Feature Learning and Classification in Neuroimaging: Predicting Cognitive Impairment from Magnetic Resonance Imaging

Shan Shi, Farouk S. Nathoo*

Department of Mathematics and Statistics, University of Victoria

* Corresponding Author: nathoo@uvic.ca

1. Introduction

- Due to the rapid innovation of technology and the desire to find and employ biomarkers for neurodegenerative disease, high-dimensional data classification problems are routinely encountered in neuroimaging studies.
- To avoid over-fitting and to explore relationships between disease and potential biomarkers, feature learning plays an important role in classifier construction and is an important area in machine learning.
- In this work, we first review several important feature learning techniques including:
  1 Lasso-based methods
  2 The two-sample t-test
  3 Principal Component Analysis (PCA)
  4 Semi-supervised stacked auto-encoders (semi-SAE)
- We then compare the effectiveness of different feature learning methods by considering a binary classification problem where the goal is to use MRI data and classify subjects into one of two groups: those with Alzheimer's disease (AD) or normal controls (NC).
- In our study, the data are from the Alzheimer's Disease Neuroimaging Initiative ADNI-I study (http://adni.loni.usc.edu). We use 632 subjects. Among them, 144 were diagnosed with AD, 179 are NC, and the remaining 309 were categorized as having mild cognitive impairment (MCI), which in our study are considered as target-unrelated subjects.

2. Lasso-Based Methods

- Given a data matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$, we here assume that each column of $\mathbf{X}$ is standardized. The lasso proposed by Tibshirani (1996) and the lasso estimate $\hat{\beta}$ in the linear model is:

$$\hat{\beta}(\lambda) = \underset{\beta}{\operatorname{argmin}} \left(\|Y - \mathbf{X}\beta\|_2^2/n + \lambda\|\beta\|_1 \right),$$

  where $\|Y - \mathbf{X}\beta\|_2^2 = \sum_{i=1}^n \left(Y_i - X_i^T\beta\right)^2$, $\|\beta\|_1 = \sum_{j=1}^p |\beta_j|$ and the penalty parameter $\lambda > 0$.
- An important property of the lasso is that the estimator is sparse so that $\hat{\beta}(\lambda)_j = 0$ for some j's.
- The features with $\hat{\beta}(\lambda)_j \neq 0$ can be thought of as those features that have been selected by the lasso.
- Applications of lasso-based feature learning in neuroimaging classification are considered in [1]. They propose to fit a linear regression model with the lasso to the training data for feature selection. Then a classifier is trained using the selected features.
- It is worth mentioning that lasso-based feature selection methods are easy to implement and could help improve classification accuracy but the selected features such as genes or SNPs may not be the right ones for understanding genome-wide disease association, due to incidental endogeneity [2].
- Generally speaking, the more covariates that are collected which are thought to potentially related to the response, the less likely all covariates are uncorrelated with the residual noise.
- When this fundamental assumption $\mathbb{E}(\epsilon X) = 0$ cannot be guaranteed to hold, [3] show that this causes model selection inconsistency of the penalized least-squares method.

3. Two Sample t-test

- The two-sample t-test is an easily employed and simple feature selection method when dealing with classification and a large number of features.
- Reference [4] show that under some mild conditions, the t-test can correctly select all important features with high probability.
- Let $\bar{X}_{kj}$ and S_{kj}^2 be the sample mean and variance of the j-th feature in class k, where $k = 0, 1$, and $j = 1, \ldots, p$. The two-sample t-test statistic for feature j is defined as:

$$T_j = \frac{\bar{X}_{0j} - \bar{X}_{1j}}{\sqrt{S_{0j}^2/n_0 + S_{1j}^2/n_1}},\ j = 1, \ldots, p.$$

