Sparse learning and hybrid probabilistic oversampling for Alzheimer's Disease diagnosis

Peng Cao[1]*, Xiaoli Liu[1], Dazhe Zhao[1], and Osmar Zaiane[2]

[1] College of Computer Science and Engineering, Key Laboratory of Medical Image Computing of Ministry of Education, Northeastern University, Shenyang, China
[2] Computing Science, University of Alberta, Canada

Abstract. Alzheimers Disease (AD) is the most common neurodegenerative disorder associated with aging. Early diagnosis of AD is key to the development, assessment, and monitoring of new treatments for AD. Machine learning approaches are increasingly being applied on the diagnosis of AD from structural MRI. However, the high feature-dimension and imbalanced data learning problem is two major challenges in the study of computer aided AD diagnosis. To circumvent this problem, we propose a novel formulation with hinge loss and sparse group lasso to select the discriminative features since features exhibit certain intrinsic group structures, then we propose a hybrid probabilistic oversampling to alleviate the class imbalanced distribution. Extensive experiments were conducted to compare this method against the baseline and the state-of-the-art methods, and the results illustrated that this proposed method is more effective for diagnosis of AD compared to commonly used techniques.

Keywords: Alzheimer's disease, Group lasso, classification, imbalanced data

1 Introduction

Alzheimers disease (AD) is the most frequent form of dementia in elderly patients, which causes progressive impairment of memory and other cognitive functions, leading directly to death. It accounts for 6070% of age related dementia, affecting an estimated 30 million individuals in 2011 and the number is projected to be over 114 million by 2050[1]. Early diagnosis of AD patients is important because it allows early treatment to improve the quality of life of the patients and their families. Therefore, effective and accurate diagnosis of AD, as well as its prodromal stage (mild cognitive impairment (MCI)), has attracted more and more attention recently. The machine learning, or classification approach has been used to provide markers for various neurological disorders including Alzheimer's disease[2, 3] on structural magnetic resonance imaging (MRI). However, two of the key challenges in designing good prediction models for diagnosis of AD with MRI lie in:

1) How to solve the dimensionality reduction in the high-dimensional data: High dimensionality of the data may affect the computational performance (processing time) and, worse, it may lead to a wrong estimation and identification of the relevant predictors[2]. Feature selection methods select a small subset of the original feature set to

* Supported in part by National Natural Science Foundation of China (61502091).

reduce the dimensionality of the data set and facilitate better generalization of training samples. In AD, features exhibit certain intrinsic group structures in the context of Alzheimer's disease diagnosis. There is natural grouping of the features, the groups correspond to specific region-of-interest (ROIs) in the brain, and the individual features (average cortical thickness, standard deviation of thickness, surface area, cortical volume and subcortical volume) are specific properties of those regions. Hence the group effect in the features need to be taken into account while doing feature selection.

2) How to address the imbalanced data distribution between different classes: In Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset, the MCI cases eligible for the study are nearly two times the AD patients and normal controls (NC)[4]. This is a typical class imbalance problem [5, 6]. Class imbalanced data has detrimental effects on the performance of conventional classifiers. The actual cause for the poor performance of conventional classifiers on the minority class is not necessarily related to only the between-class imbalance. Within-class imbalance [7] refers to the case where a class is formed of a number of sub-clusters with different sizes, concerns itself with the distribution of representative data for sub-concepts within a class. The existence of sub-concepts also increases the complexity of the problem because the amount of instances among them is not usually balanced. For the patients with AD or MCI, there exists multiple sub-concepts as the disease involves multiple different subtype or different characteristic, which results in the distribution of instances over each class concepts and may yield clusters with unequal sizes.

The aim of our work is to simultaneously address both issues, and to explore whether oversampling combined with feature selection benefits the diagnosis of individuals based on multivariate brain MRI data. A low-dimensional representation of the data not only reduces the risk of overfitting, improving the models generalization ability, but also allow the oversampling algorithm to generate much accuracy synthetic instances [15]. Based on the motivation, we propose a new formulation incorporating hinge loss combined with sparse group lasso (called HLSGL) to conduct the feature selection. The objective function is non-convex and non-smooth, which is difficult to solve in general. To address it, we use differentiable approximation of hinge loss, and adapt a accelerated proximal gradient method for solving the non-smooth formulation[9]. On the other hand, we propose a hybrid probabilistic oversampling (HPS) algorithm to balance the skew class distribution based on the lower dimensional data. Experimental results show that HLSGL-HPS achieves a considerable improvement in the prediction performance. Results also demonstrate that feature selection and oversampling are two key steps for diagnosis of AD before building classifier model, and feature selection can be very helpful when facing imbalanced data sets.

