exp \left( \frac{-d_{\text{new}}^2(\phi(x_{\text{new}}), x_p)}{2\sigma^2} \right) x_p \]

\[ = \sum_{p=1}^{K_2} e_p \exp \left( \frac{-d_{\text{new}}^2(\phi(x_{\text{new}}), x_p)}{2\sigma^2} \right) x_p \]

\[ = \sum_{p=1}^{K_2} e_p \exp \left( \frac{1}{2} \left( 2 - 2 \exp(-d_{\text{new}}^2(\phi(x_{\text{new}}), \phi(x'))) \right) x_p \right) \]

\[ = \sum_{p=1}^{K_2} e_p \exp \left( \frac{1}{2} \left( 2 - 2 \exp(-d_{\text{new}}^2(\phi(x_{\text{new}}), \phi(x'))) \right) x_p \right) \]

where \( y_p \) is a normalized value of distance between \( \phi(x_{\text{new}}) \) and \( \phi(x') \) according to (22). The MKOS algorithm is detailed in Algorithm 3.

Algorithm 3. Multi-kernel over-sampling.

Input:
- Training data set \( Tr \), oversampling ratio \( R \), number of nearest neighbors \( K_1 \), for over-sampling, number of nearest neighbors \( K_2 \) for finding pre-image

Output:
- The new balanced dataset \( Tr' \)
1: \( Tr_\text{new}=\text{NULL} \)
2: \( M = (\text{int})(R/100) \)
3: Obtain the minority class data set \( Tr_\text{min} \) from \( Tr \)
4: for \( i=1; i < |Tr_\text{min}|; i++ \) do
5: \( S_i = \text{Populate}(x_i, Tr_\text{min}, K_1, K_2, M) \) *Function to generate the synthetic samples and estimate the pre-images*
6: \( Tr_\text{new}=Tr_\text{new} \cup S_i \)
7: end for
8: \( Tr' = Tr' \cup Tr_\text{new} \)

Algorithm 4. Populate.

Input:
- Seed instance \( x \), minority training data set \( Tr_\text{min} \), number of nearest neighbors \( K_1 \) for over-sampling, number of nearest neighbors \( K_2 \) for finding pre-image, number of new instances for seed instance \( M \)

Output:
- The new instances in the original space \( S \)
1: \( S=\text{NULL} \)
2: \( NN = \text{findKNN}(\phi(x), K_1, Tr_\text{min}) \) *Function to find the K nearest neighbors*
3: while \( M > 0 \) do
4: Randomly choose one minority data example from \( NN \), \( \phi(x) \)
5: \( \phi(x^\text{new}) = \phi(x) + \lambda \times (\phi(x) - \phi(x')) \) according to Eq. (22)
6: \( NN_2 = \text{findKNN}(\phi(x^\text{new}), K_2, Tr_\text{min}) \)
7: Obtain the pre-image \( x^\text{new} \) with nearest neighbors \( NN_2 \) according to Eq. (25)
8: \( S = S \cup x^\text{new} \)
9: \( M = M - 1 \)
10: end while

Algorithm 5. findKNN.

Input:
- Seed instance \( \phi(x) \), number of nearest neighbors \( K \), data set \( Tr \)
Output:
- The set of nearest neighbors \( KNN \)
1: for each instance \( \phi(x_i) \) in \( Tr \) do
2: Calculate the distance \( d_{\text{dis}} \) between \( \phi(x) \) and seed \( \phi(x_i) \) in the kernel space
3: end for
4: Sort \( d_{\text{dis}} \) in descending order
5: \( KNN = \text{getFirstInstances}(DIS, K) \)

5. Experimental study

5.1. CAD system

Unlike other publications in which candidate nodules were selected manually for classification evaluation, we obtained the appropriate candidate nodule samples objectively using a candidate nodule detection algorithm. Our CAD scheme also contains lung segmentation, candidate nodule detection and VOI (Volume of Interest) segmentation.

5.2. Initial nodule detection

The database used in this work is the LIDC-IDRI. 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.

In the detection phase, we use the dot enhancement filter proposed by Li [49], which is aimed to simultaneously enhance objects of a specific structure (e.g. dot-like nodules) and suppress objects of other structures (e.g. line-like vessels) in multi-scale. If the nodule diameters are in the range [\( d_{\text{min}}, d_{\text{max}} \)], the scales to be considered for the Gaussian smoothing is in the range [\( d_{\text{min}}/4, d_{\text{max}}/4 \)]. The amount of scales is set to 5. After the 3D selective enhancement Hessian filter is applied on the original image, we use a thresholding technique and a 3D connected-component labeling technique to separate nodule candidates and identify all of the isolated objects [50]. The value of the threshold in the thresholding technique is set to 40, and objects with an effective diameter smaller than 2.5 mm are considered to be non-nodules and are eliminated [50].

For each nodule candidate object, we developed a 3D constrained region-growing technique to segment it accurately in the original CT
images. Fig. 7 shows some example result images of candidate VOI detection. The detected true nodule has different location and connection with the surrounding pulmonary structures. 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 from the 197 CT cases with 237 solid nodules. After obtaining the candidate nodule instances, we calculate the features for each of the nodule candidates from intensity, shape, texture and gradient distribution aspects. Then, we need to reduce the false positive nodule instances with our proposed classification model.

5.3. Feature extraction

Feature extraction plays an important role in the classification of suspicious nodules [19,51]. However, there is not a single outstanding feature that can discriminate the nodule from non-nodule completely. This is due to the fact that the nodules vary enormously in volume, shape, and appearance, and the sources of false positives are different. The majority of false positives are mainly caused by blood vessels and other normal anatomic structures. Some of the false positives can be easily distinguished from true nodule, however, a large portion of them are difficult to distinguish. Therefore, for getting a high accuracy in candidate nodule classification, we should extract more features from many aspects, such as intensity, shape and gradient distribution. Our feature extraction process generated 91 image features from volume of interest (VOI) for each potential nodule object. Using these features, we construct the input space for our classifiers. All of the extracted feature descriptors listed in Table 1 are described in our previous work [33].

5.4. Evaluating the effectiveness of MKFSS

To evaluate the classification performance of our proposed method in false positive reduction, we present several experiments to answer the following questions.

(1) How is the learning capacity of the \( C_1 \) norm as the regularization parameter? (2) Can the classification performance of the high dimensional and imbalanced nodule data be increased by feature selection and over-sampling in the kernel space? (3) How does the performance of our MKFSS compare to the state-of-the-art methods of imbalanced data learning and false positive reduction?

With these questions in mind, we will give separate discussions for each type under a set of artificial scenarios. The experiments consist of vertical comparison and horizontal comparison separately. The vertical evaluation in Experiments I to IV involves the comparison with individual feature subset as well as the other fusion strategies of FeaSets, the state-of-the-art multiple kernel learning methods, the different feature selection strategies of MKFS and the influence of the over sampling ratio parameter of MKOS. The horizontal one in Experiments V and VI includes the comparison with the state-of-the-art methods for dealing with imbalanced data in the field of machine learning and false positive reduction of Lung CAD.