- Assume that $\mu = \mathbb{E}(X|Y=0) - \mathbb{E}(X|Y=1) = (\mu_1, \ldots, \mu_p)$ is sparse with only the first s entries nonzero and some other mild conditions, the following result holds:

$$\mathbb{P}\left(\min_{j\leq s}|T_j| > x, \max_{j>s}|T_j| < x\right) \to 1,\ \text{as } n, p \to \infty,$$

  where x is some positive constant.
- When using the independence classification rule [4], the optimal number of features, m, these authors suggest can be estimated by minimizing the following misclassification rate upper bound:

$$\hat{m}_{opt} = \underset{1\leq m\leq p}{\operatorname{argmax}} \frac{1}{\hat{\lambda}_{max}^m} \frac{n\left[\sum_{j=1}^m T_{(j)}^2 + m(n_0 - n_1)/n\right]^2}{mn_0n_1 + n_0n_1\sum_{j=1}^m T_{(j)}^2},$$

  where $T_{(1)}^2 \geq T_{(2)}^2 \geq \cdots \geq T_{(p)}^2$ are the ordered squared t-test statistics, $n = n_0 + n_1$, and $\hat{\lambda}_{max}^m$ is the largest eigenvalue of the sample correlation matrix $\mathbf{R}^m$ of the truncated observation, that is, only the first m highest t-statistic features are considered.
- For any classifier other than the independence classification rule, m should be treated as a tuning parameter and cross-validation can be used to find the optimal m.

4. Principal Component Analysis (PCA)

- Given a data matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$, which in the current context represents n observations on a set of p features, we have n data points $\{X_i, i = 1, \ldots, n\}$ in $\mathbb{R}^p$ with $\mu_0 = \mathbb{E}(X|Y=0)$ and $\mu_1 = \mathbb{E}(X|Y=1)$.
- Here, we assume that the variance of each feature is one. The sample covariance matrix is $\mathbf{S} = \frac{1}{n}\sum_{i=1}^n \left(X_i - \bar{X}\right)\left(X_i - \bar{X}\right)^T$, where $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$.
- The first method to find the principal components is by finding the directions of maximum variance. The first r principal component directions, $\mathbf{V}_{p\times r}$, can be found by solving the following optimization problem, where $r \leq p$

$$\mathbf{V}_{p\times r} = \underset{\mathbf{A}:\mathbf{A}^T\mathbf{A}=\mathbf{I}_r}{\operatorname{argmax}}\ trace\left(\mathbf{A}^T\mathbf{S}\mathbf{A}\right).$$

- Another approach is based on minimizing what is referred to as the reconstruction error, where we seek the matrix $\mathbf{A}_{p\times r}$ with orthonormal columns i.e. $\mathbf{A}^T\mathbf{A} = \mathbf{I}_r$ which minimizes:

$$\frac{1}{n}\sum_{i=1}^n \| \left(X_i - \bar{X}\right) - \mathbf{A}\mathbf{A}^T\left(X_i - \bar{X}\right) \|_2^2.$$

- The projection matrix $\mathbf{H} = \mathbf{A}\mathbf{A}^T$ projects each sample point X_i to the r dimension subspace spanned by the columns of $\mathbf{A}$.
- The easiest way of applying PCA for high-dimensional classification is to reduce the dimensionality of the data by replacing $X \in \mathbb{R}^p$ by its first $m < p$ principal components.
- Interestingly, the PCs with high variance are not necessarily useful for separation between two populations. This is because the direction of highest variance may not well-separate the sample means after projection [6].

5. Semi-supervised Stacked Auto-encoder (semi-SAE)

- An auto-encoder is a feature learning approach that can be thought of as a generalization of PCA.
- While PCA is based on projecting the data onto linear subspaces, auto-encoders enable us to deal with a curved manifold in the input space, so the data are represented by projections on the curved manifold.
- Let f_θ and g_θ be functions which we will call the the encoder and decoder respectively. For each data point $X_i \in \mathbb{R}^p$, we compute a *representation* h_i from X_i using the encoder as follows: $h_i = f_\theta(X_i)$. The decoder g_θ is then a mapping from the feature space back to the input space and is intended to produce a reconstruction of the input data $\hat{X}_i = g_\theta(h_i)$.
- Like PCA, the parameters θ characterizing the encoder and decoder are estimated by minimizing the reconstruction error:

$$L(\theta) = \sum_{i=1}^n \|X_i - g_\theta(f_\theta(X_i))\|_2^2.$$

- An auto-encoder can be used as a a building block in a more complex structure known as a stacked auto-encoder (SAE).
- The semi-supervised approach is considered. That is, we use both the target-related training and target-unrelated samples to train the auto-encoders.