2 Proposed Method

In this section, we describe the proposed approach in diagnosis of AD with structural MRI brain images. At first, feature selection based on HLSGL is performed to select relevant ROIs and features, then HPS is conducted to balance the skew class distribution on the selected feature subset. Finally, we build a kernel classifier trained on lower dimensional balanced data to classify AD/NC, AD/MCI and MCI/NC. HLSGL not only reduces the dimensionality so as to improve the performance of the HPS algorithm, but also makes the classification model interpretable. Moreover, HPS offers an effective solution for within-class in tandem with the between-class imbalance according to the distribution probability without jeopardizing structure of data.

2.1 Feature Selection by hinge loss combined with sparse group lasso, HLSGL

The general goal of supervised learning is to predict for the input $\boldsymbol{x}$ an output y. To achieve this objective, learning algorithms usually use training data $\{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$ to learn a prediction function f that can correlate $\boldsymbol{x}$ with y. A common approach to obtain is to minimize the following regularized empirical error:

$$\min_{\boldsymbol{w}\in\mathbb{R}^P} l(\boldsymbol{w}; \boldsymbol{y}, \boldsymbol{X}) + \lambda r(\boldsymbol{w}) \tag{1}$$

where $l(\cdot)$ denotes the loss function, $r(\cdot)$ is the regularizer and λ is regularization parameter. In the context of classification, we employ hinge loss $l(\boldsymbol{w}) = max(1-\boldsymbol{y}\boldsymbol{w}^T\boldsymbol{X}, 0)$, because kernel methods [12, 13] have been shown to be very effective for classification in Alzheimers disease[10, 11], and hinge loss is used in the objective function of kernel methods. However, hinge loss is non-smooth loss function, which is difficult to optimize. Therefore, we propose to use a differentiable approximation of hinge loss, which is defined as:

$$l(y,t) = \begin{cases} 0 \quad if \quad yt > 1+h \\ \dfrac{(1+h-yt)^2}{4h} \quad if \quad |1-yt| \leq h \\ 1-yt \quad if \quad yt < 1-h \end{cases} \tag{2}$$

where $t = \boldsymbol{w}^T\boldsymbol{X}$ is the predicted label, and h is a parameter to choose, typically between 0.01 and 0.5.

Sparse methods have attracted a great amount of research efforts in the past decade due to its sparsity-inducing property and strong theoretical guarantees[14]. Traditionally the ℓ_1-norm (namely lasso) was effectively implemented to feature selection in high-dimensional setting. However, lasso fails to capture the correlation information among the pairwise of group features. For group of features that the pairwise correlations among them are very high, lasso tends to select only one of the pairwise correlated features and does not induce the group effect. In the context of AD, the groups correspond to specific regions of-interest (ROIs) in the brain, and the individual shape features are specific properties of those regions. The multiple shape measures from the same ROI tend to be selected together as joint predictors, and use this prior knowledge to group relevant shape features together in the same ROI guide the learning process. Let $G = \{G_1, \ldots, G_m\}$ be the groups of variables at the m ROIs considered. For our data, the number of features in each group is 4 for cortical region involving cortical thickness average (TA), standard deviation of thickness (TS), surface area (SA) cortical volume (CV), and 1 for subcortical region involving subcortical volume (SV).

