Statistical Learning - Final Project Report

Carlos Perez-Guerra (118033990013),
Department of CSE
Shanghai Jiaotong University
charliewar89@sjtu.edu.cn

Fig. 1. Screenshot of my best prediction set in the Kaggle competition, as required by the task description

***Abstract*—In this report, I examine the different classification algorithms we discussed in class as applied to the image training dataset Specifically, I will focus on discriminative supervised classification algorithms, and then enhance their power through the semi-supervised learning technique known as pseudo-labeling. The best algorithm achieved a private score of 0.93891 (top 5%, Fig. 1), although it was a late submission, hence validating the methods described herewith.**

Introduction & Methodology

In keeping with assignment guidelines, my choice of algorithms drew from class discussion. The R programming language was chosen for this task since all algorithms discussed had an implementation by their creators or affiliated researchers in this language.

Due to the high dimension of the data, PCA was performed and used as a base for the input space, and hence the number of components was treated as an additional parameter. The number of components retained is denoted with the letter p. All parameters for the algorithms in Table 1 were estimated using 5-fold, 2-repetition cross validation. Cross-Validation was chosen over Bootstrap Estimation due to the shape of the data, in which $N_k < p$. Initially, my investigation focused on supervised learning techniques, and the results of the most effective algorithms are summarised in Table 1.

However, it soon became clear that further generalization power could be achieved by considering the unlabeled data set to impose constraints on the dataset, thereby reducing the estimator's bias (semi-supervised learning).Using the pseudo-labeling algorithm, which I implemented in R and which is explained in detail below, resulted in an improvedment in generalization ability for all algorithms. Due to the nature of the pseudo-labeling algorithm (training error increases with each iteration as bias is smoothed out), cross-validation was not used. Results using pseudo-labeling are summarised in Table 2.

TABLE I
Results using some classic supervised learning algorithms

Algorithm	Parameters	Accuracy (Training, 95% CI)	Accuracy (Private Leaderboard)
HDRDA Ridge	$\lambda = 0.5$, $\gamma = 0.5$	0.98339 (0.9775, 0.98924)	0.91346
HDRDA Convex	$\lambda = 0.5$, $\gamma = 0.5$	0.98122 (0.9715, 0.9893)	0.912471
PLSDA	$n = 15$, $p = 1000$	0.98122 (0.9715, 0.9893)	0.92655
Logistic Regression		0.98122 (0.9715, 0.9893)	0.92115
LDA		0.98122 (0.9715, 0.9893)	0.90576
MDA	$n = 10$	0.98122 (0.9715, 0.9893)	0.88791
SVM - Linear		0.98122 (0.9715, 0.9893)	0.85778

TABLE II
Results using pseudo-labeling

Algorithm	Parameters	Accuracy (Private Leaderboard)
PLSDA	$n = 15, p = 1500$	0.92902
Least Squares Classifier	$\lambda = 0.1, p = 1500$	**0.93891**

Exploratory Analysis

A first glance at our data set reveals we have almost as many features as we have data points. There are 12 classes is the training set (balanced), and each predictor is real-valued, so each class is severely undersampled. Hence we should treat this as a supervised dimensionality problem, namely we wish to find a transformation that maps data to a lower dimensional space while maintaining or improving separability of the data.

A common technique in dimensionality reduction is to start by doing Principal Component Analysis. PCA is an unsupervised training technique that consists on doing SVD on the centered and scaled predictors covariance matrix. This is equivalent to projecting the input space on a new space whose basis is the set of eigenvectors, where the resulting magnitude of the eigenvalues reveals how much variance of the data is explained by each eigenvector in the new space. PCA is an expensive algorithm, but it is feasible for the size of our