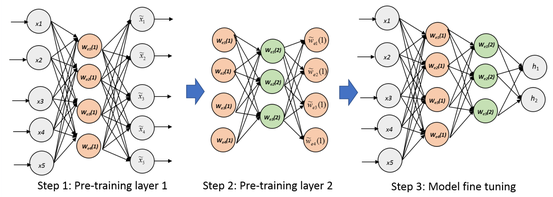

Figure 1:Stacked Auto-Encoder[a]

[a] from book *R Deep Learning Cookbook*

6. Experimental Setup

- In the experiment, a linear support vector machine (SVM) is employed as the classifier at the final level for all methods considered.
- We randomly split the dataset into a training set and a test set. The size of the test set is 20% of the total dataset.
- We use 10-fold cross-validation on the training set to choose the hyperparameters for each method, for example the number of PCs, and tuning parameters associated with penalty terms.
- We use the test set to evaluate the classification accuracy. This procedure is repeated 100 times to achieve a better evaluation of the classification accuracy.

The feature learning methods that are evaluated in this experiment are as follows:

- No feature selection (no FS), using MRI features (LLF) only.
- No feature selection, using features learned by SAE (SAEF) and by semi-supervised SAE (semi-SAEF).
- No feature selection, using features created by concatenating the high-level features (SAEF/semi-SAEF) with LLF, namely, LLF + SAEF and LLF + semi-SAEF.
- PCA, using MRI features (LLF) only.
- Two sample t-test, using MRI features (LLF) only.
- LASSO-based methods, using MRI features (LLF), LLF + SAEF, and LLF + semi-SAEF.

7. Experimental Results

Table 1:

Performance comparison of different feature learning methods in the classification of AD versus NC (in % accuracy).

	LLF	LLF+SAEF	LLF+semi-SAEF	SAEF	semi-SAEF
No FS	83.9	83.5	84.1	86.9	87.7
Lasso	85.7	84.6	85.3		
t-test	85.9				
PCA	85.7				

- **Semi-SAEF and SAEF exhibit the best performance,** 87.7% **and** 86.9% **(SAEF)**.
- Directly concatenating the high-level features (SAEF/semi-SAEF) with LLF does not improve the accuracy.
- The combination of LLF+SAEF/LLF+semi+SAEF and the lasso-based method to select a subset of the features seems to improve the performance slightly, but the performance is no better than LLF+LASSO.
- The simple approaches LLF+PCA and LLF+t-test perform essentially equivalently to LLF+LASSO.

8. Selected References

[1] Suk HI, Lee SW, Shen D, Alzheimer's Disease Neuroimaging Initiative. Latent feature representation with stacked auto-encoder for AD/MCI diagnosis. Brain Structure and Function. 2015 Mar 1;220(2):841-59.

[2] Fan J, Han F, Liu H. Challenges of big data analysis. National science review. 2014 Jun 1;1(2):293-314.

[3] Fan J, Liao Y. Endogeneity in high dimensions. Annals of statistics. 2014 Jun 1;42(3):872.

[4] Fan J, Fan Y. High dimensional classification using features annealed independence rules. Annals of statistics. 2008;36(6):2605.

[5] An L, Adeli E, Liu M, Zhang J, Lee SW, Shen D. A Hierarchical Feature and Sample Selection Framework and Its Application for Alzheimer's Disease Diagnosis. Scientific reports. 2017 Mar 30;7:45269.

[6] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence. 2013 Aug;35(8):1798-828.

[7] Jolliffe I, Principal Component Analysis, New York: Springer Series in Statistics; 2002

Acknowledgements

This research is supported by NSERC through the Discovery Grants program and through the Canada Research Chair in Biostatistics for Spatial and High-Dimensional Data; by CANSSI through a Collaborative Research Team Grant 'Joint Analysis of Neuroimaging Data: High-Dimensional Problems, Spatiotemporal Models and Computation'; by the University of Victoria through a UVic Internal Research Grant and the Faculty of Graduate Studies.

University of Victoria

CANSSI INCASS

NSERC CRSNG