literature. Quite often due to economic reasons, the tolerable error rate in industrial applications can be very stringent. This rate might not be easy to achieve when applied to all items to be tested. However, it can be expected that a good portion of them may be easier to classify than others. Thus it is useful to find a simple tool that can help locate items in such a higher-grade subpopulation. Intuitively, the group of items which are classified under an unanimous decision should have a lower rate of prediction error. Thus they are the natural candidates to be in the higher-grade subpopulation.

We shall use hand-written digit recognition as an example to illustrate the advantages of our approach. The data base consists of digitized images of zip codes on envelopes passing through Buffalo, NY. (LeCun et al., 1989). Using a conditional error rate analysis, we can identify a large portion of sample images in the test set that can be classified under simple linear classification rules at a misclassification rate less than 1%.

In section 2, we first give a brief account of the zip code data. A centre-of-mass based method for feature extraction is introduced, to be followed by a preliminary analysis using linear discriminant rules. Section 3 describes details of the three-way subclassification settings and discuss how the results can be combined using conditional error analysis. Further discussions are given in Section 4. Here we compare the three-way subclassification with binary subclassification. We introduce another way of partitioning for extracting feature variables from the zip code data. With this new feature space, we show how to combine the two-way and the three-way methods together to get better results. Section 5 gives some concluding remarks.

15.2 Data, features, and LDA.

In this section, we first describe the handwritten data set which will be used throughout this article. Then we introduce a method for extracting features that reduce the dimensionality to a level easier to manage. A preliminary analysis involving linear discriminant analysis (LDA) is also reported.

15.2.1 The zip code data.

Our data base comes from handwritten zip codes that appeared on some envelopes of U.S. mail passing through the Buffalo, NY post office. The digits were written by many different people with a great variety of writing styles and instruments. Each digit is converted into a 16 by 16 pixel image after some preprocessing as described in LeCun et al. (1989). Figure 2.1 shows some of these normalized images.

The seminal work of LeCun et al. uses a neural network with three hidden layers- 768, 192, and 30 hidden units respectively for each layer. A misclassification rate of 0.14% on the training data and 5.0% for the test set were reported. This remarkably low rate of error cannot be achieved without clever ideas and deliberated efforts on setting up clever connection architecture and contrains on weight constraints. For example, the same authors also reported that a fully connected network with one hidden layer of 40 nodes yields 8.1% error rate on the test data. It is also worth mentioning that since backpropagation is used, the

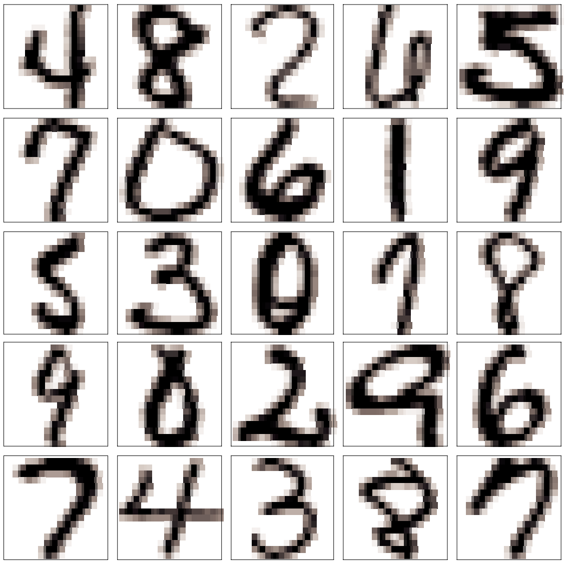


Figure 2.1: Sample Hand-Written Digits

connection coefficients are updated sequentially as each image pattern in the training set is presented. A total of m passes through the training set means 7291m pattern presentations, and the coefficients are updated accordingly. The 5% rate for test set is obtained when the number of passes is 23 (which means 23 times 7291 = 167,693 updates). Figure 2 of LeCun et al. shows how this error rate depends on the number of passes. Although there is no clear stopping rule, it is noticed that for a wide range, between 5 and 30, the error rates are falling between 5% and 6% - reaching about 5.5% at $m = 30$.

This data set has since received a great deal of academic attention. Their error rates have become the bench-marks for comparison. There are several attempts aiming directly at improving the error rates for optical digit recognition. The best improvement reported in the literature seems to be 2.5% by Simard, LeCun and Denker (1993) who use a transformation based nearest neighbor method. The transformations intend to incorporate various kinds of distortion due to factors such as shifting, scaling, rotating, and so on. Unfortunately, to implement such a discriminant rule, it requires a heavy amount of computation.

