Project 2 Report

Shengdong Zhang
Department of Computing Science
Simon Fraser University
Burnaby, BC Canada V5A 1S6
sza75@sfu.ca

Abstract

Linear discriminative analysis classifier (LDA) and logistic regression classifiers are only able to give linear decision boundaries and they make no use of the covariance information from data. I think their classification ability can be extended further with the second moment information. However, the phoneme dataset shows that adding covariance matrices to discriminant function easily overfits data and leads to poor generalization performance. In my project, several ways of regularizing sample covariance matrices and their application to extend parametric classifiers were studies and investigated via the phoneme dataset.

1 Why I Chose This Topic

Linear classifiers cannot perfectly separate non-linearly-separable data for the limit of linearity, and Quadratic classifiers are too variable to generalize well. Surely, many non-parametric classifiers, like SVM, gradient boosting trees, random forests, are able to much outperform parametric classifiers like LDA, QDA, naive Bayesian, but they also introduce much more parameters and thus more complexity. In practice, classical parametric classifiers usually have either high bias like LDA, or have high variance like QDA. I hope to extend the classification ability of these parametric classifiers with regularized estimate of covariance matrices from noisy data.

2 Methods Investigated

2.1 Common Principal Components

I happened to see the term "common principal component" when reading a machine learning textbook. I borrowed a book on this subject written by (Flury, 1988) from the library and found it was a good way to reduce variance. Soon I noticed that estimation of common principal components (CPC's) was not easy, so I read a few papers on the parameter estimation of CPC's. Once the CPC's are estimated, I thought we could simply replace the sample covariance matrices in QDA with the CPC estimates to improve its generalization performance. But the QDA discriminant function for each class includes the center(mean) of class and its MLE estimate is not robust. To overcome this problem, I decided to use minimum covariance determinant (MCD) estimator to first get robust estimates of class centers and class sample covariance matrices, and then find the CPCs estimates of the class sample covariance matrices. Before jumping into CPC's, I would like to briefly talk about MCD.

2.1.1 Minimum Covariance Determinant Estimators

Because estimates of second moments of data are sensitive to outliers, instead of using all data for estimation, only α percent of data should be used, where α is the number decided by the user. For the

phoneme dataset, I set this number to 95. However, there are many possible subsets of data which contain α percent of data, the selection criterion is that the one whose covariance matrix estimated by it has the smallest determinant. Since the determinant of a covariance matrix is also known as its generalized variance, this criterion implies that the product of eigenvalues of a sample covariance matrix should be small. Also, using only part of data is reasonable, for outliers will greatly increase the determinants of covariance matrices estimated by the subsets which include them. It can be thought of a regularization technique. There are many available packages in R which implement it, like cov.rob in **MASS**, covMCD in **robustbase**, and I used the first one in my project. Once such subset of data points with the minimum determinant is found, the mean of this subset is used as the robust estimate of the center.

2.1.2 Common Principal Components Model

Let $N = \Sigma_{i=1}^{K} n_i$ be the number of observations, where n_i is number of observations in the ith class; p be the number of variables, and K be the number of classes. Σ_i denotes the covariance matrix of the ith class. Then the common principal components model assumes that:

$$\Sigma_i = U D_i U^T, \forall i = 1, 2, \ldots, K$$

In other words, a CPC model assumes all covariance matrices share the same principal components and the variances along these components are allowed to vary. The assumption of identical eigenstructure is able to reduce variances like commonly used L1 and L2 penalties on parameters. CPC models can be less biased when incorrectly assuming all class covariances matrices are equal, like LDA, and be less variable when incorrectly assuming all class covariances matrices are arbitrary. With this CPC idea, similarities between covariance matrices can be defined.

2.1.3 Similarities Between Covariance Matrices

1. Equality: $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_K$.
2. Proportionality of all Σ_i: $\Sigma_i = \rho_i V, \forall i, \rho_i > 0$ and V is positive definite.
3. Full CPC: $\Sigma_i = U D_i U^T, \forall i = 1, 2, \ldots, K$
4. Partial CPC: $\Sigma_i = [U_q, U_i] \begin{bmatrix} D_i' & 0 \\ 0 & D_i \end{bmatrix} [U_q, U_i]^T, \forall i = 1, 2, \ldots, K$
5. All Σ_i are arbitrary.