The group lasso is a technique to do variable selection on (predefined) groups of variables[8, 16]. However, it is a strict assumption for feature selection. Our motivation in promoting structured sparsity is drawn from the fact that for predicting disease status, if a particular brain region (ROI) is irrelevant, then coefficients of all morphological features at that ROI should be zero. Furthermore, if a particular brain region is deemed relevant, then we should be able to select the most important morphological feature(s) at that region to be considered for prediction. In order to achieve the goal of structural sparse feature selection (SGL), we employ sparse group lasso integrating lasso $\|\boldsymbol{w}\|_1$ and group lasso $\|\boldsymbol{w}\|_{G_{2,1}}$ as feature selection with structural grouping sparsity, so as to allow simultaneous joint feature selection from feature level and ROI level. The hierarchical norm of SGL is defined as:

$$r(\boldsymbol{w}) = \lambda_1 \|\boldsymbol{w}\|_1 + \lambda_2 \|\boldsymbol{w}\|_{G_{2,1}} \tag{3}$$

where $\|\boldsymbol{w}\|_1 = \sum_{i=1}^{p} \|\boldsymbol{w}_i\|_2$, and $\|\boldsymbol{w}\|_{G_{2,1}} = \sum_{j=1}^{m} \sqrt{\sum_{i \in G_j} \left\|\boldsymbol{U}_{G_j}\right\|_2^2}$. Note that p and m is the size of features and ROIs size, respectively.

By approximating hinge loss, $l(\boldsymbol{w}; \boldsymbol{y}, \boldsymbol{x})$ is convex and differentiable with Lipschitz continuous gradient so that $\|f(\boldsymbol{z}) - f(\boldsymbol{w})\| \leq L\|\boldsymbol{z} - \boldsymbol{w}\|$ where L denotes the Lipschitz constant, however $r(\boldsymbol{w})$ is still convex but non-smooth. For the convex but non-smooth formulation, we solve it by designing a new accelerated proximal gradient (APG) method [9] in this work.

A well studied idea in efficient optimization of such composite objective function is to start with a quadratic approximation of the form: $Q_L(\boldsymbol{w}, \boldsymbol{w}^{(t)}) := l(\boldsymbol{w}^{(t)}) + \langle \boldsymbol{w} - \boldsymbol{w}^{(t)}, \nabla l(\boldsymbol{w}^{(t)})\rangle + \frac{L}{2}\|\boldsymbol{w} - \boldsymbol{w}^{(t)}\|^2 + \lambda r(\boldsymbol{w})$. Ignoring constant terms in $\boldsymbol{w}^{(t)}$, the unique minimizer of the above expression can be written as $\pi_L^{l,r}(\boldsymbol{w}^{(t)}) = \arg\min_{\boldsymbol{w}} \{r(\boldsymbol{w}) + \frac{L}{2}\|\boldsymbol{w} - \left(\boldsymbol{w}^{(t)} - \frac{1}{L}\nabla l(\boldsymbol{w}^{(t)})\right)\|^2$, which can be viewed as a proximal operator corresponding to the non-smooth function $r(\boldsymbol{w})$. A popular approach to solving the non-smooth problems is to simply do the following iterative update: $\boldsymbol{w}^{(t+1)} = \pi_L^{l,r}(\boldsymbol{w}^{(t)})$, which can be shown to have a $O(1/t)$ rate of convergence [19]. In practice, since the Lipschitz constant L may be unknown, we follow the adaptive strategy suggested in [9] to make sure we make progress.

For our purposes, we consider a refined version of the iterative algorithm inspired by Nesterov's accelerated gradient descent [19]. The main idea, as studied in the literature as APG algorithms [9], is to iteratively consider the proximal operator $\pi_L^{l,r}(\cdot)$ at a specific linear combination of the previous two iterates $\{\boldsymbol{w}^{(t)}, \boldsymbol{w}^{(t-1)}\}$, in particular at $\hat{\boldsymbol{w}}^{(t)} = \boldsymbol{w}^{(t)} + \alpha_{t+1}(\boldsymbol{w}^{(t)} - \boldsymbol{w}^{(t-1)})$ instead of at just the previous iterate w_t. The choice of α_{t+1} follows Nesterov's accelerated gradient descent [19] and is detailed in Algorithm 1. The iterative algorithm simply updates $\boldsymbol{w}^{(t+1)} = \pi_L^{l,r}(\hat{\boldsymbol{w}}^{(t)})$

