Communications for Statistical Applications and Methods
2014, Vol. 21, No. 1, 61–68

DOI: http://dx.doi.org/10.5351/CSAM.2014.21.1.061
ISSN 2287-7843

# Logistic Regression Classification by Principal Component Selection

Kiho Kim[a], Seokho Lee[1,a]

[a]Department of Statistics, Hankuk University of Foreign Studies, Korea

**Abstract**

We propose binary classification methods by modifying logistic regression classification. We use variable selection procedures instead of original variables to select the principal components. We describe the resulting classifiers and discuss their properties. The performance of our proposals are illustrated numerically and compared with other existing classification methods using synthetic and real datasets.

Keywords: Logistic regression classification, principal components, sparse regression.

## 1. Introduction

Logistic regression (LR) classification is a popular statistical methodology for binary classification problems, and has been widely applied to numerous application domains. Popular statistical classification methods (other than LR classification) include linear discriminant analysis (LDA) and a support vector machine (SVM) as described in standard statistical machine learning textbooks (Bishop, 2006; Hastie *et al.*, 2009; Murphy, 2012).

LR classification is based on logistic regression where the binary response variable is the class label. Suppose $y$ is a binary variable, taking values $\{0, 1\}$ depending on its class, and $\mathbf{x} \in \mathbb{R}^p$ is a feature vector having $p$ attributes for the sample. The classifier from LR classification is given as $f(\mathbf{x}) = \alpha + \mathbf{x}^T\boldsymbol{\beta}$, so that classification is made as '0' class if $f(\mathbf{x}) < 0$ and '1' class if $f(\mathbf{x}) > 0$. The parameters of the classifier are obtained by minimizing the negative Bernoulli likelihood:

$$\left(\hat{\alpha}, \hat{\boldsymbol{\beta}}\right) = \underset{\alpha,\, \boldsymbol{\beta}}{\arg\min}\, L(\alpha, \boldsymbol{\beta})$$

with $L(\alpha, \boldsymbol{\beta}) = -\sum_{i=1}^{n} l(\alpha, \boldsymbol{\beta}; y_i, \mathbf{x}_i)$ and $l(\alpha, \boldsymbol{\beta}; y_i, \mathbf{x}_i) = y_i(\alpha + \mathbf{x}_i^T\boldsymbol{\beta}) - \ln(1 + e^{\alpha + \mathbf{x}_i^T\boldsymbol{\beta}})$. To avoid overfitting, penalized logistic regression (PLR) classification is often used. PLR imposes a penalty on large fluctuations on $\boldsymbol{\beta}$ and thus on the fitted classifier. This approach is most valuable in high-dimensional situations. Types of penalty functions are optionally selective depending on the purpose of analysis. Another direction to avoid overfitting and/or high-dimensionality is to reduce the variability of the classifier estimate using dimension reduction techniques such as PC regression in the regression problems. A straightforward analogy for LR classification is to use the major principal components (PCs) as covariates in LR classification and throw away the remaining minor PCs; however, a drawback is that it may lead to poor classification in the test dataset because some of minor PCs can be strongly associated with class labels and are not used to construct classifiers.

This research was supported by a Hankuk University of Foreign Studies Research Fund of 2013.

[1] Corresponding author: Assistant Professor, Department of Statistics, Hankuk University of Foreign Studies, 81 Oedae-ro, Yongin, Gyeonggi-do 449-791, Korea. E-mail: lees@hufs.ac.kr

Published online 31 January 2014 / journal homepage: http://csam.or.kr

In this work, we propose a simple modification for PC-regression-like LR classification. Rather than a few major PCs, we select PCs by logistic regression with sparsity-inducing penalties. Thus, a sparse logistic regression procedure automatically locates the PCs that have discriminating power as covariates even though they are not considered major PCs. This simple approach is effective to reduce the misclassification rate produces a principal subspace where the class information is maximized. In Section 2, we describe our new classifiers and their properties. Numerical studies are provided in Section 3 with a comparison of other classification rules. The conclusion discusses possible extensions to multi-class classification.