These similarities are defined in (Flurry, 1988) page 60. One of these similarities is usually be assumed when modeling data. The assumptions become weaker from top to bottom, so variability of model increases from top to bottom. The first one assumes equality of all covariance matrices, just like LDA; the second one assumes all covariance matrices share the same positive definite matrix V with different coefficients ρ_i; the third one, full CPC, assume all covariance matrix share the same orthogonal matrix and different diagonal matrices; the fourth similarity, partial CPC, assumes all covariance matrices have only q common principal components and p-q private principal components. This similarity assumption is more practical, for it enables us to trade off between bias and variance. The last similarity means no similarity, all covariance matrices are arbitrary, and this is what QDA does.

2.1.4 Estimating Common Principal Components

(Chapter 4. Flury, 1988) Let $\hat{\Sigma}_i$ be the sample covariance matrix of the ith class. For the full CPC model, under the assumption of independently identically distributed data points and Gaussian assumption, we have:

$$n_i \hat{\Sigma}_i \sim W_p(\Sigma_i, n_i), \forall i = 1, \ldots, K$$

where $W_p(\Sigma_i, n_i)$ is Wishart distribution with ni degrees of freedom. So the joint likelihood function of $\Sigma_1, \Sigma_2, \cdots, \Sigma_K$, given $\hat{\Sigma}_1, \hat{\Sigma}_2, \cdots, \hat{\Sigma}_K$ is

$$L(\Sigma_1, \Sigma_2, \cdots, \Sigma_K) = C \prod_{i=1}^{K} exp(tr(-\tfrac{n_i}{2} \Sigma_i^{-1} \hat{\Sigma}_i)) det(\Sigma_i)^{-\frac{n_i}{2}}$$

where C is a constant that does not depend on the parameters. Taking the log of the likelihood function, multiply it with -1 and plugging in the CPC assumption, it becomes an optimization problem with orthogonality constraint:

$$\begin{aligned} U^*, D_1^*, \ldots, D_K^* &= arg\,max L(\Sigma_1, \Sigma_2, \cdots, \Sigma_K) \\ &= arg\,min\, g(U, D_1, \ldots, D_K) \\ s.t. U^T U &= I \end{aligned} \tag{1}$$

where $g(U, D_1, \ldots, D_K) = \Sigma_{i=1}^K (n_i - 1)(log\, det(D_i) + tr(\hat{\Sigma}_i U D_i^{-1} U^T))$.

In the previous subsection, the second similarity is just a special case of the third. After decompose V by spectral decomposition and move ρ_i inside the diagonal matrix, Σ_i can be again written as UD_iU^T. For this reason, the parameters of proportional covariance matrices case can be found by solving the same optimization problem. Adding Lagrange multipliers, we have the following optimization problem:

$$\text{Minimize } Loss(U, D_1, \ldots, D_K) = g(U, D_1, \ldots, D_K) + \vec{1}^T (M \circ (U^T U - I))\vec{1}$$

where $\vec{1}$ indicates a column vector of 1's, $\circ$ denotes entry-wise multiplication, and M is matrix containing all multipliers such that:

$$M = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1p} \\ \mu_{21} & \mu_{22} & & \vdots \\ \vdots & & \ddots & \\ \mu_{p1} & \cdots & & \mu_{pp} \end{bmatrix} \text{ with } \mu_{ij} = \mu_{ji}.$$

This optimization problem can't be solve by gradient-based optimization algorithms except for Newton's method (Wikipedia, 2015). But to use Newton's method, we need to compute its Hessian matrix, which is too hard to get in this case. The reason why usual gradient-based optimization algorithms don't work is because they are able to find local minimum points, but the solution to this problem is at a saddle point. The algorithm for solving this problem has been implemented in the **multigroup** package in R. But in my project, I used other more efficient numerical algorithm proposed by Trendafilov in 2010, called Stepwise Estimation of Common Principal Components, which also has been implemented in **cpcc** package in R. The stepwise estimation is more preferable than the one proposed by Flury in my opinion. The reason is that those CPC's computed by Flury's algorithm show up in arbitrary order, with no emphasis on the variances they explained. The stepwise estimation algorithm applies power method on each group of data to find CPCs iteratively, and power method always converge to the eigenvector corresponding to the largest eigenvalue.