As shown in [9], the algorithm has a rate of convergence of $O(1/t^2)$. With $\tilde{\boldsymbol{w}}^{(t+1)} = (\hat{\boldsymbol{w}}^{(t)} - \frac{1}{L}\nabla l(\hat{\boldsymbol{w}}^{(t)}))$, the problem of computing the proximal operator $\pi_L^{l,r}(\hat{\boldsymbol{w}}^{(t)}) := T_L^{\lambda_1,\lambda_2}(\tilde{\boldsymbol{w}}^{(t)})$ is given by $T_L^{\lambda_1,\lambda_2}(\tilde{\boldsymbol{w}}^{(t)}) = \arg\min_{\boldsymbol{w} \in \mathbb{R}^p} \{\lambda_1 \|\boldsymbol{w}\|_1 + \lambda_2 \|\boldsymbol{w}\|_{G_{2,1}} + \frac{L}{2}\|\boldsymbol{w} - \tilde{\boldsymbol{w}}^{(t)}\|^2\}$.

The proximal operator $T_L^{\lambda_1,\lambda_2}(\tilde{\boldsymbol{w}}^{(t)})$ can be computed efficiently in two steps, as outlined below:

$$\tilde{\boldsymbol{s}}^{(t)} = T_L^{\lambda_1}(\tilde{\boldsymbol{w}}^{(t)}) \ , \tag{4}$$

$$\boldsymbol{w}^{(t+1)} = T_L^{\lambda_2}(\tilde{\boldsymbol{s}}^{(t)}) = T_L^{\lambda_1,\lambda_2}(\tilde{\boldsymbol{w}}^{(t)}) \ . \tag{5}$$

$T_L^{\lambda_1}(\tilde{\boldsymbol{w}}^{(t)})$ can be obtained by soft-thresholding directly $\tilde{\boldsymbol{s}}^{(t)} = \mathrm{sgn}(\tilde{\boldsymbol{w}}^{(t)}) \max\{|\tilde{\boldsymbol{w}}^{(t)}| - \lambda_1, 0\}$. Following [17], the group lasso updates can be done by:

$$\boldsymbol{w}_j^{(t+1)} = T_L^{\lambda_2}(\tilde{\boldsymbol{s}}^{(t)}) = \frac{\max\{\|\tilde{\boldsymbol{s}}_j^{(t)}\|_F - \frac{\lambda_2}{L}, 0\}}{\|\tilde{\boldsymbol{s}}_j^{(t)}\|_F} \tilde{\boldsymbol{s}}_j^{(t)} \ . \tag{6}$$

where $\boldsymbol{w}_j^{(t+1)}$, $\boldsymbol{s}_j^{(t+1)}$ are group specific sub-vector correspond to group j in $\boldsymbol{w}^{(t+1)}$, $\boldsymbol{s}^{(t+1)}$ respectively.

2.2 Hybrid Probabilistic Sampling

In order to solve the the between-class and within-class imbalance simultaneously, we propose a hybrid probabilistic sampling (HPS) with the combination of over-sampling and under-sampling, and incorporates probability function in its data distribution re-sampling mechanism. It generates more accurate instances to generalize the decision region for the minority class, and removes the redundant instances for the majority class without destroying the structure of the data. It can deal with the between-class imbalance and within-class imbalance issues simultaneously.

Gaussian Mixture Models (GMM) are generative probabilistic models of several Gaussian distributions for density estimation in machine learning applications. A Gaussian mixture can be constructed to acceptably approximate any given density. Therefore, we assume the distribution of two classes follows the Gaussian mixture model with unknown parameters. The parametric probability density function of GMM is defined as a weighted sum of Gaussians. The finite Gaussian mixture model with k components may be written as: $p(y|\mu_1, \ldots, \mu_k; \sigma_1, \ldots, \sigma_k; \pi_1, \ldots, \pi_k) = \sum_{j=1}^{k} \pi_j N(\mu_j, \sigma_j)$, and $0 \leq \pi_j \leq 1, \sum_{j=1}^{k} \pi_j = 1$, where μ_j are the means, σ_j are covariance matrixes, π_j are the mixing proportions, and $N(\mu_j, \sigma_j)$ is a Gaussian with specified mean and variance.