This data set is also often used as an illustration for new all-purpose classification methods which may not be specially tailored for digit recognition. In such cases, the rates are usually poorer than the 5% to 6% range. For example, using a penalized discriminant rule, Hastie, Buja, and Tibshirani (1995), a rate of 8.2% is obtained, which is an improvement over the rate of about 10% by the usual LDA.

15.2.2 Centre-of-mass-based partition.

Due to the spatial arrangement, the 256 variables representing the 16 by 16 image for each sample character are highly correlated. Such redundancy among the input variables allows a certain degree of flexibility in designing strategies for feature extraction.

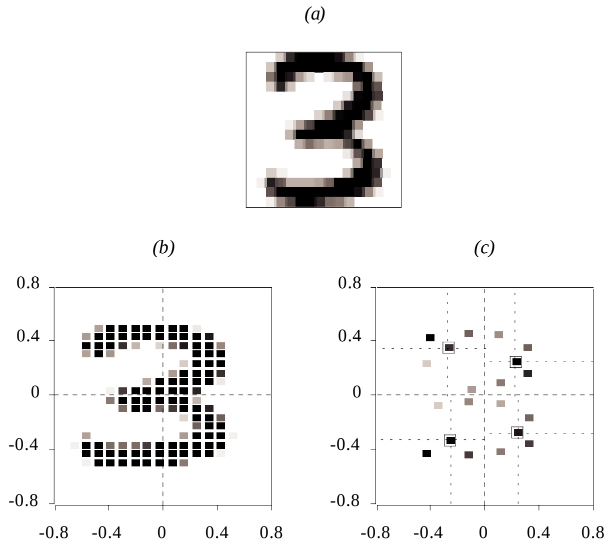


Figure 2.2: Locations of mass Centers for Digit 3: (a). Orogonal Image; (b). Transformed Digit; (c). Locations of mass Centers with Weights.

The method we use treats each character image as a piece of object with mass at each pixel (i, j) equals to its grey level w_{ij}. For each object, we first compute the centre of mass (C_{01}, C_{02})= $(\sum w^*_{ij} i, \sum w^*_{ij} j)$ where $w^*_{ij} = w_{ij} / \sum w_{ij}$ is the total mass of the object. This mass center is then used to normalize the image by shifting (C_{01}, C_{02}) to the origin $(0, 0)$. The range of the two coordinate axes are scaled to take values within -1 and 1 , using grey levels greater than a prespecified threshold value as a common denominator. Figure 2.2(b) gives a transformed digit 3 whose original image is in Figure 2.2(a). The entire image is now partitioned into four pieces, one piece from each quadrant.

The next step is to calculate the mass center for the image in each quadrant. This generates 4 pairs of locational variables. After that, each quadrant is again partitioned into four quadrants, using the mass center calculated earlier as the origin. This yields a total of 16 rectangle regions and the mass center of the image in each rectangle is computed. The 20 locations of the mass centers for the digit 3 in Figure 2.2(b) are shown in Figure 2.2(c). Our feature space consists of these 40 locational variables and the 16 weight variables for the total mass in each of the final 16 rectangles. Thus the original 256 grey-level variables is reduced to 56 feature variables.

Quadrants are certainly not the only choices we can use in our centre of mass based partition. Later on a system of 8 radial lines as described in Section 4.2. will be used, which leads to better results.

In LeCun et al. (1989), there are 7291 cases in the training set and 2007 cases in the test set. We use only 7188 and 1991 respectively as our training and test sets - this will be referred to as 7188/1991 data. Other cases are deleted for the moment because they have no mass in some of the final 16 rectangles. This problem can be corrected if we use the radial partition system - details to be discussed later.

It is not clear how the original 2007 test cases were selected in LeCun et al.. Typically, if the test set comes from the same population as the the training set. then it should exhibit characteristics similar to any randomly selected subsample of equal size from the training set. However, this is not the case here. Several authors have reported an unexpected increase in error when applied to LeCun et al.'s test set. For comparison, these authors often generate two randomly selected subsets of size 2000 each from the 7291 training set, one as the new training set and the other as the new test set. They find that the error rates are smaller for the new test set than the original size-2007 test set. We shall follow this practice and generate a 2000/2000 data - 2000 cases for the training set and 2000 cases for the test, both being randomly selected from the 7188 cases.