All the experiments are carried out by means of a 10-fold 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. For all the kernel methods, the regularization parameter \( C \) is chosen by cross-validation. A 5 fold cross-validation is conducted on the training set to optimize the value of \( C \) (try values 0.01, 0.1, 1, 10, 100). The parameters of our method \( \alpha \) and \( \beta \) in Eq. (4)) are optimized in the same range. The commonly used kernel functions are BRF kernel, and we use BRF kernels with six different kernel bandwidths \( (2^{-2}, 2^{-1}, \ldots, 2^3) \) for each feature subset, which totally yields 48 kernels. 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 \([48]\) of the new synthetic instances in the kernel space), and \( R \) (over-sampling ratio). In our experiments, the parameters are: \( K_1 = 10, K_2 = 10, R = 300 \). All these values are chosen after some preliminary runs and they show the best results. We use metrics such as sensitivity, specificity, G-mean and AUC to evaluate the performance of the learning algorithm on imbalanced nodule candidate data.

\[
G\text{-mean} = \sqrt{\text{sensitivity} \times \text{specificity}}
\]

\[
sensitivity = \frac{TP}{TP + FN} \quad \text{specificity} = \frac{TN}{TN + FP}
\]

Table 1

<table>
<thead>
<tr>
<th>FeaSet id</th>
<th>Feature id</th>
<th>Category</th>
<th>Feature Set (FeaSet)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>f₁–f₁</td>
<td>Intensity</td>
<td>Intensity statistical feature</td>
<td>The gray value within the objects was characterized by use of seven statistics (mean, variance, max, min, skew, kurt, entropy)</td>
</tr>
<tr>
<td>F2</td>
<td>f₂–f₂</td>
<td>Intensity</td>
<td>Sub-volume distribution feature</td>
<td>The average intensity within each sub-volume along the radial directions</td>
</tr>
<tr>
<td>F3</td>
<td>f₃–f₃</td>
<td>Texture</td>
<td>Shape</td>
<td>The mean value of each filtered image</td>
</tr>
<tr>
<td>F4</td>
<td>f₄–f₄</td>
<td>Texture</td>
<td>Shape</td>
<td>The volumetric shape index (SI) representing the local shape feature at each voxel was characterized by use of seven statistics</td>
</tr>
<tr>
<td>F5</td>
<td>f₅–f₅</td>
<td>CV</td>
<td>CV statistical feature</td>
<td>The volumetric curvedness (CV), which quantifies how highly curved a surface is, was characterized by use of seven statistics</td>
</tr>
<tr>
<td>F6</td>
<td>f₆–f₆</td>
<td>Geometric feature</td>
<td>Volume, surface area and compactness</td>
<td></td>
</tr>
<tr>
<td>F7</td>
<td>f₇–f₇</td>
<td>Gradient</td>
<td>GC statistical feature</td>
<td>The concentration characterizing the degree of convergence of the gradient vectors at each voxel, was characterized by use of seven statistics</td>
</tr>
<tr>
<td>F8</td>
<td>f₈–f₈</td>
<td>GS</td>
<td>GS statistical feature</td>
<td>The gradient strength of the gradient vectors at each voxel, was characterized by use of seven statistics</td>
</tr>
</tbody>
</table>
where $TP$ denotes the number of true positives, $FP$ denotes the number of false positives, $FN$ denotes the number of false negatives, and $TN$ the number of true negatives.

5.4.1. Experiment I: The comparison with the individual feature subset and other feature fusion methods

The mixed $\ell_2$ norm formulation enforces group sparsity among different $FeaSets$, 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. 8. Fig. 8 shows the weight for each base kernel and the associated variable. The weight values correspond to the contribution for representations. 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 retained the $FeaSets$ with high weight as the most discriminative feature representation and remove the $FeaSets$ of Gabor and Gradient strength with least importance, so as to obtain the heterogeneous feature subset fusion and feature selection from the view of the $FeaSets$.

Depending on the task, different fusion methods lead to different classification performances. There are three main types of fusion schemes, depending on the stage the information is combined. We investigate the three different types of data fusion methods, namely early (feature-level fusion) [23,50], late (decision-level fusion) [33,22], and intermediate (kernel-level fusion) fusion [40,41] in our experiment. The feature-level methods combine multiple $FeaSets$ into a single fused one that is then used in a conventional SVM classifier, whereas decision-level ones combine several diverse classifiers to make a stronger final classifier.

We compare our proposed multiple kernel learning based method with the two other fusion strategies for heterogeneous $FeaSets$ information. SVM-All (feature-level fusion) concatenate all the $FeaSets$ from the different sources into a single feature vector. 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. Ensemble classifiers (decision-level fusion) construct multiple base classifiers, then combine the predicted results by different classifiers via voting to reach the final output, such as bootstrap (Bagging), random subspace method (RSM) and Adaboost. For three ensemble methods, the amount of base classifier is fixed to 50. Besides the comparable methods, the comparison is conducted between our method and the intermediate method in MKFSS, such as MKFF ($\ell_2$ norm multi-kernel method without feature selection and over-sampling), MKF ($\ell_1$ norm multi-kernel method combined with Relief Feature Selection without over-sampling), MKOS ($\ell_1$ norm multi-kernel method combined with over-sampling without feature selection), MKFSS ($\ell_1$ norm multi-kernel method combined with over-sampling and feature selection, but the over-sampling is conducted prior to the Relief Feature Selection) as well. In addition, we also report the classification performances by SVM on each individual type of features as baselines.

From the results in Table 2, 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. Even a simple concatenation can improve the performance. It demonstrates that different sources of feature set provide complementary information, which may be useful for classification of nodule candidates when used together. Among the different fusion methods, we empirically demonstrate that MKFSS outperforms the other two common fusion methods. Our MKFSS can successfully recognize 184 true nodules from 209 in 10-fold cross-validation. We can see that under the condition where the feature selection and over-sampling are not carried out, compared with the two other fusion methods, MKF have a higher performance due to the combination of kernels instead of using only one specific kernel. For the feature-level fusion, features extracted from different views are highly correlated. Thus, simple normalization is often not enough to effectively fuse features. For the decision-level fusion of ensemble classifiers, only the final decision classification outputs are used, resulting in information loss. Thanks to the integration of $\ell_1$ norm into multiple kernel learning, the proposed algorithm is able to accurately fuse the complementary and heterogeneous feature sets, and automatically prune the irrelevant and redundant $FeaSets$ to form a more discriminative feature set.

From Table 2, we can also observe that the performance of the

![Fig. 8](image-url)

Fig. 8. The kernel weights of heterogeneous $FeaSets$ of MKF algorithm. Different colors indicate the base kernel functions with different kernel bandwidths ($2^{-3}, 2^{-2,...,2^{3}}$).