We estimate the parameters of GMM with FJ algorithm [18], which can overcome the major weaknesses of the basic EM algorithm particularly vis-à-vis the initialization, and can automatically select the number of component, we use it here to estimate the parameters of GMM. Each instance $\boldsymbol{x}_i$ will then be assigned to the cluster k where it has the largest posterior probability $p(k|\boldsymbol{x}_i)$. When calculating the probability of each instance on each component, the probabilities for the feature is obtained by a Gaussian density function. At the same time, we obtain the parameters of each Gaussian component. For different clusters, the re-sampling rates are different; within the cluster, the probabilities of each instance to be chosen for re-sampled are different.

We use the oversampling combined with undersampling to balance the class size. The sizes of the two classes are N^+ and N^-. The gap N_G between two uneven classes is: $N_G = N^- - N^+$. Thus, the amount of instances in the minority class for oversampling is: $N_\alpha^+ = N_G \times \alpha$, and the amount in the majority class for undersampling is: $N_\alpha^- = N_G \times (1 - \alpha)$. To adjust the within class imbalance, we need to balance cluster sizes in each class. For the minority class, the number of instances to be oversampled is inversely proportional to the size of the cluster; for the majority class, the numbers of instances to be undersampled are proportional to the size of the cluster. Furthermore, we use the probabilities of each instance to conduct the re-sampling with maintaining the data structure, in order to address the two type imbalance issues. In the clusters of the majority class, the instances with higher probability are dense, they are frequent in the subclass, and hence they have higher chance to be undersampled. We choose the instances to be undersampled according to the Gaussian distribution. In the clusters of the minority class, the new instances are produced according to the probability function of Gaussian distribution, resulting in finding more potentially interesting regions. The main steps in undersampling for the clusters of the majority class and oversampling for the clusters of the minority class are conducted as following:

Step 1: In the oversampling for the minority class, the smaller the size of cluster within the class, the more instances are over-sampled, so as to avoid the small disjuncts. For the $i\text{-}th$ cluster, the amount of synthetic instances needed to be generated is: $N_\alpha^{i+} = (\frac{1}{size_+^i} / \sum_{j=1}^{S_+} \frac{1}{size_+^j}) \times N_\alpha^+$, where $size_+^i$ is the size of $i\text{-}th$ cluster, S_+ is the number of clusters in the minority class.

Step 2: In the $i\text{-}th$ cluster, N_α^{i+} instances are generated with the parameters from the current Gaussian distribution. The new instances are generated according to the probability function of the Gaussian distribution with parameters learned from the available data. Firstly, the probability from the Gaussian distribution of each instance is calculated and normalized: $\hat{p_k} = p_k / \sum_{j=1}^{size_+^i} p_j$. Then, the amount of new instances for each instance x_k is obtained according to: $n_k = N_\alpha^{i+} \times \hat{p_k}$. For ensuring that synthetic instances created via this method always lay in the region near $\boldsymbol{x}_k$, the n_k instances are generated in its $K(K = 5)$ nearest neighbors region. It can extend more potential regions rather than being limited along the line between the minority example and its selected nearest neighbors. In addition, this guarantees the creation of minority samples in the cluster, and avoids any incorrect synthetic instance generation.

Step 3: In the undersampling for the majority class, we calculate the amount of instances to be under-sampled for each cluster. The number of instances to be undersampled are proportional to the size of clusters. For the $i\text{-}th$ cluster, the amount of instances needed to be removed is: $N_\alpha^{i-} = (size_-^i / \sum_{j=1}^{S_-} size_-^j) \times N_\alpha^-$, where $size_-^i$ is the size of $i\text{-}th$ cluster, S_- is the number of clusters in the majority class.

Step 4: In each component Gaussian distribution, the center region is denser than the border region. These instances from the center are more possible to be redundant, and so are better candidates to be under-sampled. We need to choose the instances to be ignored or removed located on the center of the distribution more than the border. The probabilities to be chosen for undersampling are proportional to the normalized probability $\hat{p}$ of the Gaussian distribution for each instance in a cluster.