15.2.3 A preliminary analysis.

We first apply linear discriminant analysis (LDA) to the 2000/2000 data. The error rate is 5.75% for the training set and 8.3% for the test set. We then apply LDA to the 1991/7188 data. As expected, the result is even worse, 7.2% for the training set and 10.7% for the test set.

We take a closer look at the training set of 2000/2000 data. The first three canonical variates from LDA are shown in Figure 2.3. Noticeable clustering patterns can be found. For example, a good portion of digit 0 visually separable from other digits is found in the lower left corner of Figure 2.3(a). In the same figure, digit 1 appears to occupy in another corner.

We are led to the suggestion of isolating these two clusters from the rest of data because they appear easier to classify than others. To do so, we can formulate a three-way classification problem by considering ”0” as one class, ”1” as another class, and pooling all other digits together as a third class called ”Others”. We first reduce the dimension of the feature space from 56 to 2 using the canonical variates from LDA. Figure 2.4 shows the scatterplot of the two canonical variates. The clustering pattern is more clear-cut than what is seen in Figure 2.3. Since we are interested in isolating cases that can be predicted with higher precision, the boundaries of discriminant regions for each class are pushed a bit toward the centers of 0 and 1 - this is done by setting the prior probability of 0 and 1 to be .0005 each. After we isolate these two clusters with mostly 0 and 1, we proceed by considering another three-way classification for the class ”Others”. This time we choose digits 6 and 7 as two classes and pooling all other digits (including 0 and 1) together as ”Others”. We continue to partition the left-over ”Others” till we obtain a three-way classification tree as shown in Figure 2.5.

From the tree we obtained and the scatterplots we generated along the analysis, we see that a good portion of data forms distinctive clusters which can be isolated in an iterative fashion. Out of the 2000 test cases, about a half of them can be classified with 1% of error. What should we do with the left-over cases? One simple suggestion is to apply the linear discriminant analysis. Another possibility is to repeat the same procedure again before linear discriminant analysis is applied. The result is given in Figure 2.6.

One problem with this three-way tree approach is how to decide the ordering of partition.

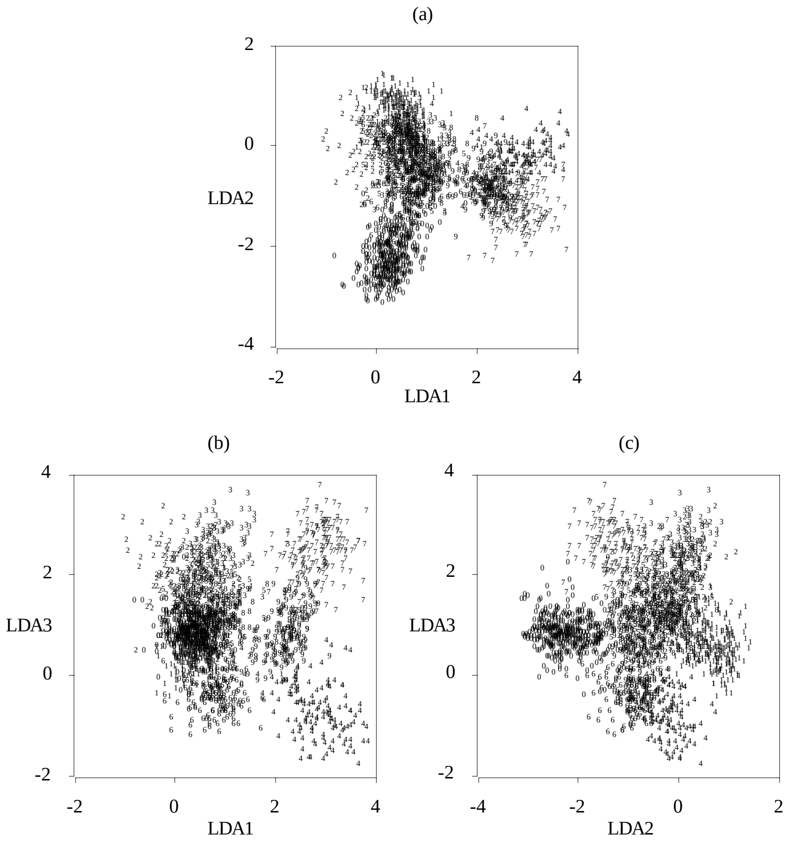


Figure 2.3: LDA Cannonical Variates: (a). LDA1 versus LDA2; (b). LDA1 versus LDA3; (c). LDA2 versus LDA3.