## 2. Method

Suppose $\mathbf{X} = (\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_n^T)^T$ is the matrix of $n \times p$, whose columns represent $p$ feature variables and rows represent $n$ samples. We assume that all columns are centered. Principal components are derived from the linear transformation $\mathbf{Z} = \mathbf{XV}$, where $\mathbf{V} \in \mathbb{R}^{p\times r}$ with $r = \text{rank}(\mathbf{X})$ is the orthogonal matrix of principal component loadings. Thus, principal components for the $i^{\text{th}}$ sample is given by $\mathbf{z}_i = \mathbf{V}^T\mathbf{x}_i$ for $i = 1, 2, \ldots, n$, where $\mathbf{z}_i$ is the $i$th row of $\mathbf{Z} \in \mathbb{R}^{n\times r}$. The linear classifier can be rewritten in terms of principal components as

$$f(\mathbf{x}) = \alpha + \mathbf{x}^T\boldsymbol{\beta} = \alpha + \mathbf{z}^T\boldsymbol{\gamma},$$

where $\boldsymbol{\gamma} = \mathbf{V}^T\boldsymbol{\beta} \in \mathbb{R}^r$. Our new proposal is to minimize the negative penalized Bernoulli likelihood with sparsity-inducing penalty $\texttt{pen}_\lambda(\boldsymbol{\gamma})$ on the coefficient vector $\boldsymbol{\gamma}$ of principal components. Thus, the new classifier we suggest is $f(\mathbf{x}) = \hat{\alpha} + \mathbf{x}^T\mathbf{V}\hat{\boldsymbol{\gamma}}$ from

$$(\hat{\alpha}, \hat{\boldsymbol{\gamma}}) = \underset{\alpha,\, \boldsymbol{\gamma}}{\arg\min}\, L(\alpha, \boldsymbol{\gamma}) + \texttt{pen}_\lambda(\boldsymbol{\gamma}).$$

The sparsity-inducing penalty forces some principal component coefficients to be zero if they do not contribute to class separation. Minor principal components may survive penalization if they are still helpful for class separation. The resulting classification rule implies the following properties:

1. Unlike PC regression scheme, where a few major principal components are included in the set of covariates, important minor principal components are retained if they are meaningful in classification.

2. Similar with LDA or penalized LDA, low dimensional subspace (principal subspace) can be found and its basis (principal component directions) delivers how the original variables contribute to create a subspace where maximal class separation is possible (for the reduced-rank LDA, refer to Section 4.3.3. in Hastie *et al.*, 2009).

3. The dimensionality of principal subspace (the number of PCs selected) is automatically selected through a sparse logistic regression procedure. Moreover, such dimension can be even larger than the number of classes (2 for two-classification). In contrast, LDA or penalized LDA find the subspace of dimension $G - 1$ for $G$-class classification (see Chapter 12 in Hastie *et al.*, 2009). This implies that our proposal may have a better position than LDA when classes have complicated shapes in the data space (*e.g.*, samples from the single class are generated from mixture distribution.)

4. The subspace from our proposal has an interpretational advantage in that its bases are PC loading vectors. Each direction explains its own mode of variability of data space as principal component

analysis does. The orthogonality of basis makes its interpretation easy, while the direction from LDA are not orthogonal in the data space. In genome-wide association studies, genetic variations are highly associated with population stratification (a systematic difference in genetic variables due to different ancestry) and such variations are well represented by principal components (Patterson *et al.*, 2006; Price *et al.*, 2006). Consequently, a disease status of interest classified with some principal components associated with population stratification may be explained by ethnic difference.

In this study, we consider 2 types of sparsity-inducing penalties: (1) Lasso penalty, $\texttt{pen}_\lambda(\boldsymbol{\gamma}) = \lambda \sum_{j=1}^{p} |\gamma_j|$ (Tibshirani, 1996), and (2) SCAD penalty, $\texttt{pen}_\lambda(\boldsymbol{\gamma}) = \sum_{j=1}^{p} p_\lambda(|\gamma_j|; a)$ with $p_\lambda(x; a) = 2\lambda x I(x \le \lambda) - \{(x^2 - 2a\lambda x + \lambda^2)/(a-1)\} I(\lambda < x \le a\lambda) + (a+1)\lambda^2 I(x > a\lambda)$ (Fan and Li, 2005). We name the classifiers associated with 2 types of penalties as PPCLR-L and PPCLR-S respectively.