Table 2

<table>
<thead>
<tr>
<th>Method</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>G-mean (%)</th>
<th>AUC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM-Inte</td>
<td>57.77 ± 3.09 (&lt;0.01)</td>
<td>87.19 ± 1.52* (0.029)</td>
<td>71.38 ± 2.50* (0.018)</td>
<td>74.31 ± 3.44* (0.009)</td>
</tr>
<tr>
<td>SVM-Text</td>
<td>52.47 ± 4.27 (&lt;0.01)</td>
<td>84.66 ± 3.34* (0.011)</td>
<td>67.12 ± 1.99* (0.013)</td>
<td>71.74 ± 4.16* (0.008)</td>
</tr>
<tr>
<td>SVM-Grad</td>
<td>56.32 ± 3.57 (&lt;0.01)</td>
<td>86.05 ± 4.19* (0.022)</td>
<td>69.85 ± 3.67* (0.017)</td>
<td>72.48 ± 4.01* (0.009)</td>
</tr>
<tr>
<td>SVM-Shape</td>
<td>59.72 ± 3.89 (&lt;0.01)</td>
<td>88.42 ± 3.22* (0.022)</td>
<td>73.11 ± 4.19* (0.032)</td>
<td>74.91 ± 3.20* (0.013)</td>
</tr>
<tr>
<td>SVM-All</td>
<td>64.69 ± 5.17 (&lt;0.01)</td>
<td>89.74 ± 4.78* (0.033)</td>
<td>76.55 ± 4.19* (0.025)</td>
<td>77.36 ± 3.34* (0.014)</td>
</tr>
<tr>
<td>MKFF</td>
<td>75.23 ± 1.59* (0.023)</td>
<td>91.33 ± 2.32* (0.037)</td>
<td>82.03 ± 1.99* (0.033)</td>
<td>87.23 ± 2.89* (0.036)</td>
</tr>
<tr>
<td>MKFS</td>
<td>79.23 ± 2.86* (0.028)</td>
<td>93.27 ± 2.88* (0.143)</td>
<td>85.57 ± 2.65* (0.037)</td>
<td>91.39 ± 3.33* (0.043)</td>
</tr>
<tr>
<td>MKOS</td>
<td>83.60 ± 3.82* (0.031)</td>
<td>91.55 ± 4.18* (0.066)</td>
<td>87.49 ± 1.89* (0.044)</td>
<td>90.16 ± 4.66* (0.037)</td>
</tr>
<tr>
<td>MKFSS</td>
<td>87.55 ± 3.12</td>
<td>92.41 ± 2.84</td>
<td>89.95 ± 3.16</td>
<td>94.17 ± 2.41</td>
</tr>
<tr>
<td>Bagging</td>
<td>85.62 ± 4.05* (0.038)</td>
<td>92.75 ± 4.18* (0.095)</td>
<td>89.10 ± 2.20* (0.133)</td>
<td>92.88 ± 3.97* (0.078)</td>
</tr>
<tr>
<td>RSM</td>
<td>70.82 ± 3.33* (0.012)</td>
<td>91.57 ± 4.07* (0.098)</td>
<td>80.66 ± 2.44* (0.027)</td>
<td>79.59 ± 2.61* (0.022)</td>
</tr>
<tr>
<td>Adaboost</td>
<td>72.11 ± 4.39* (0.015)</td>
<td>92.72 ± 3.96* (0.154)</td>
<td>81.74 ± 4.52* (0.026)</td>
<td>82.45 ± 3.50* (0.018)</td>
</tr>
</tbody>
</table>

(+) denotes the number of true positives, (−) the number of false negatives, and (%) the number of true negatives.
kernel classifier, regardless of single SVM classifier or multiple kernel classifier, is affected by the presence of class imbalance and high irrelevant features, which demonstrates that feature selection and over-sampling were both important for the classification of nodule candidate data, but the most important factor was the over-sampling. The results show that our MKFSS is effective for tackling the issues of multiple heterogeneous FeaSets and imbalanced data distribution for false positive reduction in lung CAD. It can demonstrate that solving both the issue of heterogeneous FeaSets and imbalanced data distribution substantially improve the performance of false positive reduction. Our MKFSS conducts a feature selection from both the FeaSets level and individual feature level can remove the irrelevant FeaSets and features, so as to facilitate the classification between true nodules and false nodules. It is also apparent that the proposed MKOS algorithm can improve the performance of the multiple kernel methods on the imbalanced nodule candidate data regardless of feature selection or not, which can show that MKOS has the ability to reduce the bias inherent in the learning procedure due to the class imbalance. Moreover, the combination of feature selection and over-sampling could result in a greater performance than any individual part. Furthermore, the interaction between feature selection and over-sampling are complex. The over-sampling after feature selection (MKFSS) leads to a better performance compared with the over-sampling before feature selection (MKFSS). The over-sampling on the irrelevant features indeed hinders the overall performance since over-sampling using these features cannot generate the accurate data samples.

5.4.2. Experiment II: The comparison with the state-of-the-art multiple kernel learning methods

In this experiment we evaluated the effectiveness of our proposed ℓ₁ norm regularized algorithm, and study how the behavior of multiple kernel learning is affected by the norm used to regularize the kernel weights for suspicious nodule classification. In order to demonstrate the effectiveness, we also implemented the ℓ₁ norm, ℓ₂ norm, weighted MKL proposed in [41] and uniformly weighted MKL for heterogeneous feature fusion, which commonly used multiple kernel learning methods. For both ℓ₁ norm and ℓ₂ norm, only the norm formulation in constraint is substituted by the \( \| D \| \leq 1 \) and \( \| D \| \leq 1 \), respectively.

In order to visualize the contribution of each feature type in different multi-kernel based fusion schemes, we plot the kernel weights of the base kernels for weighted MKL, ℓ₁ norm MKL, ℓ₂ norm MKL, and our ℓ₁ norm MKL in Fig. 9(a), (c), (b) and (d), respectively. From Fig. 9, we see that our ℓ₁ 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.

We compared the four multi-kernel based fusion schemes comprehensively with multiple data processing methods: none (without feature selection and over-sampling), FS (feature selection), OS (over-sampling) and FSOS (feature selection and over-sampling) in Table 3. All the processing methods are conducted after the fusion heterogeneous FeaSets. The ℓ₁ 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 ℓ₁ norm MKL and ℓ₂ norm MKL do not take the structure of feature subset into consideration, resulting in the prediction performance being dominated by our ℓ₁ norm MKL. The experimental results demonstrated that the ℓ₂ norm framework achieves the best. We obtained a more generalized performance for heterogeneous potential nodule data. Moreover, we listed the amount of FeaSets and features selected for the different types of MKL methods in Table 3. The amount of FeaSets selected can be automatically determined according to the threshold ratio, while the amount of feature selected is empirically set to 25.

5.4.3. Experiment III: The comparison between feature selection with two strategies in MKFSS

After feature fusion and selection from the feature subset level, there still exist many features and not all features are relevant to the discrimination of nodule candidates. Feature selection methods aim at dimensionality reduction and the removal of irrelevant features. To quantitatively evaluate the effect of the feature selection, we test the classification performance with respect to different numbers of the selected features. We compared the two different feature selection introduced in terms of G-mean and AUC. In addition, we implement the Recursive Feature Elimination (RFE-SVM) method, whose goal is to find a subset of features among the feature set, eliminating those features whose removal lead to the largest margin of class separation [52]. For the three different feature selection strategies, we compare them comprehensively on the imbalanced nodule dataset and balanced nodule dataset conducted by MKOS separately. We show the performance versus the number of selected features using different methods in Fig. 10(a) and (b). From Fig. 10(a) and (b), we can observe that the feature selection has improved the performance for fused heterogeneous FeaSets. The proposed MKFSS only consider the selection from the level of FeaSets. However, it trains a weight for each type of features, resulting in all features from the same subset are weighted equally. Additionally, the results from Fig. 10(a) and (b) can show that there still exist the irrelevant and redundant features in the fused FeaSets, which negatively affect the classification performance. Two different MKFS methods explore the feature level by calculating the predictive powers, so as to overcome the limitation of MKFSS and can allow our model perform simultaneous FeaSets-level and feature-level analysis. The incorporation of MKFS and MKFSS not only assigns proper weight to each type of FeaSet, but also rewards the relevances of the individual features inside a give feature type, so as to remove the irrelevant features and identify the discriminative features.

Moreover, from the results we can observe that the supervised feature selection algorithm Relief performs better than the unsupervised feature selection algorithms SPEC, as they use label information. Furthermore, we can see that both feature selection methods working on the balanced class distribution are better than the one working on the imbalanced one, which implies that both feature selection methods are affected by the unequal class distribution and benefit from the multi-kernel over sampling. Feature selection and classification of imbalanced data sets are two of the most interesting machine learning challenges, and feature selection also suffers from the skewed class distribution. Our two developed multi-kernel feature selection methods are based on the pairwise similarity measure and do not take into account the class distribution. The skewed class distribution could allow feature selection to bias to majority (negative) class and select the features favoring the majority class, resulting in an unexpected performance of minority (positive) class. The results tell us that the feature selection can improve the classification performance on the imbalanced nodule dataset. Furthermore, as confirmed by the results, we can conclude that the feature selection after over-sampling leads to a better performance compared to the feature selection alone without over-sampling from the experiment.