For the partial CPC model having q common principal components, the likelihood function is almost the same, with emphasis on uncommon principal components:

$$\begin{aligned} U_1^*, \ldots, U_K^*, D_1^*, \ldots, D_K^* &= arg\,min\, g(U_1, \ldots, U_K, D_1, \ldots, D_K) \\ s.t.\ U_i^{*T} U_i^* &= I\ \ \forall i = 1, \ldots, K \end{aligned} \tag{2}$$

where $U_i = [U_c\ U_i'] = [\vec{u}_1, \ldots, \vec{u}_q, \vec{u}_{q+1}^{(i)}, \ldots, \vec{u}_p^{(i)}]$, and $\vec{u}_{q+j}^{(i)}$ with $j = 1, \ldots, p-q$ are private principal components of the covariance matrix of ith class.

$$g(U_1, \ldots, U_K, D_1, \ldots, D_K) = \Sigma_{i=1}^K (n_i - 1)(log\, det(D_i) + \Sigma_{j=1}^q \frac{\vec{u}_j^T \hat{\Sigma}_i \vec{u}_j}{\lambda_{ij}} + \Sigma_{j=q+1}^p \frac{\vec{u}_j^{(i)T} \hat{\Sigma}_i \vec{u}_j^{(i)}}{\lambda_{ij}})$$

However, finding the q common principal components is difficult in practice. First, we don't know this q beforehand. One possible way to do this is to try q from 0 to p and compute the likelihood ratio statistic for comparing the CPC(q) model against the unrestricted (all Σ_i's are arbitrary) model, and select the one with the largest p value. (Flury, 1988) proved that for partial CPC model, the true common PCs are included in the ones found by the algorithm for full CPC model. It brings up the second problem: given q, there are p-choose-q combinations of common PCs to try, which could be a large number in practice. Flury provided an approximation algorithm based on maximum likelihood

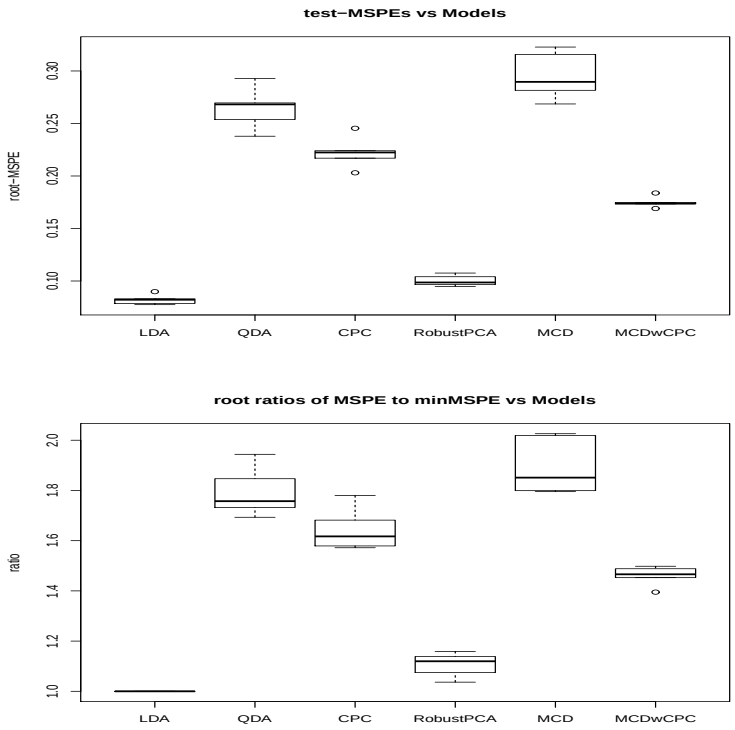
estimation. (Beaghen, 1997) proposed a least square estimation algorithm for it. Unfortunately it has no implementation in R and I did not have enough time to implement it, I didn't use the partial CPC model on the phoneme dataset.

2.1.5 Robust Estimates For Arbitrary Covariance Matrices

When assuming no specific relationship between covariance matrices, the quality of their estimates all depends on the data. QDA makes no such assumption thus any outliers in data can heavily influence its estimates of covariance matrices. (Croux, 2007) proposed an algorithm to estimate robust covariance matrices without any assumption. This algorithm replaces the least square optimization criterion used in classical PCA with other more robust criterion, and combines them with the idea of projection pursuit , resulting in robust covariance estimate. I haven't fully understand Croux's paper but the experiment result of it was not bad at all.

2.1.6 Experiment to Improve QDA

The experiment was performed on the 5 resamples of the phoneme dataset. The estimates of covariance matrices and class centers were computed by the four algorithms mentioned before (Full CPC, RobustPCA, Minimum Covariance Determinant, and CPCs estimated by MCD) on each resample, then replace the sample covariance matrices and class centers in the discriminant functions in QDA with them to get new sets of discriminant functions. Then computed the misclassification rates on the out-of-bag samples. As the MSPE boxplot and the ratio boxplot shown below, RobustPCA, the one gives non-linear decision boundaries, had much better performance than classical QDA. In addition, CPCs estimated from MCD covariance estimates seems better than the CPCs estimated directly from sample covariances matrices. Results of LDA and QDA are also provided for comparison.