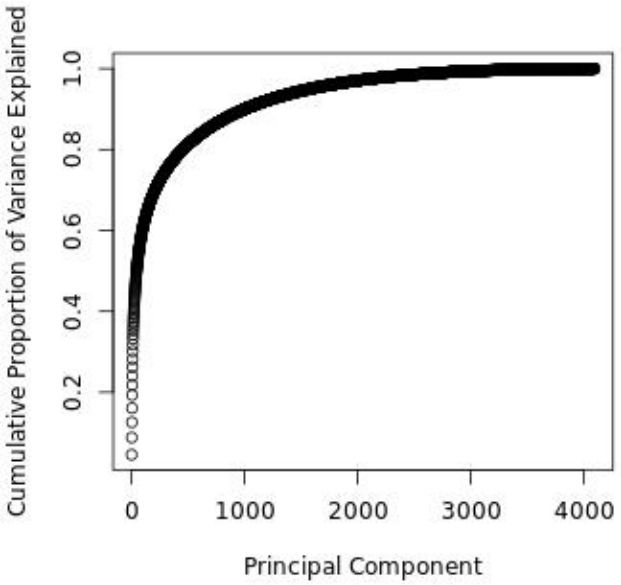

Fig. 2. Scree plot reveals most components are highly correlated. The first 1000 components explain 90% of variance in the data

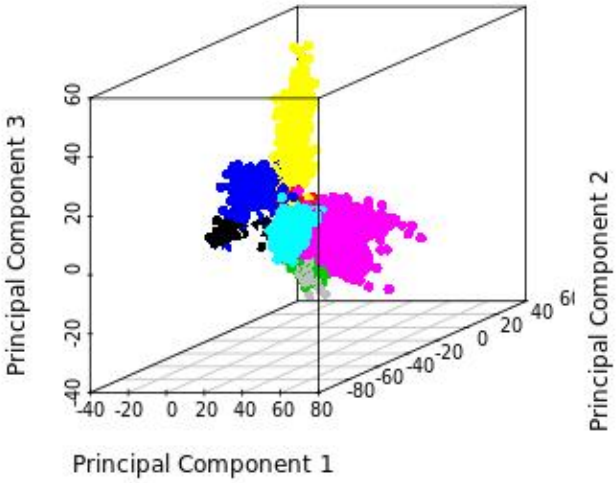

Fig. 3. A 3D scatterplot of the inputs in PC space using the first 3 components.

dataset (especially since there are parallel implementations). The results are summarised in Figs. 2,3, 4 and Table 1:

TABLE III
PERCENTAGE OF VARIANCE EXPLAINED AS NUMBER OF PRINCIPAL COMPONENTS INCREASES

# PC's	% of Variance explained
3	13%
8	26%
50	52%
425	79%
1000	90%
1500	94%
4094	100%

PCA reveals two predictors are redundant, and most of the variance is explained with 2000 components. It is very likely that most of the near-zero components are simply perturbations or noise. Somewhere between 1000 and 2000 components there exist optima, where variance (resp. separability) is preserved and dimensionality is reduced.

DISCRIMINANT ANALYSIS

There exist many discriminant analysis approaches, all derived from the maximum likelihood estimate of log odds between two classes of data assumed to be normally distributed. As required, I will describe briefly their origin and formulation:

Recall that the MLE of samples from a normal distribution is given by:

$$\hat{\pi}_k = \frac{N_k}{N} \tag{1}$$

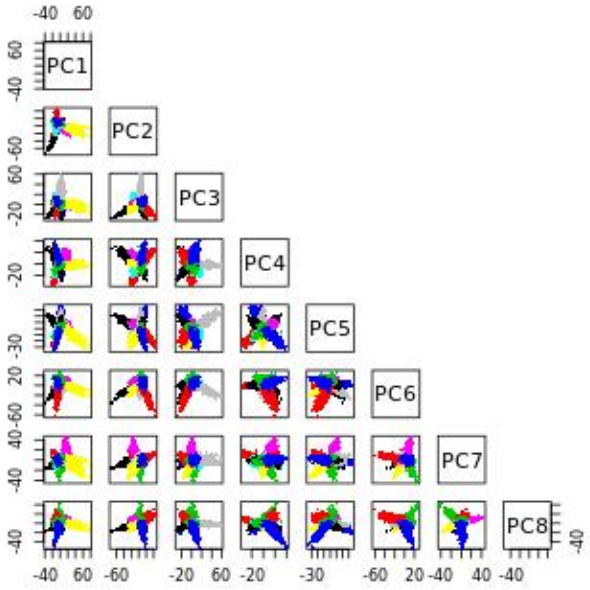

Fig. 4. Pair scatterplots of the first 8 components (which explain 26% of the variance)

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N_k} x_n \tag{2}$$

$$\hat{\Sigma}_k = \sum_{n=1}^{N_k} (x_n - \hat{\mu}_k)(x_n - \hat{\mu}_k)^T \tag{3}$$

We can express the odds using Bayes' Rule:

$$\frac{\hat{\pi}_k \hat{f}_k}{\hat{\pi}_j \hat{f}_j} = 1 \tag{4}$$

where the equality of numerator and denominator will occur at the class boundary. We can further assume that the classes are multivariate Gaussian. Taking logs and doing some algebra we get the QDA discriminant:

$$\delta^{(k)}_{QDA} = \log \hat{\pi}_k - \frac{1}{2}[log|\hat{\Sigma}_k| + (x_n - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (x_n - \hat{\mu}_k)] \tag{5}$$

The LDA discriminant follows by allowing the covariance matrix to be a pooled covariance (weighted sum of the covariance matrices of all classes). Then terms that don't depend on $\hat{\mu}_k$ cancel out and we have:

$$\delta^{(k)}_{LDA} = \log \hat{\pi}_k + \hat{\mu}_k^T \hat{\Sigma}^{-1} x - \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k \tag{6}$$

Friedman (1989) proposed Regularized Discriminant Analysis, which is essentially a refinement of QDA in which the pooled covariance matrix $\hat{\Sigma}$ is used to smooth $\hat{\Sigma}_k$ by a parameter λ, hence the new covariance matrix is now a convex combination:

$$\tilde{\Sigma}_k(\lambda) = \lambda \hat{\Sigma}_k - (1 - \lambda)\hat{\Sigma} \tag{7}$$

Lastly, Ramey et al. (2016) proposed HDRDA, whose formulation is:

$$\tilde{\Sigma}^{(k)}_{HDRDA}(\lambda, \gamma) = \alpha \tilde{\Sigma}_k(\lambda) - \gamma I \tag{8}$$

In this case if $\alpha = 1$ then we have Friedman's RDA with a shrinkage term on the eigenvectors, which the authors define as *HDRDA Ridge*, while if $\alpha = 1 - \gamma$ we have a different algorithm which they define as *HDRDA Convex*.

Hence it is clear that we can do a search on λ and γ and it is equivalent to testing the data on LDA,QDA and RDA, while also accounting for variance in high-dimensional space with shrinkage. The cross-validated optimum of the Accuracy error surface was found to be $\lambda = 0.5, \gamma = 0.5, \alpha = 1 - \gamma$. The $sparsediscrim$ package (written by the authors of the paper) was used.

Mixture Discriminant Analysis

It is clear that while HDRDA performs well on the data, there is still room for improvement. Could the error be reduced by fitting a non-linear model? The logical extension is MDA, first proposed by Hastie and Tibshirani (1996), which attempts to model each class as a Gaussian Mixture Model instead of as a single Gaussian. However this model performed poorly, as can be seen in Table 1 and Fig. 6.