For SPEC and Relief, we can further identify the most discriminative features among the top 20 features in Table 4. Although we can find that different feature selection strategies generate different feature rankings, some important features can be identified, such as the statistics of SI, CV, GC and some geometric features.

5.4.4. Experiment IV: The influence of the over-sampling ratio parameter of MKOS

The optimal over-sampling ratio may be unknown, and the parameter of over-sampling ratio plays a vital role for the performance of imbalanced data learning. Many over-sampling methods in the literature over-sample the minority class into a completely balanced training set. However it is not an optimal way for generating synthetic
instances. It is desirable for an over-sampling method to be robust with respect to the different over-sampling ratio. In our research, we evaluated classifier performance using a variety of over-sampling ratios. We compared our methods MKFSS and MKOS with the popular over-sampling algorithm, SMOTE [47] and SMOTEBoost [57], and observe the influence of the re-sampling ratio in the different over-sampling on the classification performance. SMOTE [47] is an over-sampling method, whose main idea is to create new minority class examples by interpolating several minority

![Image](image1)

**Fig. 9.** Base kernel weights comparison of different MKL algorithms on the nodule dataset. Different colors indicate the base kernel functions with different kernel bandwidths ($2^{-1}$, $2^{-2}$,...,$2^{-9}$).

<table>
<thead>
<tr>
<th>Method</th>
<th>Data processing</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>G-mean (%)</th>
<th>AUC (%)</th>
<th>Size of selected FeaSets (features)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell_1$ None</td>
<td>69.41 ± 2.19*(0.007)</td>
<td>90.44 ± 1.85*(0.044)</td>
<td>78.22 ± 2.77*(0.015)</td>
<td>81.88 ± 2.16*(0.008)</td>
<td>2 (25)</td>
<td></td>
</tr>
<tr>
<td>FS</td>
<td>71.28 ± 3.25*(0.013)</td>
<td>92.11 ± 2.23*(0.081)</td>
<td>80.52 ± 3.45*(0.024)</td>
<td>83.43 ± 3.39*(0.011)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>74.62 ± 3.96*(0.021)</td>
<td>91.55 ± 3.55*(0.047)</td>
<td>82.42 ± 4.46*(0.026)</td>
<td>84.52 ± 3.74*(0.016)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSOS</td>
<td>77.28 ± 3.72*(0.024)</td>
<td>91.39 ± 2.11*(0.069)</td>
<td>84.04 ± 3.91*(0.029)</td>
<td>85.86 ± 2.60*(0.016)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\ell_2$ None</td>
<td>69.28 ± 1.19*(0.007)</td>
<td>90.44 ± 1.85*(0.044)</td>
<td>78.22 ± 2.77*(0.015)</td>
<td>81.88 ± 2.16*(0.008)</td>
<td>2 (25)</td>
<td></td>
</tr>
<tr>
<td>FS</td>
<td>71.28 ± 3.25*(0.013)</td>
<td>92.11 ± 2.23*(0.081)</td>
<td>80.52 ± 3.45*(0.024)</td>
<td>83.43 ± 3.39*(0.011)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>74.62 ± 3.96*(0.021)</td>
<td>91.55 ± 3.55*(0.047)</td>
<td>82.42 ± 4.46*(0.026)</td>
<td>84.52 ± 3.74*(0.016)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSOS</td>
<td>77.28 ± 3.72*(0.024)</td>
<td>91.39 ± 2.11*(0.069)</td>
<td>84.04 ± 3.91*(0.029)</td>
<td>85.86 ± 2.60*(0.016)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weighted None</td>
<td>69.28 ± 1.19*(0.007)</td>
<td>90.44 ± 1.85*(0.044)</td>
<td>78.22 ± 2.77*(0.015)</td>
<td>81.88 ± 2.16*(0.008)</td>
<td>2 (25)</td>
<td></td>
</tr>
<tr>
<td>FS</td>
<td>71.28 ± 3.25*(0.013)</td>
<td>92.11 ± 2.23*(0.081)</td>
<td>80.52 ± 3.45*(0.024)</td>
<td>83.43 ± 3.39*(0.011)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>74.62 ± 3.96*(0.021)</td>
<td>91.55 ± 3.55*(0.047)</td>
<td>82.42 ± 4.46*(0.026)</td>
<td>84.52 ± 3.74*(0.016)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSOS</td>
<td>77.28 ± 3.72*(0.024)</td>
<td>91.39 ± 2.11*(0.069)</td>
<td>84.04 ± 3.91*(0.029)</td>
<td>85.86 ± 2.60*(0.016)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\ell_{2,1}$ None</td>
<td>75.23 ± 1.59*(0.021)</td>
<td>91.33 ± 2.32*(0.037)</td>
<td>82.03 ± 1.99*(0.033)</td>
<td>87.23 ± 2.89*(0.038)</td>
<td>6 (25)</td>
<td></td>
</tr>
<tr>
<td>FS</td>
<td>79.23 ± 2.86*(0.028)</td>
<td>93.27 ± 2.88*(0.143)</td>
<td>85.57 ± 2.65*(0.037)</td>
<td>91.39 ± 3.37*(0.043)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>83.60 ± 3.82*(0.033)</td>
<td>91.55 ± 4.18*(0.066)</td>
<td>87.49 ± 3.89*(0.044)</td>
<td>90.16 ± 4.66*(0.037)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSOS</td>
<td>87.55 ± 3.12</td>
<td>92.41 ± 2.84</td>
<td>89.95 ± 3.66</td>
<td>94.17 ± 2.41</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3
The comparison of performance using multiple feature representation methods with different norm. (Note that the value in parenthesis indicate p-value and * stands for the case with $p \leq 0.05$.)
class instances that lie together. SMOTE creates instances by randomly selecting one (or more depending on the over-sampling ratio) of the $k$ nearest neighbors a minority class instance and the generation of the new instance values from a random interpolation of both instances. SMOTEBoost [57] is an over-sampling method based on a combination of the SMOTE algorithm with AdaBoost. It introduces synthetic instances more aggressively and decreases the weight of costly correct classifications more conservatively.

AdaCost [53]: AdaCost uses the cost of misclassifications to update the training distribution on successive boosting runs. Specially, AdaCost updates the instance weights of costly wrong classifications more aggressively and decreases the weight of costly correct classifications more conservatively.

Cost sensitive neural network (CSNN) [55]: CSNN uses the original training set to train a neural network, and the cost-sensitivity is introduced in the test phase by threshold-moving. Threshold-moving moves the output threshold toward positive class such that examples with higher costs become harder to be misclassified. It is obvious that threshold-moving is very different from sampling because the latter relies on the manipulation of the training data while the former relies on manipulating the outputs of the classifier.

The base classifier in AdaCost and MetaCost is SVM, and the ratio cost of them are set to the reverse of the size. All the sizes of ensemble methods are set to 50. In the setting of the neural network classifier of CSNN, the number of input neurons is equal to the number of features, and the number of neurons in the hidden layer is set to be 50. The sigmoid function is used as the activation function, and the inner training epochs is set to be 200 with a learning rate of 0.1.

AdaCost updates the instance weights of costly wrong classifi-
cations to update
in the training data.
threshold-moving is very di-
different from sampling because the latter
relies on the manipulation of the training data while the former relies on manipulating the outputs of the classifier.

Cost sensitive neural network (CSNN) [55]: CSNN uses the original training set to train a neural network, and the cost-sensitivity is introduced in the test phase by threshold-moving. Threshold-moving moves the output threshold toward positive class such that examples with higher costs become harder to be misclassified. It is obvious that threshold-moving is very different from sampling because the latter relies on the manipulation of the training data while the former relies on manipulating the outputs of the classifier.