3 Experimental study

3.1 Dataset and Setting

The data used in this paper were obtained from the ADNI database (adni.loni.usc.edu) [21]. The MRI features used in our experiments are based on the imaging data from the ADNI database processed by a team from UCSF (University of California at San Francisco), who performed cortical reconstruction and volumetric segmentations with the FreeSurfer image analysis suite (http://surfer.nmr.mgh.harvard.edu/). There were $p = 319$ MRI features (covariates) in total. In this work, only ADNI1 subjects with no missing feature and cognitive outcome information baseline data are included. This yields a total of $n = 788$ subjects, who are categorized into 3 baseline diagnostic groups: Cognitively Normal (CN, $n_1 = 225$), Mild Cognitive Impairment (MCI, $n_2 = 390$), and Alzheimer's Disease (AD, $n_3 = 173$).

For the HLSGL, the regularization parameters λ_1, λ_2 in Eq. (1) and C in SVM are chosen by nested cross-validation strategy on the training data (trying values 0.01, 0.1, 1, 10, 100,1000). In HPS, α is set to 70%. To evaluate the performance of different classification methods, we use a 10-fold cross-validation strategy to compute the sensitivity, the specificity, G-mean and AUC.

3.2 Effectiveness of proposed kernel framework

To evaluate and compare the performances of each method, three classification experiments were performed: AD vs CN, AD vs MCI and MCI vs NC. We compare our method (HLSGL-HPS) with five baseline methods involving SVM working on the original dataset (ALL), hinge loss combined lasso (HLL), HPS on the original dataset(All-HPS) and HLL combined with HPS(HLL-HPS). In the experiment of CN vs AD, the two class are nearly balanced, therefore oversampling is not need to be conducted. Prediction performance results of three groups, measured by five measure metrics of HLSGL-HPS and baseline methods are shown in Tables 1. The results demonstrate that reducing the feature dimensionality can mitigate the overfitting problem and improve a models generalizability. Moreover, sparse group lasso based feature selection can achieve robust classification performance compared with lasso. Furthermore, the results indicate that feature selection is as important as the oversampling in the imbalanced data classification. Oversampling on the high dimensional irrelevant feature leads to the creation of wrong instances when the class dispersion or the class noise exists.

Moreover, we provides an empirical evaluation of the proposed method compared with other classification methods commonly used in diagnosis of AD, such as ELM(Extreme learning machine) [3], logistic regression [22] and random forest [23]with respect to G-mean and AUC in Table 2. From Table 2 and Fig. 1 we observe that our proposed framework is always better than the other three classification methods, even significantly outperform them in the most cases.

We applied HLSGL to conduct the ROI selection. In each fold, the ROIs are ranked based on the weight value corresponding to this ROI by frobenius norm, and the top-10 are identified as important. The important regions identified such as Hippocampus, Entorhinal and ParsOpercularis are relevant according to existing AD domain knowledge and in accordance with prior studies [2, 20].

Table 1. Comparison of baseline methods and HLSGL-HPS

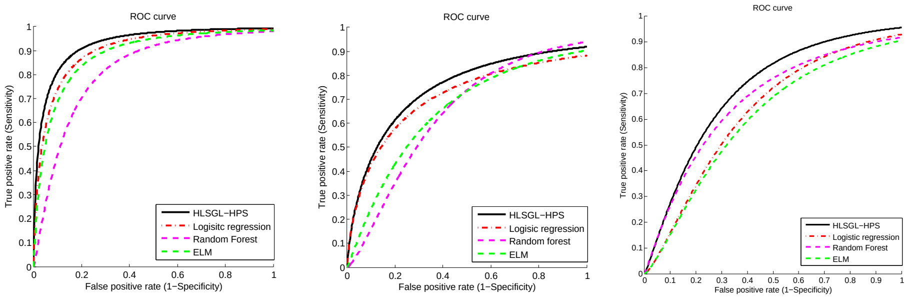
	AD vs. NC				AD vs. MCI				MCI vs. NC			
	Sen	Spec	G-mean	AUC	Sen	Spec	G-mean	AUC	Sen	Spec	G-mean	AUC
All	0.802	0.932	0.865	0.859	0.277	0.905	0.501	0.510	0.342	0.865	0.544	0.433
HLL	0.811	0.941	0.874	0.879	0.253	**0.928**	0.485	0.523	0.208	**1**	0.456	0.517
HLSGL	**0.829**	**0.948**	**0.882**	**0.892**	0.334	0.919	0.536	0.571	0.253	0.965	0.494	0.594
All-HPS	-	-	-	-	0.586	0.737	0.657	0.682	0.418	0.687	0.536	0.593
HLL-HPS	-	-	-	-	0.645	0.672	0.659	0.671	0.544	0.692	0.613	0.606
HLSGL-HPS	-	-	-	-	**0.691**	0.696	**0.693**	**0.710**	**0.646**	0.576	**0.650**	**0.682**