Why 0 and 1 are chosen first? There are certainly many criteria that can be used. We simply choose the pair which gives the smallest error rate for the training data. Another problem is the choice of prior in determining how much the lines should be pushed away from the center of the ”Others”. This is another optimization problem, which needs to be resolved.

While the above line of thoughts may be worth pursuing further, in this article we shall alter our strategy a bit in order to bypass such difficult optimization decisions. There is no need to generate three-way trees. Instead, we shall synthesize results from all three-way classifications in a way to be discussed in the next section.

15.3 Three-way subclassification.

We begin with a general description of our three-way subclassification method. Suppose there are in total k groups under consideration. For each pair of groups, i and j, we formulate a three-group problem - group i , group j , and a third group which is the union of all other groups. Suppose from the training data, a classifier denoted by T_{ij}, is obtained. For any

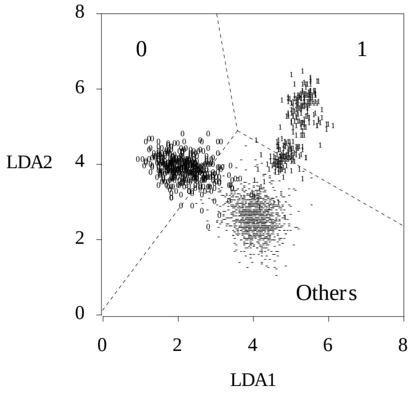
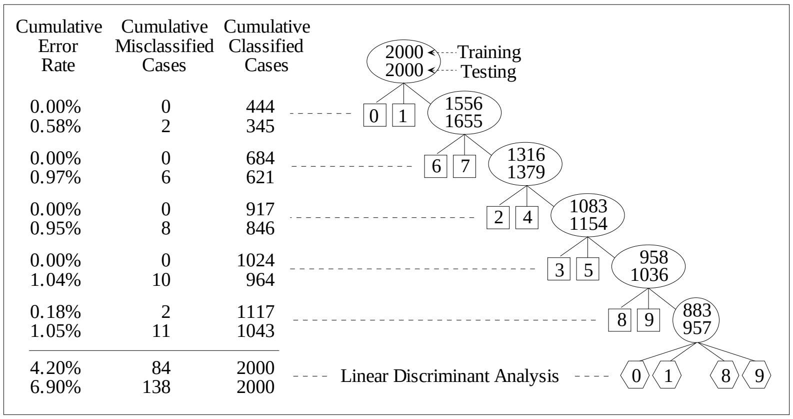


Figure 2.4: LDA Cannonical Variates for Group 0, 1 and Others(-).



Cumulative Error Rate	Cumulative Misclassified Cases	Cumulative Classified Cases
0.00% 0.58%	0 2	444 345
0.00% 0.97%	0 6	684 621
0.00% 0.95%	0 8	917 846
0.00% 1.04%	0 10	1024 964
0.18% 1.05%	2 11	1117 1043
4.20% 6.90%	84 138	2000 2000

Figure 2.5: Three-Way Classification Tree for the Hand-Written Digit Recognition Problem.

unit with feature **x**, the classifier assigns it a membership $T_{ij}(\mathbf{x}) = i, j$ or $Others$ with $Others$ standing for the third group. Subclassification is carried out for all of the $k(k-1)/2$ three-group problems. At the end, for each **x**, we can have a total of $k(k-1)/2$ values, $T_{ij}(x), 1 \leq i < j \leq k$.

For the digit data, there are $k = 10$ classes, yielding a total of 45 subclassification problems. For each problem, we consider a linear partition rule with a simple tree structure (Figure 3.1). The tree consists of two levels of partitions - the first one is to divide the training set into two nodes $i/Others$ and $j/Others$ along the projection that best separates class i and class j. Node $i/Others$ is expected to contain most members from digit i and a good portion of members from digits other than i and j. This node is further divided into two children nodes, i and $Others$ again by a best linear partition rule. The other node $j/Others$ is similarly partitioned into two nodes, j and $Others$. There are certainly quite a few alternative partition rules worth trying. But since our main focus in this article is on the strategy