The base classifier in AdaCost and MetaCost is SVM, and the ratio cost of them are set to the reverse of the size. All the sizes of ensemble methods are set to 50. In the setting of the neural network classifier of CSNN, the number of input neurons is equal to the number of features, and the number of neurons in the hidden layer is set to be 50. The sigmoid function is used as the activation function, and the inner training epochs is set to be 200 with a learning rate of 0.1.

AdaCost updates the instance weights of costly wrong classifi-
cations to update
in the training data.
threshold-moving is very di-
different from sampling because the latter
relies on the manipulation of the training data while the former relies on manipulating the outputs of the classifier.

Cost sensitive neural network (CSNN) [55]: CSNN uses the original training set to train a neural network, and the cost-sensitivity is introduced in the test phase by threshold-moving. Threshold-moving moves the output threshold toward positive class such that examples with higher costs become harder to be misclassified. It is obvious that threshold-moving is very different from sampling because the latter relies on the manipulation of the training data while the former relies on manipulating the outputs of the classifier.

The base classifier in AdaCost and MetaCost is SVM, and the ratio cost of them are set to the reverse of the size. All the sizes of ensemble methods are set to 50. In the setting of the neural network classifier of CSNN, the number of input neurons is equal to the number of features, and the number of neurons in the hidden layer is set to be 50. The sigmoid function is used as the activation function, and the inner training epochs is set to be 200 with a learning rate of 0.1.

AdaCost updates the instance weights of costly wrong classifi-
cations to update
in the training data.
threshold-moving is very di-
different from sampling because the latter
relies on the manipulation of the training data while the former relies on manipulating the outputs of the classifier.

Cost sensitive neural network (CSNN) [55]: CSNN uses the original training set to train a neural network, and the cost-sensitivity is introduced in the test phase by threshold-moving. Threshold-moving moves the output threshold toward positive class such that examples with higher costs become harder to be misclassified. It is obvious that threshold-moving is very different from sampling because the latter relies on the manipulation of the training data while the former relies on manipulating the outputs of the classifier.

The base classifier in AdaCost and MetaCost is SVM, and the ratio cost of them are set to the reverse of the size. All the sizes of ensemble methods are set to 50. In the setting of the neural network classifier of CSNN, the number of input neurons is equal to the number of features, and the number of neurons in the hidden layer is set to be 50. The sigmoid function is used as the activation function, and the inner training epochs is set to be 200 with a learning rate of 0.1.

AdaCost updates the instance weights of costly wrong classifi-
cations to update
in the training data.
threshold-moving is very di-
different from sampling because the latter
relies on the manipulation of the training data while the former relies on manipulating the outputs of the classifier.

Cost sensitive neural network (CSNN) [55]: CSNN uses the original training set to train a neural network, and the cost-sensitivity is introduced in the test phase by threshold-moving. Threshold-moving moves the output threshold toward positive class such that examples with higher costs become harder to be misclassified. It is obvious that threshold-moving is very different from sampling because the latter relies on the manipulation of the training data while the former relies on manipulating the outputs of the classifier.

The base classifier in AdaCost and MetaCost is SVM, and the ratio cost of them are set to the reverse of the size. All the sizes of ensemble methods are set to 50. In the setting of the neural network classifier of CSNN, the number of input neurons is equal to the number of features, and the number of neurons in the hidden layer is set to be 50. The sigmoid function is used as the activation function, and the inner training epochs is set to be 200 with a learning rate of 0.1.

AdaCost updates the instance weights of costly wrong classifi-
cations to update
in the training data.
threshold-moving is very di-
different from sampling because the latter
relies on the manipulation of the training data while the former relies on manipulating the outputs of the classifier.

Cost sensitive neural network (CSNN) [55]: CSNN uses the original training set to train a neural network, and the cost-sensitivity is introduced in the test phase by threshold-moving. Threshold-moving moves the output threshold toward positive class such that examples with higher costs become harder to be misclassified. It is obvious that threshold-moving is very different from sampling because the latter relies on the manipulation of the training data while the former relies on manipulating the outputs of the classifier.

The base classifier in AdaCost and MetaCost is SVM, and the ratio cost of them are set to the reverse of the size. All the sizes of ensemble methods are set to 50. In the setting of the neural network classifier of CSNN, the number of input neurons is equal to the number of features, and the number of neurons in the hidden layer is set to be 50. The sigmoid function is used as the activation function, and the inner training epochs is set to be 200 with a learning rate of 0.1.
and AUC value than the other contender methods. Tomek Link performs the least since it is hard to identify the noise when the distribution is complex and imbalanced. Some border points may also be removed as noise while they are useful for training, resulting in loss of information. SMOTE is the most common technique, and many extensions of SMOTE have been proposed. However, SMOTE helps in blindly broadening the decision region of the minority class without regard to the distribution of the majority class. This leads to over-generalization so as to inevitably decrease the accuracy of the majority class. SMOTEBoost can improve the basic SMOTE since it benefits from the diversity of the ensemble framework. These re-sampling methods only consider the class skew and properties of the dataset as a whole, while datasets often exhibit characteristics and properties at a local, rather than global level. Hence, it becomes important to analyze and consider the datasets in the reduced subsets (reduced features and instances). The two general cost sensitive learning methods (AdaCost and MetaCost) are better than Tomek Link, but perform slightly worse than the other advanced re-sampling methods. It may be because the ratio misclassification cost based on the size ratio between two classes is not appropriate, resulting in obtaining an unexpected performance. The misclassification cost is vital for cost sensitive learning, and needs to be searched by some heuristic methods. Moreover, MKFSS can improve the positive recognition ability so as to achieve excellent overall performance with G-mean and AUC.

5.4.6. Experiment VI: The comparison with the state-of-the-art methods for false positive reduction of Lung CAD

In this experiment, we compare our proposed method against some current methods for false positive reduction in lung nodule detection. The selected several state-of-the-art algorithms for comparison are:

Linear discriminant analysis (LDA): LDA is a linear based method, which is most frequently used in removing false positives [19].

Random forest (RF): RF is an ensemble of unpruned classification or regression trees, induced from bootstrap samples of the training data, using random feature selection in the tree induction process. RF is used to discriminative nodule and non-nodule patterns in [28]. The parameters of random forest are set as follows: the number of trees is set to 50, the number of features at each split is set to 5, and the prediction is made by majority voting.

Asymmetric AdaBoost (AsyAdaboost) [34]: AsyAdaboost is an iterative greedy algorithm to construct a strong classifier from a predefined set of weak ones. During learning, the two errors are penalized unequally. The asymmetry factor is set to 10 in the AsyAdaboost algorithm, and neural network is chosen as the base classifier.

Cost sensitive SVM (CS-SVM): CS-SVM is a good solution on the unbalanced data sets [26,46]. Radial Basis Function (RBF kernel) is a reasonable first choice for the classification of the nonlinear datasets, as it has fewer parameters ($\gamma$). Hence we choose RBF as the kernel function of CS-SVM in this experiment. We fix $C = C^+$ and $C = C \times C^f$, where $C$ and $C^f$ are respectively the regularization parameter and the ratio misclassification cost factor. The ranges for $C$ and $\gamma$ are $(2^{-5}, 2^5)$ and $(2^{-15}, 2^1)$, respectively. The ratio cost is usually suggested to be the ratio of the amounts of two classes. However, our experiments show that it is not always optimal. Therefore, the ratio cost is searched iteratively to get a best one, after obtaining the optimal intrinsic parameters of SVM. The range of ratio misclassification cost factor $C^f$ was empirically chosen between 1 and $100 \times N_{nur}/N_{pos}$. Hence, we optimize the best parameter pair ($C$ and $\gamma$) and the cost ratio parameter simultaneously [46]. However, it is not feasible to use a triple circulation for optimizing the best parameters, so we optimize the best parameter pair($C$ and $\gamma$) and the cost ratio parameter simultaneously [46]. G-mean is used to guide the search of the intrinsic parameters and ratio cost. All the experiments are conducted by 10-fold cross-validation.