Table 2. Average values of G-mean and AUC for all compared classification methods(Note that $\star$ stands for the case with $p \leq 0.05$)

Methods	G-mean			AUC		
	AD vs NC	AD vs MCI	MCI vs NC	AD vs NC	AD vs MCI	MCI vs NC
Random Forest	$0.861^\star$	$0.609^\star$	0.632	$0.833^\star$	0.707	$0.653^\star$
ELM	$0.855^\star$	$0.661^\star$	0.631	$0.857^\star$	$0.691^\star$	$0.636^\star$
Logistic regression	$0.727^\star$	$0.652^\star$	$0.619^\star$	$0.820^\star$	$0.663^\star$	$0.617^\star$
Proposed method	**0.882**	**0.693**	**0.650**	**0.892**	**0.722**	**0.682**

3.3 The effect of re-sampling ratio on the performance

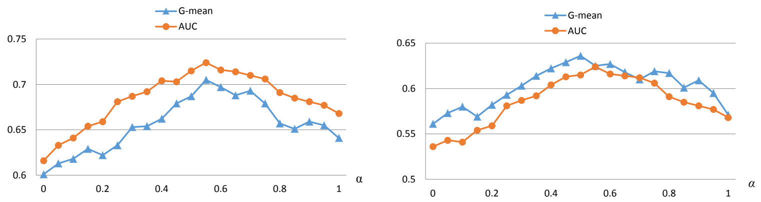
In HPS, the optimal re-sampling ratio α may be unknown, and the parameter plays a vital role for the performance of hybrid re-sampling on the imbalanced data learning. In the previous experiments, α is set to 70% empirically. The experiment shows the performance by tuning the re-sampling ratio. The range of α is [0,100%]; the step is set to 5%. With each value of α, we conduct 10-fold cross validation to obtain an averaged G-mean and AUC results. From Fig. 2, we can see the changes of G-mean and AUC when varying the value of α. When α is 0, only undersampling for majority class is carried out and no new instances are generated. Important information of majority class may be lost, hence the performance is lowest. When increasing the value of α, the



(a) AD vs. NC (b) MCI vs. NC (c) AD vs. MCI

Fig. 1. The ROC of multiple computing methods

two performances increase. When α is 1, oversampling for minority class is performed without removing redundant instances for majority class. The issue of overfitting may occur due to the large amount of the minority class as well as the redundant information of majority class. Moreover, we found the G-mean and AUC to achieve the best when α is 55% for AD vs MCI, and 45% as well as 55% for MCI vs NC respectively. It demonstrates that the hybrid re-sampling scheme with an appropriate re-sampling ratio can achieve optimal classification performance.



(a) AD vs MCI (b) MCI vs NC

Fig. 2. The performance of varying oversampling ratio with respect to G-mean and AUC

4 Conclusion

In this paper, we focused on the high feature-dimension and class imbalanced distribution problem in AD diagnosis. Specifically, we proposed a novel feature selection method by sparse group lasso and hybrid probabilistic oversampling to preprocess the high dimensional imbalanced data, to improve the prediction performance. In our experimental results on the ADNI dataset, we validated the efficacy of the proposed method by enhancing classification performance in terms of G-mean and AUC. In our future works, we will extend the proposed linear sparse feature selection to the nonlinear s-parse modal via kernel functions to capture complex patterns for AD diagnosis.

5 Acknowledgments

This research was supported by the National Natural Science Foundation of China (61502091), the Fundamental Research Funds for the Central Universities (N140403004), and the Postdoctoral Science Foundation of China (2015M570254).