2.2 Quadratic Logistic Classifiers

Logistic regression for K classes classification assumes the log odds to be linear e.g. $log\frac{p_k(x;\beta)}{p_K(\vec{x};\beta)} = \beta_{k0} + \vec{\beta}_k^T\vec{x}$ (Notes from Lecture 17). To extend its ability, I added K quadratic forms $\vec{x}^T W_k \vec{x}$ to it, and used the softmax function to define posterior probabilities:

$$p(G=k|X=\vec{x}_j) = \frac{exp(\vec{x}_j^T W_k \vec{x}_j + \vec{\beta_k}^T \vec{x}_j + \beta_{k0})}{\Sigma_{l=1}^K exp(\vec{x}_j^T W_l \vec{x} + \vec{\beta_l}^T \vec{x}_j + \beta_{l0})} = \frac{exp(a_{kj})}{\Sigma_{l=1}^K exp(a_{lj})} = p_{kj}$$

In this case, the log odds become:

$$log\frac{p(G=j|X=\vec{x})}{p(G=k|X=\vec{x})} = \vec{x}^T(W_j - W_k)\vec{x} + (\vec{\beta_j} - \vec{\beta_k})^T\vec{x} + \beta_{j0} - \beta_{k0}$$

It obviously introduce many more parameters than usual K classes logistic classifiers. These W_k's could be thought as covariance matrices. But instead of getting them directly from samples like QDA, we learn them by minimizing the cross entropy function:

$f(W_1, \ldots, W_K, \vec{\beta}_1, \ldots, \vec{\beta}_K, \beta_{10}, \ldots, \beta_{K0})$
$= -L(\{\vec{x}_1 \ldots \vec{x}_N\} | W_1, \ldots, W_K, \vec{\beta}_1, \ldots, \vec{\beta}_K, \beta_{10}, \ldots, \beta_{K0})$
$= -\Sigma_{i=1}^N \Sigma_{j=1}^K I(\vec{x}_i = j) log\, p_{ij}$

where $L(\cdot)$ is the likelihood function. Two ways of regularizing the learning of the W_k's were investigated on the phoneme dataset, and the learning was done using a quasi-Newton method call LBFGS.

2.2.1 L2 penalty

Adding the penalty term to the cross entropy function to penalize large weights for forcing the decision boundaries learned to be not too far away from being flat, we get a new loss function:

$$E = f(W_1, \ldots, W_K, \vec{\beta}_1, \ldots, \vec{\beta}_K, \beta_{10}, \ldots, \beta_{K0}) + \frac{\lambda}{2}\Sigma_{i=1}^K(||W_i||_F^2 + ||\vec{\beta}_i||^2)$$

λ is a hyper-parameter to be tuned.

2.2.2 Additive Gaussian Noise

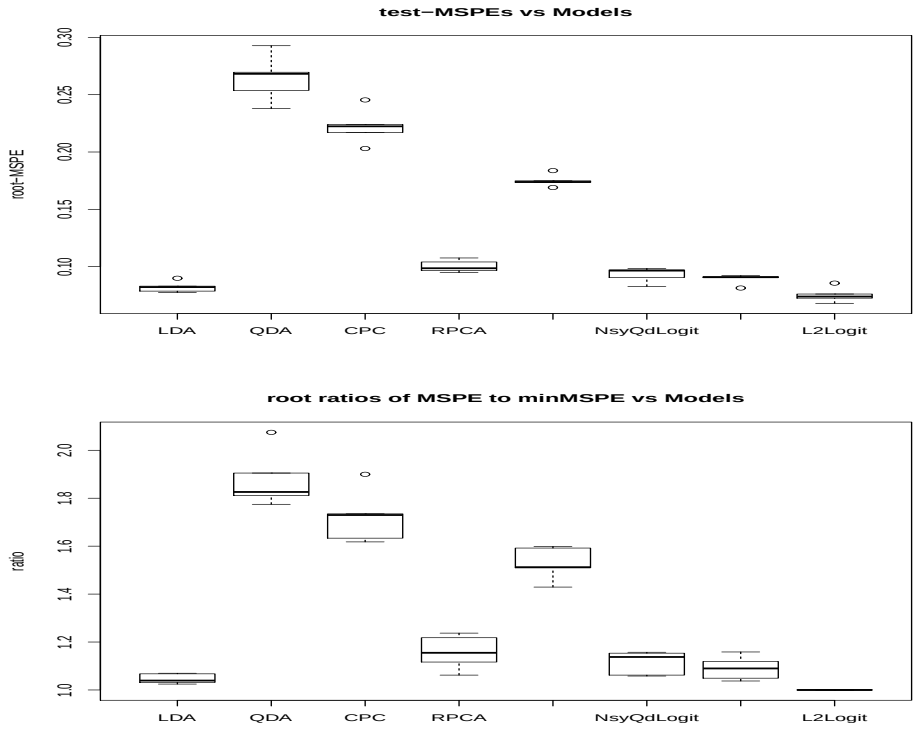
When learning the parameters $W_1, \ldots, W_K, \vec{\beta}_1, \ldots, \vec{\beta}_K, \beta_{10}, \ldots, \beta_{K0}$, in each iteration, add additive Gaussian noise to the data to get the corrupted version of them.