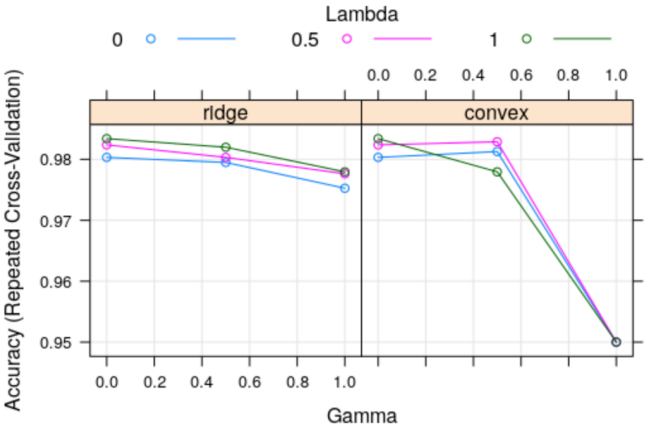

Fig. 5. Parameter tuning for the HDRDA classifier)

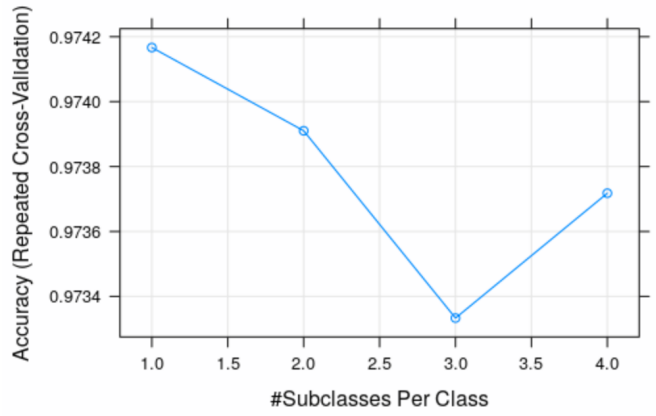

Fig. 6. Accuracy clearly decreases with subclasses per class for the MDA classifier.

Partial Least Squares Discriminant Analysis (PLSDA)

Partial Least-Squares Discriminant Analysis (PLS-DA) is a supervised multivariate dimensionality-reduction tool [15], [2] that has been popular in the field of chemometrics for well over two decades (Gottfries et al. (1995), Ståhle and Wold (1987), Perez and Narasimhan (2018)). Chemometrics data sets are characterized by large volume, large number of features, noise and missing data (Barker and Rayens (2003)). Because of the difficulty in obtaining labeled data, and the large number of features, usually $n_k < p$, as in our case.

PLSDA can be thought of as the supervised counterpart of PCA is a different approach to dimensionality reduction. In contrast to PCA, PLSDA considers the correlation of each predictor with the labeled data. It is similar to Fisher Discriminant Analalysis, but allows for a feature space that is larger than $k - 1$ dimensions, and is hence able to capture more nuanced data patterns.

The way it works is not mysterious. In the two-class case, the labels are encoded numerically as {0,1}, then the

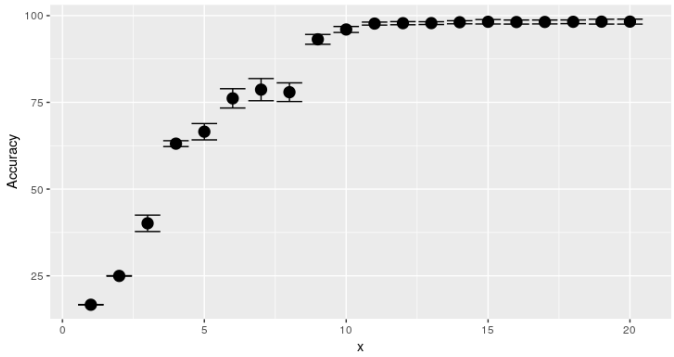

Fig. 7. PLS-DA Accuracy with 95% CI as number of components is increased. The optimum is found to be 15 components.

classic partial least squares algorithm is applied (Algorithm 1). The only parameter that requires tuning is the number of components in the new space.