## 3. Numerical Results

### 3.1. Synthetic data analysis

We tested PPCLR-L and PPCLR-S and compared them with other existing methods using synthetic datasets. In the simulation, two random vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ were generated from the multivariate normal distributions with $N_p(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $N_p(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ respectively. First, consider the covariance matrix $\boldsymbol{\Sigma} = (\Sigma_{ij})$ with

$$\Sigma_{ij} = \frac{1}{|i-j|+1} \quad (i, j = 1, 2, \ldots, p)$$

as in Kondylis and Whittaker (2008). This covariance matrix $\boldsymbol{\Sigma}$ represents that there exists a high correlation between the variables. Its spectral decomposition is

$$\boldsymbol{\Sigma} = \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^\top = \sum_{\ell=1}^{p} \lambda_\ell \mathbf{p}_\ell \mathbf{p}_\ell^\top,$$

where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_p)$ and $\mathbf{P} = (\mathbf{p}_1, \ldots, \mathbf{p}_p)$ with the ordered eigenvalues $\lambda_1 \ge \cdots \ge \lambda_p \ge 0$ and the associated eigenvectors $\mathbf{p}_1, \ldots, \mathbf{p}_p$. Now we consider 4 setups for data generation:

(1) **Case 1**: $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$ and $\boldsymbol{\mu}_1 = \boldsymbol{\mu}, \boldsymbol{\mu}_2 = -\boldsymbol{\mu}$ with

$$\boldsymbol{\mu} = \frac{\sum_{\ell=1}^{k} \sqrt{\lambda_\ell}\mathbf{p}_\ell}{\sqrt{\sum_{\ell=1}^{k} \lambda_\ell}},$$

where $k$ is the minimum number of major eigenvalues whose sum takes up more than 90% of total variability.

(2) **Case 2**: $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$ and $\boldsymbol{\mu}_1 = \boldsymbol{\mu}, \boldsymbol{\mu}_2 = -\boldsymbol{\mu}$ with

$$\boldsymbol{\mu} = \frac{\sum_{\ell=k+1}^{p} \sqrt{\lambda_\ell}\mathbf{p}_\ell}{\sqrt{\sum_{\ell=k+1}^{p} \lambda_\ell}},$$

where $k$ is the minimum number of minor eigenvalues whose sum takes up more than 10% of total variability.

(3) **Case 3**: $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}_2 = \mathbf{P}\boldsymbol{\Lambda}_\pi\mathbf{P}^\top$ where

$$\boldsymbol{\Lambda}_\pi = \text{diag}(\lambda_p, \lambda_{p-1}, \lambda_3, \ldots, \lambda_{p-2}, \lambda_2, \lambda_1),$$

and $\boldsymbol{\mu}_1 = \boldsymbol{\mu}, \boldsymbol{\mu}_2 = -\boldsymbol{\mu}$ with

$$\boldsymbol{\mu} = \frac{\sqrt{\lambda_1 + \lambda_p}\mathbf{p}_1 + \sqrt{\lambda_2 + \lambda_{p-1}}\mathbf{p}_2}{\sqrt{\lambda_1 + \lambda_2 + \lambda_{p-1} + \lambda_p}}.$$

(4) **Case 4**: $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}_2 = \mathbf{P}\boldsymbol{\Lambda}_\pi\mathbf{P}^\top$ with $\boldsymbol{\Lambda}_\pi = \text{diag}(\lambda_{\pi_\ell})$ where $\pi$ denotes the random permutation of $\{1, 2, \ldots, p\}$ and $\pi_\ell$ is the $\ell$th element of that permutation. Define the pooled covariance of $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ as $\boldsymbol{\Sigma}^* = p_1\boldsymbol{\Sigma}_1 + p_2\boldsymbol{\Sigma}_2$ where $p_1$ and $p_2$ are sample proportions of $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively. Let $\mathbf{p}_1^*$ be the eigenvector associated with the first eigenvalue of $\boldsymbol{\Sigma}^*$. Then, we set $\boldsymbol{\mu}_1 = \boldsymbol{\mu}$ and $\boldsymbol{\mu}_2 = -\boldsymbol{\mu}$ with $\boldsymbol{\mu} = \mathbf{p}_1^*$.