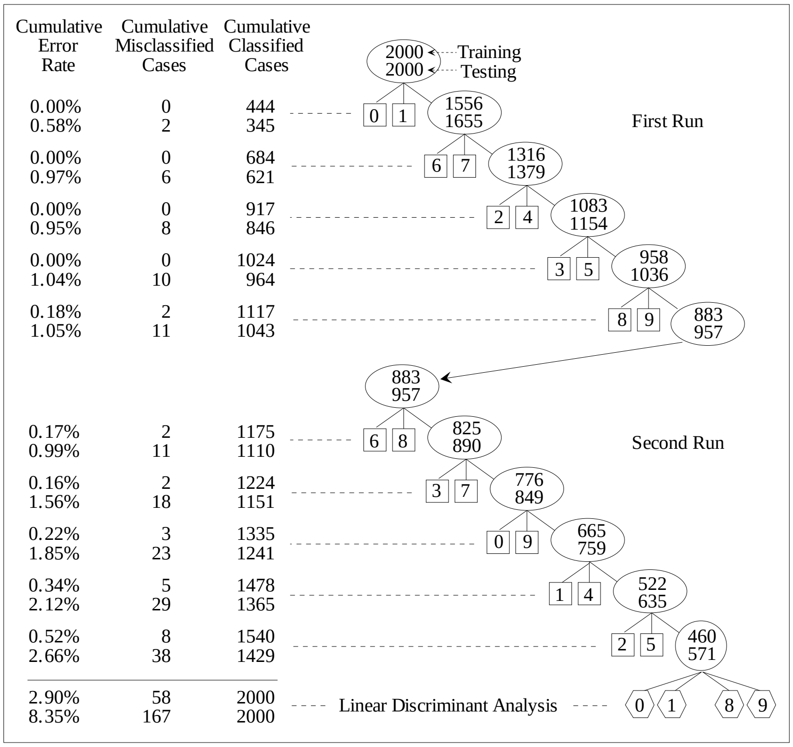


Figure 2.6: Repeated Three-Way Classification Tree for the Hand-Written Digit Recognition Problem.

of three-way partition itself, to avoid further distraction, we shall not make such attempts here.

15.3.1 Majority rule.

To combine the results from each subclassification, we first count the frequency that each class is assigned. For any fixed unit $\mathbf{x}$, we obtain a k-dimensional vector $(c_1(\mathbf{x}), \cdots, c_k(\mathbf{x}))'$ where for each l between 1 and k, $c_l(\mathbf{x})$ equals the total number of times that $T_{ij}(\mathbf{x})$, $1 \leq i < j \leq k$, equals l. We can think of each classifier T_{ij} as assigning the winner for a match between players i and j, but with possibility for a tie in which $T_{ij}(\mathbf{x})$ equals Others. The vector $(c_1(\mathbf{x}), \cdots, c_k(\mathbf{x}))'$ is just the score board showing the winning record for each player. Then it should be clear that the largest value that $c_l(\mathbf{x})$ can take is $k - 1$ because that is the total number of matches that class l has participated. We can rearrange the score board in a non-increasing order, $M_1(\mathbf{x}) \geq M_2(\mathbf{x}) \geq \cdots \geq M_k(\mathbf{x})$.

Consider the majority rule for the final class membership assignment- the class winning the most is assigned. We may classify unit $\mathbf{x}$ into the class l achieving $c_l(x) = M_1(\mathbf{x})$. To keep the procedure simple, if necessary, randomization can be used as a tie-breaker.

We apply the majority rule to the 2000/2000 data and the 7188/1991 data. The results are

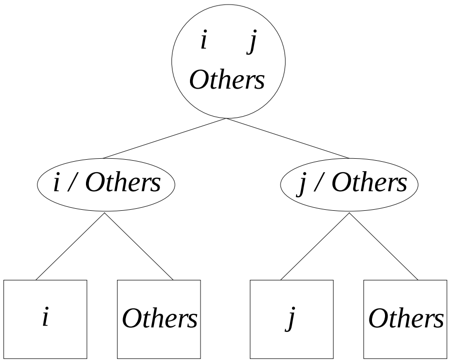


Figure 3.1: Tree-Structure for Three-Way Subclassification.

summarized by two misclassification matrices in each case - one for the training set and the other for the test set; Tables 3.1(a)-(b) and 3.2(a)-(b). A cell in each matrix shows the number of times a digit in the beginning of a row is misclassified as another digit on the top of the corresponding column. For example, take a look at the row with the digit 2 in the training set of 7188/1991 data. We see that 2 is misclassified as 0 two times, as 1 three times, and so on. It is also unclassified 16 times as the last column ”Other” indicates. For the 2000/2000 data, the overall error rate is 2.6% for the training and 4.85% for the test set; the unclassified rate is 1.45% for the training and 1.65% for the test set. For the 7188/1991 data, we have error rates 3.7% (training set) and 6.9% (test set) and unclassified rates 2.0% (training set) and 2.5% (test set). In general, these numbers are not impressive. However, there are better ways of using three-way subclassification than the straightforward application of the majority rule. This is to be explored next.