input : X, y, k
output: $[\phi]_{\dim(X)\times k}$
$k \triangleq$ Desired number of components
begin
  /* Center and normalize each predictor to have zero mean and unit variance */
end
for $m = 1, 2, .., k$ **do**
  $\phi_{mj} = x_j^{(m-1)} \cdot y$;
  $z_m = \sum_{j=1}^{p} \phi_{mj} x_j$;
  $\hat{\theta}_m = \frac{z_m \cdot y}{z_m \cdot z_m}$;
  Orthogonalize each $x_j^{(m-1)}$ w.r.t z_m
end

Algorithm 1: The PLS Algorithm

In the multiclass case, the two-class case is extended by means of an indicator matrix Y. Then, the two-class algorithm is applied to each column of the matrix Y, and then k transformation matrices are computed. The discriminant is then calculated using softmax criteria on each new space.

The PLSDA algorithm enjoys widespread use for image analysis in the Chemometrics community (Chevallier et al. (2006)).

In my comparative study, PLSDA had the greatest generalization ability out of all supervised algorithms.

Other algorithms

SVMs performed poorly on the dataset. Both Linear and RBF kernels were employed using hinge loss, the results were significantly lower than other methods discussed in this report. This is somewhat surprising considering SVMs are more robust when dealing with noisy data, as the boundaries only depend on the support vectors.

Furthermore, generally speaking, nonlinear models performed poorly on the data set. MDA has already been mentioned, Neural Nets were also attempted with disappointing results.

Pseudo-labeling

Pseudo-labeling, also known as self-training or Yarowsky's Algorithm, is a commonly used technique for semi-supervised learning (Algorithm 2). A classifier is first trained with the small amount of labeled data. The classifier is then used to classify the unlabeled data. The classifier is re-trained and the procedure repeated, namely the classifier uses its own predictions to teach itself (Zhu (2006)).

input : classifier$(\cdot,\cdot)$, $y^{(0)}, X^{(0)}, X^{(u)}$, max_iter
output: model
$y^{(0)} \triangleq$ Labels vector (labeled set);
$X^{(0)} \triangleq$ Predictor matrix (labeled set);
$X^{(u)} \triangleq$ Predictor matrix (unlabeled set);
begin
  /* We first train a model using the classifier function and label the unlabeled set */
  model := classifier$(X^{(0)}, y^{(0)})$;
  $\hat{y}^{(u)}$:= model.predict$(X^{(u)})$;
end
while $j < maxiter$ **do**
  $j = j + 1$;
  $X := \begin{bmatrix} X^{(0)} \\ X^{(u)} \end{bmatrix}$;
  $\hat{y} := \begin{bmatrix} y^{(0)} \\ \hat{y}^{(u)} \end{bmatrix}$;
  model := classifier$(X, \hat{y})$;
  $\hat{y}^{(u)}$:= model.predict$(X^{(u)})$;
end

Algorithm 2: The Pseudo-Labeling Algorithm

Pseudo-labeling is a wrapper algorithm, and is hard to analyze in general. However Abney (2004) provides an analytic proof of how the algorithm essentially "optimizes either likelihood or a closely related objective function K". Like many other semi-supervised learning techniques, it is in common use in fields where labeled data is scarce with respect to test data. In my review of the literature I found that test set sizes are always significantly larger than training sets, and my own experiments showed that pseudo labeling doesn't have a strong impact when the test set is similar in size or smaller than the training set. However, in our case the test set is significantly larger than the training set, and pseudo-labeling was shown to significantly improve accuracy on the test set.

PLSDA showed a marked improvement over it's supervised counterpart, as seen in Table 2.

However the algorithm that performed the best out of all is Least Squares Classifier with L2 regularization. This algorithm is equivalent to assigning each class a numeric value, then performing Ridge Regression on the data set. To predict new values, the weights are applied as in Linear Regression and the class value closest to $\hat{y}_i$ is the predicted class. A grid

search was performed on $p = 1000 : 2000$ in steps of 100, and $lambda = 0 : 1$ in steps of 0.1.