Hybrid Probabilistic Sampling combined with Diverse Random Subspace (HPS-DRS) [33]: The hybrid probabilistic sampling (HPS) method adopts the combination of over-sampling and under-sampling,

Table 5

The comparison between our methods with other approaches for imbalanced data learning. (Note that the value in parenthesis indicate p-value and $*$ stands for the case with $p \leq 0.05$.)

<table>
<thead>
<tr>
<th>Method</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>G-mean (%)</th>
<th>AUC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMOTE</td>
<td>78.51 ± 5.44$^*$0.023</td>
<td>85.57 ± 4.77$^*$0.017</td>
<td>81.93 ± 4.85$^*$0.027</td>
<td>82.62 ± 4.27$^*$0.011</td>
</tr>
<tr>
<td>SMOTEBoost</td>
<td>82.13 ± 4.50$^*$0.028</td>
<td>87.93 ± 4.32$^*$0.039</td>
<td>84.95 ± 4.19$^*$0.033</td>
<td>86.30 ± 2.76$^*$0.021</td>
</tr>
<tr>
<td>CBO</td>
<td>81.92 ± 2.82$^*$0.024</td>
<td>88.22 ± 2.33$^*$0.039</td>
<td>81.90 ± 3.06$^*$0.031</td>
<td>83.17 ± 2.15$^*$0.014</td>
</tr>
<tr>
<td>Tomek link</td>
<td>72.35 ± 6.06$^*$0.007</td>
<td>85.24 ± 5.26$^*$0.032</td>
<td>78.70 ± 4.80$^*$0.021</td>
<td>80.78 ± 4.72$^*$0.010</td>
</tr>
<tr>
<td>ENN</td>
<td>71.67 ± 6.18$^*$0.005</td>
<td>84.73 ± 5.73$^*$0.029</td>
<td>77.89 ± 5.22$^*$0.024</td>
<td>79.27 ± 4.09$^*$0.012</td>
</tr>
<tr>
<td>CSNN</td>
<td>75.45 ± 4.50$^*$0.015</td>
<td>89.14 ± 4.32$^*$0.063</td>
<td>84.95 ± 4.19$^*$0.040</td>
<td>82.22 ± 2.76$^*$0.018</td>
</tr>
<tr>
<td>MetaCost</td>
<td>77.23 ± 2.82$^*$0.014</td>
<td>86.98 ± 2.33$^*$0.044</td>
<td>81.90 ± 3.06$^*$0.028</td>
<td>83.17 ± 2.15$^*$0.021</td>
</tr>
<tr>
<td>AdaCost</td>
<td>81.49 ± 1.89$^*$0.022</td>
<td>87.22 ± 2.26$^*$0.034</td>
<td>84.25 ± 4.07$^*$0.030</td>
<td>81.53 ± 2.04$^*$0.015</td>
</tr>
<tr>
<td>MKFSS</td>
<td>87.55 ± 3.12</td>
<td>92.41 ± 2.84</td>
<td>89.95 ± 3.66</td>
<td>94.17 ± 2.41</td>
</tr>
</tbody>
</table>
and incorporates probability function in its data distribution re-sampling mechanism. It generates more accurate instances to generalize the decision region for the nodule class, and removes the redundant instances for the non-nodule class without destroying the structure of the data. In addition, the proposed Diverse Random Subspace (DRS) ensemble framework can inject more diversity into the ensemble to acquire better classification performance and generalization capability. In the parameters setting of HPS-DRS, the re-sampling parameter is set to 70% and the parameters $K$ set to 5. Neural network is chosen as the base classifier.

**Cost sensitive Adaptive Random Subspace Method (CS-ARSM)** [36]: CS-ARSM is a cost sensitive adaptive random subspace ensemble algorithm by adjusting decision threshold for learning imbalanced potential nodule data. It can adaptively learn the optimal cost matrix in cost sensitive learning and the ensemble size in random subspace ensemble classifier. Neural network is chosen as the base classifier.

Experimental results (Table 6) demonstrate the benefit of MKFSS compared to other commonly used methods for false positive reduction in Lung nodule CAD. We can see that MKFSS outperforms other methods in terms of AUC and G-mean. LDA performs worst compared with other sophisticated classifiers or ensemble methods. That may be because high-dimensional features make LDA overfitting and the features extracted from nodule candidates are not generally linear. Although random forest is an ensemble classifier, it can also suffer from the curse of learning from an imbalanced training data set, resulting in lower G-mean. Although CS-SVM incorporates the cost sensitive learning and kernel parameter optimization, it performs worse than MKFSS due to that it does not consider the selection of FeaSets and features. At confidence level $\alpha = 0.05$, the results in Table 6 show that the proposed method significantly outperforms the other commonly used methods except HPS-DRS. We also showed the ROC analysis of the various approaches in Fig. 12.

To sum up, our empirical results are very encouraging and have demonstrated the promise of the MKFSS method in the false positive reduction. MKFSS combines the advantage of two techniques from two aspects of feature selection and over-sampling in the optimal kernel space, so obtain an improvement in prediction performance on the positive class (sensitivity) and overall improved G-mean as well as AUC. By the horizontal comparison, our methods have been shown to improve on the performance of reducing false positives while maintaining the high recognition of true nodules i.e. our methods do not jeopardize the high sensitivity of the initial detection.

### Table 6
The comparison between our methods with other approaches for false positive reduction.

<table>
<thead>
<tr>
<th>Metric</th>
<th>MKFSS</th>
<th>RF</th>
<th>AsyAdaboost</th>
<th>CS-SVM</th>
<th>LDA</th>
<th>CS-ARSM</th>
<th>HPS-DRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-mean</td>
<td>$89.95 \pm 3.16$</td>
<td>$78.24 \pm 4.39$</td>
<td>$79.25 \pm 5.21$</td>
<td>$77.23 \pm 6.46$</td>
<td>$75.64 \pm 6.27$</td>
<td>$86.16 \pm 2.77$</td>
<td>$84.69 \pm 4.77$</td>
</tr>
<tr>
<td>$p$-Value</td>
<td>-</td>
<td>0.036</td>
<td>0.049</td>
<td>0.035</td>
<td>0.032</td>
<td>0.062</td>
<td>0.058</td>
</tr>
<tr>
<td>AUC</td>
<td>$94.17 \pm 2.41$</td>
<td>$79.43 \pm 3.48$</td>
<td>$80.74 \pm 3.39$</td>
<td>$81.57 \pm 2.91$</td>
<td>$77.82 \pm 2.04$</td>
<td>$87.92 \pm 1.73$</td>
<td>$89.70 \pm 5.11$</td>
</tr>
<tr>
<td>$p$-Value</td>
<td>-</td>
<td>0.024</td>
<td>0.039</td>
<td>0.038</td>
<td>0.013</td>
<td>0.045</td>
<td>0.055</td>
</tr>
</tbody>
</table>

5.4.7. Experiment VII: The influence of the number of training samples and FeaSets

The purpose of this experiment is to examine how the performance of MKFSS changes with respect to the size of the training samples and FeaSets to the classification performance for previous three comparisons. Fig. 13 shows the average AUC results as a function of the number of training samples, with a 10-fold cross-validation strategy. Fig. 13 shows that with the increase in the number of samples in the training set, the performances of all the methods improve. Moreover, increasing the amounts of training data from the performance of MKFSS improves more significantly than other baselines. In fact, with only 30% of the training samples, MKFSS has already achieved comparable performance than the comparable methods with 50% of the training data. This also demonstrates that when the training data are limited, the fusion and selection for heterogeneous FeaSets enables MKFSS to overcome the small sample size problem. The small sample size makes it difficult to build a generalized model, and the high dimensional data could lead to the over-fitting issue.