Case 1 and Case 2 assume the common covariance for both classes. For Case 1, means for two classes locate along the direction of the weighted average of major eigenvectors, while in Case 2, means are on the direction of the weighted average of minor eigenvectors. Thus, we expect that PCLR (see the below for its definition) is still good in Case 1 and performs badly in Case 2. Case 3 and Case 4 have more complicated situations: Case 3 has different covariances and the means lie on the direction of the weighted average of two specific eigenvectors. Case 4 also assume different covariances and the means locate on the direction of the first eigenvector of the pooled covariance.

From the above simulation setups, we generated $n/2$ samples from the distributions of $\mathbf{x}_1$ and $\mathbf{x}_2$ equally. The number of samples ($n$) and variables ($p$) we considered are: (1) large sample case of $p = 10$, $n = 100, 200, 400, 800, 1600$ and (2) high dimensional case of $n = 100$, $p = 100, 200, 400, 800, 1600$.

For the comparison, 4 methods are also applied to the simulated datasets as well as 2 proposals (PPCLR-L and PPCLR-S). 4 competitors are:

- PCLR : logistic regression classification with the major principal components that explain a 90% variability of feature space.
- PLR : penalized logistic regression classification with ridge penalty (Friedman *et al.*, 2008). `glmnet` package was used.
- PLDA : penalized linear discriminant analysis with ridge penalty (Hastie *et al.*, 1995). An `mda` package was used.
- SVM : support vector machine with radial basis kernel. A `kernlab` package was used.

We applied our proposals and 4 competitors to the simulated dataset to compare them based on test misclassification rate. For all methods (except for PCLR), 10-fold cross-validation chose the optimal penalty parameter associated with the penalty functions. The test misclassification rate is evaluated as a cross-validation error rate. To obtain the consistent results, we iterated the same simulation 1,000 times and provided the averages of 1,000 test misclassification rates in Figure 1 and Figure 2 as graphical summaries.

Figure 1 indicates that the test misclassification rates from all methods tend to decrease as the sample size increases. In Case 1 and Case 2, PPCLR-L and PPCLR-S outperform other competitors; however, SVM has the best performance among other methods (including our proposals in Case 3