Conclusion, Future Research and Personal Thoughts

The results show that for a data set of these characteristics, namely, small proportion of labeled data points, high dimensionality, less samples per class than features ($n_k < p$), noise, and collinearity, we can use the sort of techniques that are used in Chemometrics and Biostatistics, where many data sets are of this nature. Discriminant Analysis, PLSDA, logistic regression -the fundamental tools of Statistical Learning Theory - all yielded good results.

However, it is clear that semi-supervised learning techniques are very useful in dealing with scenarios where unlabeled data far exceeds labeled data. Using pseudo-labeling always reduced the generalization error. If I had more time I would have attempted other classifiers in conjunction with pseudo-labeling, for example SVMs. Furthermore, it is clear that there are many semi-supervised learning techniques which I should have explored but didn't, so these present are a clear opportunities for improvement.

One drawback of semi-supervised learning, though, is the inherently serial nature of many of its algorithms. Most algorithms are very much like the EM algorithm family or the Pseudo-labeling algorithm, in that they iterate until convergence using results from the previous iteration, and hence don't lend themselves to parallel implementations. This increase in running time is compounded by the fact that semi-supervised learning uses a lot more data (several times more). Having said that, one can still run many different algorithms, each using one core, but this would essentially require containers, or look for a parallel implementation of the base classifier if it exists - but these pose a greater technical challenge for the average practitioner.

It is clear that a lot of the properties of these algorithms have still to be researched. A clear example is how pseudo-labeling, which is a very simple algorithm, is generally not well understood. For my personal work, I observed that training error increased with each iteration of the pseudo-labeling algorithm, but it is not rigourously proven whether there is a lower bound, and hence convergence, and whether this convergence is some sort of optima, or whether the algorithm eventually also overfits and misses its optimum.

Bibliography

Abney, Steven. 2004. "Understanding the Yarowsky Algorithm." *Computational Linguistics* 30 (3): 365–95.

Barker, Matthew, and William Rayens. 2003. "Partial Least Squares for Discrimination." *Journal of Chemometrics: A Journal of the Chemometrics Society* 17 (3): 166–73.

Chevallier, Sylvie, Dominique Bertrand, Achim Kohler, and Philippe Courcoux. 2006. "Application of Pls-Da in Multivariate Image Analysis." *Journal of Chemometrics: A Journal of the Chemometrics Society* 20 (5): 221–29.

Friedman, Jerome H. 1989. "Regularized Discriminant Analysis." *Journal of the American Statistical Association* 84 (405): 165–75.

Gottfries, Johan, Kaj Blennow, Anders Wallin, and CG Gottfries. 1995. "Diagnosis of Dementias Using Partial Least Squares Discriminant Analysis." *Dementia and Geriatric Cognitive Disorders* 6 (2): 83–88.

Hastie, Trevor, and Robert Tibshirani. 1996. "Discriminant Analysis by Gaussian Mixtures." *Journal of the Royal Statistical Society. Series B (Methodological)*, 155–76.

Perez, Daniel Ruiz, and Giri Narasimhan. 2018. "So You Think You Can Pls-Da?" *bioRxiv*. https://doi.org/10.1101/207225.

Ramey, John A, Caleb K Stein, Phil D Young, and Dean M Young. 2016. "High-Dimensional Regularized Discriminant Analysis." *arXiv Preprint arXiv:1602.01182*.

Ståhle, Lars, and Svante Wold. 1987. "Partial Least Squares Analysis with Cross-Validation for the Two-Class Problem: A Monte Carlo Study." *Journal of Chemometrics* 1 (3): 185–96.

Zhu, Xiaojin. 2006. "Semi-Supervised Learning Literature Survey." *Computer Science, University of Wisconsin-Madison* 2 (3): 4.