References

1. Brookmeyer, R., Johnson, E., Ziegler-Graham, K.: Forecasting the global burden of Alzheimers disease, *Alzheimer's & dementia*, **3**(3):186-191 (2007)
2. Zhu, X., Suk, H., Shen, D.: Subspace regularized sparse multi-task learning for multi-class neurodegenerative disease identification, *IEEE Transactions on Biomedical Engineering*, **63**(3): 607–618 (2015)

3. Peng, X., Lin, P., Zhang, T., Wang, J.: Extreme learning machine-based classification of ADHD using brain structural MRI data, *PloS one*, **8**(11) (2013)
4. Dubey, R., Zhou, J., Wang, Y., Thompson, P.M., Ye, J., others.: Analysis of sampling techniques for imbalanced data: An n= 648 ADNI study, *NeuroImage*, **87**: 220–241 (2014)
5. He, H., Garcia, E.A.: Learning from imbalanced data, *IEEE Transactions on Knowledge and Data Engineering*, **21**(9):1263–1284 (2009)
6. Cao, P., Zhao, D., Zaiane, O.: An optimized cost-sensitive SVM for imbalanced data learning, *Advances in Knowledge Discovery and Data Mining*:280–292 (2013)
7. Weiss, G.: The impact of small disjuncts on classifier learning, *Annals of Information Systems*, **5**(8):193–226 (2010)
8. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **68**(1): 49–67 (2006)
9. Beck A.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM journal on imaging sciences*:183–202 (2009)
10. Liu, F., Zhou, L., Shen, C., Yin, J.: Multiple kernel learning in the primal for multimodal Alzheimers disease classification, *IEEE journal of biomedical and health informatics*, **18**(3):984–990 (2014)
11. Hinrichs, C., Singh, V., Peng, J., Johnson, S.: Q-mkl: Matrix-induced regularization in multikernel learning with applications to neuroimaging, *Advances in neural information processing systems*, 1421–1429, 2012.
12. Gu, B., Sheng, V.S.: A robust regularization path algorithm for ν-support vector classification, *IEEE Transactions on Neural Networks and Learning Systems* (2016)
13. Gu, B., Sheng, V.S., Wang, Z., Ho, D., Osman, S., Li, Shuo.: Incremental learning for ν-support vector regression, *Neural Networks*, **67**:140–150 (2015)
14. Ye, J., Liu, J.: Sparse methods for biomedical data, *ACM Sigkdd Explorations Newsletter*, **14**(1):4–15 (2012)
15. Maldonado, S., Weber, R., Famili, F.: Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines, *Information Sciences*, **286**:228–246 (2014)
16. Yuan, L., Liu, J., Ye, J.: Efficient methods for overlapping group lasso, *Advances in Neural Information Processing Systems*, 352–360 (2011)
17. Liu, J., Ye, J.: Moreau-Yosida regularization for grouped tree structure learning, *Advances in Neural Information Processing Systems*, 1459–1467 (2010)
18. Figueiredo M.A.T., Jain A.K., Doi, K.: Unsupervised learning of finite mixture models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(3):381–96 (2002)
19. Nesterov, Y.: Smooth minimization of non-smooth functions, *Mathematical Programming*, **103**(1):127-152 (2005)
20. Wan, J., Zhang, Z., Rao, B.D., Fang, S., Yan, J., Saykin, A.J., Shen, L.: Identifying the neuroanatomical basis of cognitive impairment in Alzheimer's disease by correlation-and nonlinearity-aware sparse Bayesian learning, *IEEE transactions on medical imaging*, **33**(7): 1475–1487, 2014.
21. Weiner, M.W., Aisen, P.S., Jack, C.R., Jagust, W.J., others: The alzheimer's disease neuroimaging initiative: progress report and future plans. *A*lzheimers Dement, **6**:202–211 (2010)
22. Ye, J., Farnum, M., Yang, E., Verbeeck, R., others: Sparse learning and stability selection for predicting MCI to AD conversion using baseline ADNI data, *BMC neurology*, **12**(46): 1–12 (2012)
23. Lebedev, A.V., Westman, E., Van Westen, G.J.P., Kramberger, M.G., others: Random Forest ensembles for detection and prediction of Alzheimer's disease with a good between-cohort robustness, *NeuroImage: Clinical*, **6**:115–125 (2014)