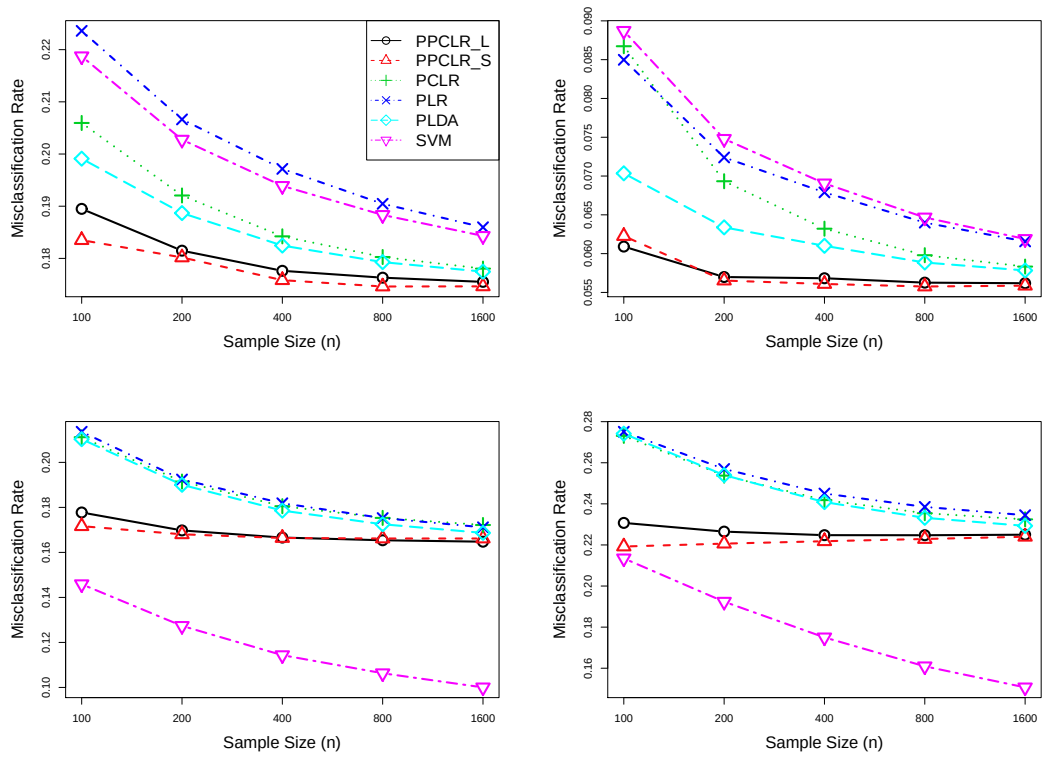


Figure 1: *Average test misclassification rate with $p = 10$ (Top-left panel is for Case 1, top-right is for Case 2, bottom-left is for Case 3, and bottom-right is for Case 4)*

and Case 4) because SVM produces flexible classifiers that are best when the covariance structure of two classes are not homogeneous. SVM is the best in Case 3 and Case 4; however, PPCLR-L and PPCLR-S are the second best among others. Figure 2 shows the results for a high-dimensional situation, where test misclassification rates from all methods increase as dimensionality increases. However, their trends differ by methods. SVM is still good in $p = 100$ and comparable to sample size $n = 100$; however, but it quickly deteriorates as $p$ increases. For Case 1 and Case 2, PPCLR-L, PPCLR-S, and PLDA are relatively better than others and seem comparable to each other. However, PPCLR-L and PPCLR-S tend to have the smallest test misclassification rates when the dimension is very large ($p = 1600$). The simulation studies suggest that PPCLR-L and PPCLR-S work well in high-dimensional scenarios. We also notice that PPCLR-S seems to indicate better performance than PPCLR-L even though the difference looks marginal. We guess that oracle property (unbiasedness of $\hat{\boldsymbol{y}}$) of SCAD penalty in variable selection under regression problems carries over to PPCLR-S to enhance the classification. In addition, we applied PLDA and PLR with a Lasso penalty to the same simulated dataset as well. Their results (not shown here) are similar to the the ridge penalty case; in addition, the priority of PPCLR-L and PPCLR-S is unchanged. We also note that the sparse logistic regression classification using original variables (SPLR) was applied to the simulated data sets and its results (not shown here) are comparable to PPCLR. The performance comparison between PPCLR and SPLR highly depends on the situations where the data is generated; however, we do not consider a PPCLR and SPLR comparison since it is not the primary interest of this study