Fig. 14 shows the performance variation w.r.t. the number of FeaSets in terms of AUC. We observe improved predictive performance of MKFSS as the number of FeaSets increases, which demonstrates that MKFSS can efficiently fuse heterogeneous feature subsets. MKFSS successfully handles discrepancies in the discriminative power of different features by assigning lower weights to less discriminative FeaSets, while other comparable methods employ the concatenation strategy with an equal confidence for each feature type. The effective fusion mechanism avoids the problem of normalizing the data when features are collected from different domains, which is why our method was able to handle heterogeneous FeaSets and to outperform standard MKL or other comparable methods in the experiments. However, when the number of FeaSets is too small, MKFSS is not competitive with the other competing methods using all features with respect to AUC, this may be due to that MKOS cannot perform well when the descriptors are not sufficient for representing and discriminating the nodule candidates.

5.4.8. Experiment VIII: The comparison with the learned features by deep learning methods

Deep learning has been one of the most prominent machine learning techniques nowadays, being the state-of-the-art on a broad range of applications where automatic feature extraction is needed. The objective of deep learning methods is to discover a set of discriminative features from the proposed hierarchical neural networks, so as to simplify conventional lung nodule malignancy suspiciousness classification by removing nodule segmentation and hand-crafted feature (e.g., texture and shape compactness) engineering work. We compare our multi-kernel based method with two deep learning methods, working on computation of deep features from volume data and fusion of heterogeneous hand-crafted feature, respectively. The aim of both deep learning methods is to learn high-level suspiciousness specific
features for lung nodule classification.

Convolutional neural networks (CNN) [64]: Recent work on deep learning has examined how deep networks can be trained to produce useful representations for lung nodules [65,66]. CNN is used to learn highly discriminative features for nodule detection in lieu of hand-engineered ones such as geometric shape or texture. We employ 3D CNN to examine how deep networks can be trained to produce useful representations for 3D suspicious nodules. The compared CNN was implemented based on CAFFE [59]. This comparison allows us to evaluate if the feature representations by deep learning can outperform the hand-crafted VOI-based features. We hierarchically constructed a deep 3D CNN model by stacking the C (convolutional layer), M (max-pooling layers) and FC (fully connected layers), as shown in Fig. 15. The last layers are fully connected ones which were followed by logarithmic loss to be minimized. The kernel size is $5 \times 5$ in all convolutional layers. To capture a majority of nodule morphology, the input nodule patch size was set to $32 \times 32 \times 16$ voxels, and all the training 3D volumes were flattened into a feature vector as its input. The ultimate output layer employed the softmax activation to yield the prediction probabilities. An $\ell_2$ norm weight decay was adopted to avoid overfitting during the training process.

Multimodal DBM (MDBM): A Deep Boltzmann Machine [60] is a network of symmetrically coupled stochastic binary units. It contains a set of visible units and a sequence of layers of hidden units. We adopt Multimodal DBM to model the joint distribution over the multiple heterogeneous FeaSets, so as to generate a fused representation from multiple heterogeneous FeaSets. The MDBM method can be viewed as a composition of unimodal undirected pathways. Each pathway of each FeaSet can be pretrained separately in a completely unsupervised fashion, which allows us to leverage a large supply of unlabeled data. Only one hidden layer is used since the input of each DBM is hand-crafted feature rather than the raw data. After each DBM model is trained, the fused representation of the data was extracted and the difference of each DBM trained from heterogeneous FeaSets is bridged by the hidden layer. The joint layer contains 500 hidden units. At last, the fused representation is feed to a separate logistic regression. The structure is shown in Fig. 16.

Our CNN and MDBM implementations were based on CAFFE [59] and Multimodal learning with Deep Boltzmann Machines proposed in [67]. Additionally, CNN and MDBM are combined with SMOTE to overcome the imbalanced data distribution. The difference is that SMOTE is applied prior to the procedure of deep learning for MDBM, while SMOTE is applied after the procedure of deep learning for CNN since SMOTE cannot work on the original raw pixel data with high dimensionality. Additionally, we also compared two methods incorporating CNN and MDBM with MKL, named MK-CNN and MK-MDBM. In both the hybrid methods, the learned generic features by pre-training by CNN and MDBM are inputted into a multi-kernel learning with $\ell_2$-norm and trained, in order to directly compare the representation obtained by different mechanisms based on the kernel classifier.

We conducted experiments for comparing our method with two deep learning methods under the following three settings, in order to study various aspects of MKFSS.

- Fully supervised learning.
Experimental results are reported in Table 7 where the best results are boldfaced. A first glance at the results shows that our MKFSS significantly outperforms all the other comparable deep learning methods except MK-MDBM on both metrics. Furthermore, the classification performance of the various approaches with varying sizes of training data are plotted in Fig. 17(a). These results reveal several interesting points:

(1) Obviously, for the different amounts of the training samples, MKFSS achieves better performance than deep learning methods, which confirms the capability of the proposed multi-kernel learning. It is worth noting that CNN does not provide very good performance, as the results in Table 7. The reason is that CNN requires a large amount of labeled training data, however the size of three dimensional data (VOI) from nodules is much less compared with two dimensional data (ROI) since one CT scan (patient) generally contains 1–2 nodules. The limited available training data degrades the performance of CNN, and leads to overfitting since the volume of interest from three dimensions introduce more parameters in the network. The provided training data are not sufficient for optimizing the parameters. Both of the above reasons avoid CNN to achieve best performance.

(2) Both of MKFSS and MDBM are developed for fusing multiple heterogeneous hand-crafted FeaSets. From Table 7, we find that MKFSS performs better than MDBM and MK-MDBM in terms of G-mean and AUC. The result also confirms that through the multi-kernel based FeaSets fusion, feature selection and over-sampling, MKFSS has obtained better feature representations for the nodule data.

(3) We also notice that with the learned feature representation by CNN, using fine tuning with logistic regression usually gets a better classification performance than MK-CNN without fine tuning, while the results of MDBM get an opposite observation, which indicates that the learned deep features at pre-training stage without fine-tuning for CNN cannot represent the nodule candidates well, while the fused representation of FeaSets learned by MDBM is more effective when combined with multi-kernel learning. That is may be because the input for MDBM is the high level features which are potentially discriminative, thus the need for fine tuning with label information is eliminated compared to CNN learned from the raw pixel. Another reason may be that the deep features fused by MDBM has a nonlinear property, which can be modeled well by kernel classifiers.