15.3.2 Conditional error analysis and unanimity.

The performance of a classification rule is usually evaluated in terms of two misclassification rates, one for the training data and one for the test data. They are simply the proportion of cases being incorrectly classified in the respective data set. We shall examine these rates more closely.

Our conditional error analysis exploits the largest two values on the score board for each unit $\mathbf{x}$, $M_1(\mathbf{x})$ and $M_2(\mathbf{x})$.

The most interesting condition is when $M_1(\mathbf{x}) = k - 1$ and $M_2(\mathbf{x}) = 0$. Suppose there is an ideal unit $\mathbf{x}$ from say class l, which is very easy to classify. Then for this unit the maximum score $M_1(\mathbf{x})$ is very likely to be $k - 1$ and it is expected to be achieved by class l, $c_l(\mathbf{x}) = M_1(\mathbf{x}) = k - 1$. This is because when class l competes with any other group ($k - 1$ times in total) , the classifier should return l. On other hand, when the competition is between any two classes i, j other than class l, the classifier should return $Others$ because class l is contained in “Others”. Thus $(M_1, M_2) = (k-1, 0)$ represents the situation where an unanimous decision is reached. We anticipate that the final classification to be most accurate

Table 3.1: Misclassification Matrices for Three-way Subclassification Using Majority Rule with the 55-Features-2000/2000 data: (a). Training Set; (b). Test Set.

(a)

$y\backslash\hat{y}$	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*Other*
0	0	0	0	1	0	0	2	0	0	1	6
1	0	0	2	0	0	0	0	0	0	1	6
2	0	1	0	1	1	1	3	0	2	1	2
3	0	0	3	0	0	0	0	0	0	1	4
4	0	1	0	0	0	0	1	0	0	11	0
5	2	0	0	1	0	0	0	0	1	0	0
6	3	1	0	0	0	1	0	0	0	0	2
7	0	0	1	0	0	0	0	0	0	3	0
8	2	0	0	0	0	0	0	0	0	1	4
9	0	0	0	0	0	1	0	1	1	0	5

(b)

$y\backslash\hat{y}$	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*Other*
0	0	0	0	1	0	0	0	0	0	0	5
1	0	0	0	0	0	0	0	0	0	1	7
2	1	0	0	2	1	0	0	2	1	2	2
3	4	0	4	0	0	0	0	0	0	1	4
4	0	1	1	1	0	0	8	1	0	8	0
5	6	0	3	6	0	0	2	0	1	2	3
6	1	0	1	0	5	3	0	0	0	0	2
7	0	0	0	0	0	0	0	0	0	6	1
8	1	0	0	0	1	1	1	0	0	1	8
9	1	2	0	0	1	0	1	11	0	0	1

under this condition.

For the 2000/2000 data, we find that there are 1604 cases in the training set with $(M_1, M_2) = (9, 0)$. From this subset, there are only 4 misclassification cases, representing an error rate of 0.25% which is much smaller than the overall error rate 2.6% given earlier. In the test set, there are 1460 cases with $(M_1, M_2) = (9, 0)$, and 10 out of them are misclassified. This amounts to 0.68% of conditional error, which are again much smaller than the overall error 4.85%. Substantial reduction in conditional error rate also occurs for the 7188/1991 data - 31 out of 5592 (= 0.55%) for the training set and 20 out of 1407 (= 1.4%) for the test set, as compared to the overall rates of 3.7% and 6.9% respectively.

For each fixed value of M_1, we can also anticipate the quality of final classification to go down as M_2 increases because this reflects that the leader faces a stronger challenge from the runner-up and the degree of unanimity goes down. Similarly, for a fixed M_2, the classification quality also degrades as M_1 decreases. Such trends can be found from Tables 3.3(a)-(b)

Table 3.2: Misclassification Matrices for Three-way Subclassification Using Majority Rule with the 55-Features-7188/1991 data: (a). Training Set; (b). Test Set.