$$\tilde{\vec{x}} = \vec{x} + \vec{\epsilon} \text{ where } \vec{\epsilon} \sim N(\vec{0}, \sigma^2 I)$$

In other words, when training the classifier, it will see different version of corrupted data in each iteration, and we force it to be still able to classify them correctly. However, since different variables have different variances, we need to standardize the data before the training process. Once the classifier is successfully trained, it becomes insensitive to perturbation added to input data and thus more robust. The σ is a hyper-parameter in this case. If data is standardized to have zero mean and unit variance, then this σ can be a small number like 0.1 or 0.2. This number can't be too large, otherwise the training processing can never converge. This idea is borrowed from (Vincent, 2010).

2.2.3 Experiment on Phoneme

On each of the five resamples of phoneme data, the training data were first standardized before training. Both the hyper-parameters λ and σ^2 were manually tuned. Since there are no available packages in R implementing these logistic models and R is not very friendly to matrix algebra, I implemented these models in MATLAB. Also, I implemented a usual logistic classifier with L2 penalty, and it gave the best result. It was even better than the results of LDA and SVM. The MSPE boxplot and the ratio boxplot are shown below. From left to right, the models are: LDA, QDA, CPC, RobustPCA, CPC with MCD, Noisy Quadratic Logistic Classifier, L2 Penalty Quadratic Logistic Classifier, and Logistic Classifier with L2 penalty.



3 Conclusion

Extending linear classifiers to be quadratic with second moment information is a way to generalize them. As long as we learn the second moment information properly, I believe these linear parametric classifiers will give better performance than they did before. For the phoneme dataset, it appears that linear decision boundary can give better classification result. Meanwhile, the quadratic models implemented in my project, except for the CPC model, gave the performance comparable to the best one. Maybe that the covariance matrices share the same eigen-structure is not a good assumption in this case.

In addition, I am so confused that why LDA works on the phoneme dataset so well. I did Bartlett's test on it and its p value is less than $2.2e-16$, which indicates there should be no equality of class covariance matrices. So the only explanation I can give is this dataset favors linear boundaries.

I also investigated how to use information criteria like AIC, BIC, to decide what covariance matrix similarity to use, but due to the time limit I couldn't summarize my result in time. But the AICs I got all suggested me to use the CPC model on the phoneme dataset.

PS: Tom, thank you for teaching me so much stuff during this semester. I got a lot of new ideas inspired by them. I am sorry I didn't do a good job on the presentation. I was nervous for it was the first time I presented in front of that many people. I didn't cover the quadratic logistic regression part in the presentation for it was impossible to go through everything in 10 minutes, so I decided to not talk about it. Right after the presentation, I realized I didn't well present my point of why to regularize covariance matrices and how. I hope this project report can make things clear. Hope, if there is a chance, I can do research with you in the future.

References

[1] Flury (1988). Common Principal Components and Related Multivariate Models

[2] Lecture Note 17 (2015). Loughin, Thomas. Introduction to Classification, Part2

[3] Trendafilov, Nickolay (2010). Stepwise estimation of common principal components. *Computational Statistics and Data Analysis 54 (2010) 3446-3457*

[4] Vincent, Pascal & Larochelle, Hugo etc (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research 11 (2010) 3371-3408*

[5] Petersen, Kaare Brandt & Pedersen, Michael Syskind (2012). The Matrix Cookbook

[6] Croux, Filzmoser, Oliveira (2007). Algorithms for ProjectionPursuit robust principal component analysis. *Chemometrics and Intelligent Laboratory Systems 87 (2007) 218225*

[7] Softmax Regression: http://ufldl.stanford.edu/wiki/index.php/SoftmaxRegression

[8] Wikipedia on Lagrange Multipliers (2015): https://en.wikipedia.org/wiki/Lagrangemultiplier

[9] Beaghen, Michael(1997). Canonical Variate Analysisand Related Methodswith Longitudinal Data