Many works have shown the strong representation and classification performance of deep learning compared with the hand-crafted features, which is not completely fair for the hand-crafted features since less work extracted sufficient and complementary features from multiple views. Even a more complete representation of nodules is provided, the simple concatenation of features is applied without being fused in a principled way in prior work [23,33,50]. Moreover, many

---

**Table 7**
The comparison between our methods with other approaches for false positive reduction.

<table>
<thead>
<tr>
<th>Metric</th>
<th>MKFSS</th>
<th>CNN</th>
<th>MK-CNN</th>
<th>MDBM</th>
<th>MK-MDBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-mean</td>
<td>89.95 ± 3.16</td>
<td>81.35 ± 5.42</td>
<td>77.94 ± 5.39</td>
<td>85.90 ± 4.87</td>
<td>88.12 ± 4.34</td>
</tr>
<tr>
<td>p-Value</td>
<td>0.013</td>
<td>0.005</td>
<td>0.039</td>
<td>0.082</td>
<td>0.082</td>
</tr>
<tr>
<td>AUC</td>
<td>94.17 ± 2.41</td>
<td>84.15 ± 5.60</td>
<td>79.68 ± 4.77</td>
<td>88.51 ± 4.22</td>
<td>90.84 ± 3.78</td>
</tr>
<tr>
<td>p-Value</td>
<td>0.019</td>
<td>0.011</td>
<td>0.040</td>
<td>0.076</td>
<td>0.076.</td>
</tr>
</tbody>
</table>

---

**Fig. 15.** The structure of 3D-CNN.

**Fig. 16.** The structure of MDBM.
previous works do not take into account the issues of feature selection and imbalanced data learning at the same time for the false positive reduction. The previous results in this work imply that the promising advantage of joint feature selection and over-sampling for false positive reduction. However, both of them are generally often ignored. Another benefit of MKFSS is that it uses less computation time of the optimization of kernel parameters, which makes it more efficient than existing deep learning methods. Training a deep CNN requires extensive computational and memory resources compared with the traditional extraction of hand-crafted features and classification. Table 8 shows the training time of the three methods. Compared with the deep features automatically learned by the trained CNN from the original image, the traditional feature extraction has less computation complexity. Furthermore, the deep features learned by CNN has less interpretability, which is critical for medical domain. Therefore, our MKFSS combined with heterogeneous hand-crafted FeaSets has advantages with respect to the effectiveness and efficiency for the classification of nodule candidates.

- Data augmentation.

Deep learning networks need to be trained on a huge number of training images to achieve satisfactory performance. The reason that two deep learning methods underperformed MKFSS may be due to the insufficient training samples. In order to verify this hypothesis and compare them on more training data, we augmented the training data by generating synthetic samples and extensively validate the comparability of MKFSS and two deep learning algorithms.

Table 8 shows the training time of the three methods. Compared with the deep features automatically learned by the trained CNN from the original image, the traditional feature extraction has less computation complexity. Furthermore, the deep features learned by CNN has less interpretability, which is critical for medical domain. Therefore, our MKFSS combined with heterogeneous hand-crafted FeaSets has advantages with respect to the effectiveness and efficiency for the classification of nodule candidates.

- Semi-supervised learning.

In order to make the CAD systems perform well, a large amount of labeled instances are required for constructing a classifier model. However, the size of the labeled instances are usually insufficient since the task of providing label for each instance is quite time-consuming [58]. An effective strategy is needed to learn a model from a small amount of labeled instances combined with a large amount of unlabeled instances to enhance the performance of the learned model.

We here experimentally evaluated the performance of the comparable methods for semi-supervised learning. Here we target at studying the ability of the proposed MKL method and deep learning methods in exploiting unlabeled data to learn complicated data distributions, with rather limited supervision information. The comparable deep learning methods and our MKL cannot directly perform semi-supervised learning. We extend these methods to the semi-supervised learning by incorporating an embedding-based regularizer proposed in [61,62], which aims to enhance the performance of the classification model by using the labeled and unlabeled data together. MK-CNN and MK-MDBM perform a greedy layer-wise pre-training of weights using unlabeled data alone followed by a supervised MKL classifier.

In solving such problems, the given data set is often partially labeled, and one is required to make use of both the class information of the labeled data and the intrinsic relationships of the unlabeled ones to accomplish the learning tasks. Assume that we are given L labeled data and U unlabeled data. Incorporating the manifold regularization, we extend LapSVM [61] to multiple kernel circumstance, we obtain the following formulation:

\[
\min f \sum_{i=1}^L \ell(f(x_i), y_i) + C_1 \|f\|_H^2 + C_2 R_f(f) + C_3 R_u(d)
\]

where \( f = (f(x_1), \ldots, f(x_N)) \). \( N = L + U \). The first term represents the empirical loss on the labeled instance, the second term represents norm of \( f \) in \( H \) which aims at controlling the complexity of \( f \) in \( H \) for regularization purpose. The third term is a graph-based regularizer to preserve the similarities between samples in the whole \( L + U \) dataset. According to [61], we can rewrite the objective function as:

\[
\min_{a,d} \sum_{i=1}^L \xi_i + \frac{C_1}{2} a^T Ka + \frac{C_2}{2} a^T KL Ka + C_3 \|d\|_1
\]
we investigated and presented a multiple kernel learning framework for comparable methods with different amounts of labeled data. We varied the size of labeled and unlabeled subsets, such that the ratio of #unlabeled to #labeled images increases from 1, 5, 10 to 50.

The observation that MKFSS consistently outperforms both deep learning methods with varying numbers of external unlabeled data, and it has a good trend to give more accurate results when the number of labeled instances increases compared with other methods in Fig. 18. These results clearly demonstrate the power of MKFSS in adapting model to data complexity and providing better generalization performance.

To sum up, our empirical results have demonstrated the promise of MKFSS in the application of false positive reduction in lung nodule detection. The deep learning methods were no better than those achieved using the hand crafted features for classifying the nodule candidates, but it depends on how to properly fuse the complementary FeaSets, select the discriminative features from the heterogeneous feature subset, and how to effectively solve the imbalanced data distribution for the traditional approaches with successive steps: feature extraction and classification. The results suggested that machine learning methods of false positive reduction in three dimensional lung nodule detection should still employ the traditional strategy of feature extraction combined with classification due to its advantage in effectiveness and efficiency. Moreover, the performance of our MKFSS for discriminating benign and malignant nodules is a subject for future investigations. We also would like to verify if our findings about learned and hand-crafted features hold in other CAD system.

6. Conclusion

Focusing on improving the performance of false positive reduction, we investigated and presented a multiple kernel learning framework for lung nodule classification. Specifically, we have proposed a kernel method for integrating heterogeneous FeaSets for false positive reduction. We further extend the kernel framework for selecting features and over-sampling from the fused heterogeneous FeaSets. Through theoretical justifications and empirical studies, we demonstrated the effectiveness of the method MKFSS on the performance of reducing false positives vertically and horizontally. The proposed methods could be applied for the detection of many other potential lesions, such as mass and polyp. In the future, we will extend our framework to the work of diagnosis [6].

Acknowledgments

This research was supported by the National Science Foundation for Distinguished Young Scholars of China under Grant (71325002), the National Key Technology Research and Development Program of the Ministry of Science and Technology of China under grant No. (2014BAI17B01), the National Natural Science Foundation of China (615102091), the Fundamental Research Funds for the Central Universities (N140403004, N140402003), and the Postdoctoral Science Foundation of China (2015M570254).

References

P. Cao et al.

Pattern Recognition 64 (2017) 327–346


Peng Cao is a postdoctoral fellow at Northeastern University, China. He earned his Ph.D. degree in computer application in 2014 at Northeastern University, China. His research interests include imbalanced data learning, data mining, and medical data mining.

Xiaoli Liu is a Ph.D. candidate in Key Laboratory of Medical Image Computing of Ministry of Education, Northeastern University, China.

Jinzhu Yang is an associate professor at Northeastern University, China. His research interests include image processing and image segmentation.

Dazhe Zhao is a professor in the Department of Computer Science and Engineering, Northeastern University, China. She is the chairperson of the chair of Computer Program of Ministry of Education, Northeast China, University. She is also a senior member of China Computer Association. She is interested in software engineering, and medical image computing.

Wei Li is a research fellow at Key Laboratory of Medical Image Computing of Ministry of Education, China. His current research interests include medical big data processing and analysis.

Min Huang is a professor in the Department of Information Science and Engineering, Northeastern University, China.

Osmar Zaiane is a Professor in Computing Science at the University of Alberta, Canada, and Scientific Director of the Alberta Innovates Centre for Machine Learning. He is Associate Editor of many International Journals on data mining and data analytics and serves as a program chair and general chair for scores of international conferences in the field of data discovery and data mining. His current research interests include data mining, healthcare informatics.