(a)

$y\backslash\hat{y}$	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*Other*
0	0	0	0	2	0	1	5	0	0	2	16
1	0	0	3	0	0	0	0	0	0	2	28
2	2	3	0	13	1	3	5	2	6	6	16
3	4	0	5	0	0	6	0	1	2	2	20
4	0	7	2	2	0	0	24	2	3	34	0
5	13	0	4	6	0	0	5	0	2	2	15
6	8	3	0	0	2	7	0	0	1	0	10
7	1	0	2	0	0	0	0	0	1	20	3
8	4	0	0	2	2	6	1	0	0	4	24
9	0	2	0	1	1	1	1	14	1	0	12

(b)

$y\backslash\hat{y}$	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*Other*
0	0	1	0	0	0	0	1	2	1	1	14
1	0	0	3	1	0	0	2	1	2	2	5
2	3	2	0	2	1	4	2	1	2	1	5
3	3	0	2	0	0	7	0	1	2	4	7
4	0	4	3	0	0	0	4	2	1	18	1
5	4	0	0	5	0	0	3	1	0	1	5
6	3	0	1	0	3	2	0	0	0	1	1
7	0	1	0	1	2	1	0	0	2	5	1
8	1	1	3	4	0	1	0	0	0	2	8
9	0	0	0	0	1	0	1	2	0	0	3

and 3.4(a)-(b). For example, in Table 3.3(a) and (b), combining numbers from 4 cells corresponding to $M_1 = 5, 6, 7, 8$, $M_2 = 0$ the error rates are $3/90$ (training) and $8/87$ (test) which are lower than the corresponding error rates for $M_1 = 5, 6, 7, 8$, $M_2 =$ 1- 5/20(training) and 8/ 36(test). For $M_1 = 9$, $M_2 = 1, \cdots, 5$, the error rates are 1/124(training), 5/ 199(test) which are lower than the corresponding error rates for $M_1 = 8$, $M_2 = 1, \cdots, 5$ - 4/9(training) and 6/ 26 (test).

15.4 Further considerations.

In this section, we discuss possible ways of enhancing the three-way subclassification approach.

Table 3.3: Conditional Error Matrices for Three-way Subclassification with the 55-Features-2000/2000 data (Number of Misclassification Cases / Number of Total Cases): (a). Training Set; (b). Test Set.

(a)

$M_1\backslash M_2$	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
0	29/29								
1	8/29	2/2							
2	0/8	4/8	0/1						
3	3/7	0/5	3/4	0/1					
4	2/10	2/3	1/2	0/1	1/1				
5	0/16	0/3	1/3	0/1	2/3	0/1			
6	0/12	0/3	1/3	2/5	0/0	1/1	1/1		
7	0/19	3/10	1/2	0/0	0/1	0/0	2/4	0/1	
8	3/43	2/4	1/2	1/1	0/2	0/0	0/0	0/2	1/1
9	4/1604	0/64	0/20	1/15	0/13	0/12	0/6	0/8	0/3

(b)

$M_1\backslash M_2$	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
0	33/33								
1	14/32	6/7							
2	3/19	4/5	1/1						
3	1/14	3/6	4/6	0/1					
4	1/9	1/3	0/2	1/2	1/2				
5	3/19	1/4	1/3	1/1	2/2	0/0			
6	2/21	1/9	1/2	2/3	0/1	2/3	0/0		
7	1/18	4/11	1/2	0/2	4/6	2/3	1/2	1/2	
8	2/29	2/12	1/6	0/1	3/7	0/3	0/2	1/4	0/2
9	10/1460	2/105	1/44	1/21	1/17	0/12	2/8	0/6	1/5

15.4.1 Conditional error analysis for binary classification.

Unlike three-way subclassification, binary subclassification does not have a straightforward apparatus for unanimity assessment. One possibility is to follow a similar conditional analysis as in the three-way approach. Let $M_1(\mathbf{x})$ and $M_2(\mathbf{x})$ be the highest two scores again , but now obtained from the scoreboard by binary-subclassifications. The ideal condition for most accurate prediction requires $M_1(\mathbf{x})$ to be as large as possible and $M_2(\mathbf{x})$ be as small as possible. The larger the gap between them, the less competitive the runner up is, thus reflecting certain degree of unanimity.

For the digital problem, since there are 10 classes, it is easy to argue that if $M_1 = 9$, then M_2 cannot be smaller than 5. The condition $M_1 = 9, M_2 = 5$ represents the most favorable situation for better classification. We shall anticipate the error rate to increase

Table 3.4: Conditional Error Matrices for Three-way Subclassification with the 55-Features-7188/1991 data (Number of Misclassification Cases / Number of Total Cases): (a). Training Set; (b). Test Set.