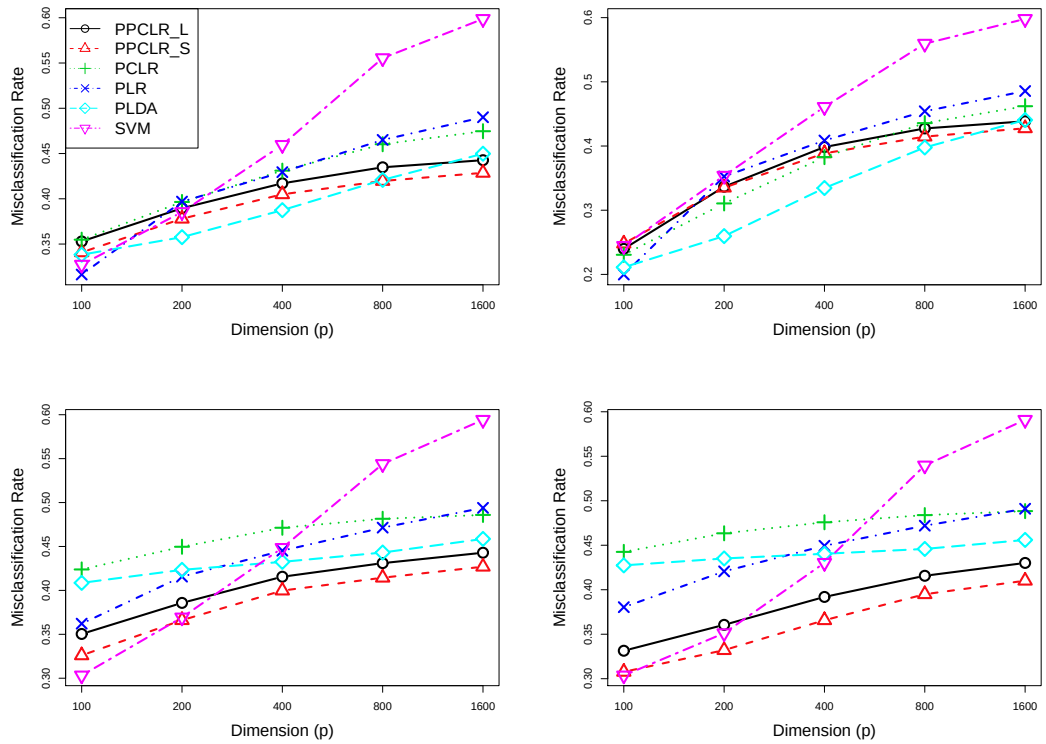


Figure 2: *Average test misclassification rate with $n = 100$ (Top-left panel is for Case 1, top-right is for Case 2, bottom-left is for Case 3, and bottom-right is for Case 4)*

Table 1: Description of real datasets for binary classification.

| Data | Samples ($n$) | Dimensions ($p$) | Type / Application | Source |
|---|---|---|---|---|
| `Ionosphere` | 351 | 34 | signal / physics | UCI |
| `Sonar` | 208 | 60 | signal / material science | KEEL |
| `Spambase` | 4597 | 57 | text / text mining | KEEL |
| `Spectheart` | 297 | 44 | image / medical science | KEEL |
| `WDBC` | 569 | 30 | image / medical science | KEEL |
| `Chin` | 118 | 22215 | microarray / medical science | `datamicroarray` package |
| `Chowdary` | 104 | 22283 | microarray / medical science | `datamicroarray` package |
| `Gravier` | 168 | 2905 | microarray / medical science | `datamicroarray` package |
| `Gordon` | 181 | 12533 | microarray / medical science | `datamicroarray` package |
| `Singh` | 102 | 12600 | microarray / medical science | `datamicroarray` package |
| `Shipp` | 58 | 6817 | microarray / medical science | `datamicroarray` package |

## 3.2. Real data analysis

We applied all methods to several real datasets from various application domains (see Table 1). We collected 11 datasets from UCI machine learning repository (Bache and Lichman, 2013), the data repository from KEEL webpage [http://www.keel.es] (Alcalá-Fdez *et al.*, 2011), and `datamicroarray` package from GitHub [http://github.com]. All datasets consist of 2 classes and are suitable for the binary classification task. Sample size and dimensionality are radically different across datasets.

Table 2 summarizes the test error rates from all methods considered in this paper that added the

Table 2: Misclassification rates (%).

| Data | PPCLR-L | PPCLR-S | PCLR | PLR | SPLR | PLDA-R | SVM |
|---|---|---|---|---|---|---|---|
| Ionosphere | 14.53 | 15.10 | 14.27 | 14.81 | 15.67 | 14.25 | **5.13** |
| Sonar | 20.67 | 21.63 | 24.05 | 25.00 | 24.52 | 22.07 | **18.24** |
| Spambase | 7.22 | 7.33 | 33.89 | 9.25 | 7.48 | 11.46 | **6.74** |
| Spectheart | **18.35** | 18.73 | 20.90 | **18.35** | 20.60 | 24.00 | 20.61 |
| WDBC | 3.34 | 4.04 | 9.66 | 3.69 | **2.64** | 4.56 | **2.64** |
| Chin | 11.86 | **11.02** | 30.53 | 14.41 | 11.86 | - | 14.55 |
| Chowdary | 3.85 | **2.88** | 21.27 | 3.85 | 3.85 | - | 3.00 |
| Gravier | **23.81** | 26.19 | 31.51 | 25.00 | 24.40 | - | 26.80 |
| Gordon | **0.00** | 1.66 | 1.08 | 1.66 | 1.66 | - | 2.22 |
| Singh | **6.86** | 12.75 | 14.91 | 8.82 | 7.84 | - | 13.64 |
| Shipp | 10.39 | 6.49 | 8.75 | 6.49 | **5.19** | - | 21.96 |

logistic regression classification with a lasso that also used the original variables (SPLR). The test error rate is evaluated as a cross-validation error rate similar to the synthetic data analysis. PLDA using `mda` package often failed to produce the result when the dimension of data is larger than 2,000. SVM performs best among all methods when the sample size is larger than the dimension (as observed in the simulation studies). However, PPCLR-L and PPCLR-S show superior performance to other methods when the dimension exceeds the sample size. This indicates PPCLR-L and PPCLR-S can be the best option for high-dimensional binary classification problems (as observed in the synthetic data analysis).

## 4. Conclusion

We propose binary classification methods by selecting principal components in logistic regression classification. Their performance was illustrated and compared with existing methods that used simulated datasets under various situations and several real datasets from various application domains. Such numerical studies confirm that our proposals are competitive, especially for high-dimensional binary classification. We note that our numerical comparisons are not comprehensive because many supervised learning approaches that used principal component analysis (Jolliffe, 2004) and partial least squares (Barker and Rayens, 2003) are not considered and compared in this study.

In this manuscript, we focus exclusively on a binary classification problem, but this idea can be straightforwardly extended to multi-class classification problem as well. For $G$-class classification, the multinomial distribution for class label is appropriate. For $y \in \{1, 2, \ldots, G\}$, the success probabilities of multinomial distribution are

$$p(y = g|\mathbf{X} = \mathbf{x}) = \frac{\exp\left(\alpha_g + \mathbf{z}^T\boldsymbol{\gamma}_g\right)}{1 + \sum_{l=1}^{G-1}\exp\left(\alpha_l + \mathbf{z}^T\boldsymbol{\gamma}_l\right)} \quad \text{for } g = 1, \ldots, G-1,$$

$$p(y = G|\mathbf{X} = \mathbf{x}) = \frac{1}{1 + \sum_{l=1}^{G-1}\exp\left(\alpha_l + \mathbf{z}^T\boldsymbol{\gamma}_l\right)},$$

where $\mathbf{z} = \mathbf{V}^T\mathbf{x}$ and $\mathbf{V}$ is the orthogonal PC loading matrix. The negative log likelihood is $L(\boldsymbol{\alpha}, \boldsymbol{\gamma}) = -\sum_{i=1}^{n}\sum_{g=1}^{G} y_{ig}p(y_{ig} = 1|\mathbf{X} = \mathbf{x}_i)$ with new variable $y_{ig} = 1$ if $y_i = g$ and 0 otherwise. Penalty function becomes $\sum_{g=1}^{G}\mathtt{pen}_{\lambda_g}(\boldsymbol{\gamma}_g)$. This can be easily implemented as described in Section 4.4.5 in Hastie *et al.* (2009) or Section 4.3.4 in Bishop (2006). In the multi-class classification, PC loadings selected by the procedure may reveal the importance of variables in classification between some specific classes; consequently, this is an additional interpretational advantage in classification problems and represents a direction for future research.

## References

Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L. and Herrera, F. (2011). KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, *Journal of Multiple-Valued Logic and Soft Computing*, **17**, 255–287.

Bache, K. and Lichman, M. (2013). UCI machine learning repository [http://archive.ics.uci.edu/ml] Irvine, CA: University of California, School of Information and Computer Science

Barker, M. and Rayens, W. (2003). Partial least squares for discrimination, *Journal of Chemometrics*, **17**, 166–173.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, Springer.

Fan, J. and Li, R. (2005). Variable selection via non concave penalized likelihood and its oracle properties, *Journal of the American Statistical Association*, **98**, 1348–1360.

Friedman, J., Hastie, T. and Tibshirani, R. (2008). Regularization paths for generalized linear models via coordinate descent, *Journal of Statistical Software*, **33**, 1–22.

Hastie, T., Buja, A. and Tibshirani, R. (1995). Penalized discriminant analysis, *The Annals of Statistics*, **23**, 73–102.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Element of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd Edition, Springer.

Jolliffe, I. T. (2004). *Principal Component Analysis*, 2nd Edition, Springer.

Kondylis, A. and Whittaker, J. (2008). Spectral preconditioning of Krylov spaces: combining pls and pc regression, *Computational Statistics & Data Analysis*, **52**, 2588–2603.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*, The MIT Press.

Patterson, N. J., Price, A. L. and Reich, D. (2006). Population structure and eigen-analysis. *PLoS Genetics*, **2:e190**, doi:10.1371.

Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A. and Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies, *Nature Genetics*, **38**, 904–909.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society, Series B*, **58**, 267–288.

*Received October 28, 2013; Revised December 3, 2013; Accepted December 31, 2013*