(a)

$M_1 \backslash M_2$	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
0	144/144								
1	23/93	9/16							
2	15/72	14/28	2/2						
3	8/52	6/15	3/10	3/9					
4	6/63	5/13	1/3	1/4	1/1				
5	6/53	4/10	4/7	2/3	2/3	1/2			
6	9/63	2/9	2/10	2/3	2/6	3/6	2/3		
7	6/67	7/15	2/7	3/5	7/13	3/3	2/4	4/5	
8	10/114	12/29	2/12	5/12	4/4	2/4	3/8	8/10	3/4
9	31/5592	4/257	1/98	2/51	0/55	2/32	2/35	2/30	2/19

(b)

$M_1 \backslash M_2$	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
0	50/50								
1	14/42	5/10							
2	8/32	4/9	3/3						
3	1/17	0/1	4/6	2/3					
4	4/15	0/6	2/2	4/4	0/0				
5	1/18	2/2	2/2	2/2	1/2	1/1			
6	0/19	4/6	1/3	2/3	2/4	1/2	0/0		
7	2/20	1/4	1/4	0/1	1/2	1/1	0/0	2/2	
8	5/49	1/5	2/3	0/3	3/4	0/0	1/2	5/7	0/3
9	20/1407	5/100	5/33	2/19	1/11	1/19	2/10	3/8	4/10

as M_2 increases. This is indeed what we can find from Tables 4.1(a)-(b) and 4.2(a)-(b). For example, in the test set for 2000/2000 data, when fixing M_1 at 9, the error rates are seen to increase : - 0/3, 0/96, 15/506(=2.96%), 50/1337(=3.7%), respectively for $M_2 = 5, 6, 7, 8$. However, an undesirable pattern is that the most favorable condition $M_2 = 5$ is only satisfied by three cases, while the least favorable condition $M_2 = 8$ has 1337 cases. Thus the conditional error analysis on binary classification does not lead to a useful way of finding a large portion of cases which can be classified with very high precision. The error rate has already reached $15/(3 + 96 + 506) = 2.5\%$ when conditioning on $M_1 = 9, 5 \leq M_2 \leq 7$ as compared to $10/1460 = 0.68\%$ for $M_1 = 9$, $M_2 = 0$ from three-way subclassification reported earlier.

However, a positive note for binary classification is that it can be used to improve the non-(9,0) group from three-way subclassification. The error rate is reduced from $120/540 =$

Table 4.1: Conditional Error Matrices for Binary Classification with the 55-Features-2000/2000 data (Number of Misclassification Cases / Number of Total Cases): (a). Training Set; (b). Test Set.

(a)

$M_1 \backslash M_2$	5	6	7	8
5	0/0			
6	0/0	0/0		
7	0/0	0/0	0/0	
8	0/0	0/0	0/3	1/8
9	0/4	0/113	4/571	15/1301

(b)

$M_1 \backslash M_2$	5	6	7	8
5	0/0			
6	0/0	0/0		
7	0/0	1/1	4/5	
8	0/0	1/1	12/18	21/33
9	0/3	0/96	15/506	50/1337

22.22% (among which $33/540 = 6.11\%$ were unclassified) to $94/540 = 17.41\%$.

15.4.2 Radial partition.

Encouraged by the promising results from three-way analysis, we apply the same strategy to another set of feature variables. This new feature space has 69 variables. Just like the old feature space, they are also constructed using the centre of mass to guide the partition. The difference comes from the geometric configuration of the partitioning lines. We use an 8-region radial partition system.

We begin with the centre of mass of the whole digit. Instead of quadrants, 8 regions are obtained by further partitioning each quadrant diagonally into two equal pieces. The location of the mass center for each of these 8 regions is computed. After that, the whole digit is horizontally divided into two halves - one half is above the x-axis and the other half is below the x-axis. For each half, we then compute the new centre of mass and apply the 8 region radial partition system to get another 8 locations of mass centers. Figure 4.1(a) shows how the same digit 3 is partitioned. A total of 24 mass centers are located in Figure 4.1(b). Our new feature space consists of these 48 locational variables and together with 21 weight variables. Each weight variable represents the total mass from one of the 24 regions obtained before. Note that due to colinearity among each of the three 8-region partitions, we cannot use all 24 weight variables.

Using this new set of feature variables, the error rate from LDA is 5.8 % = 425 / 7291 for

Table 4.2: Conditional Error Matrices for Binary Classification with the 55-Features-7188/1991 data (Number of Misclassification Cases / Number of Total Cases): (a). Training Set; (b). Test Set.

(*a*)

$M_1 \backslash M_2$	*5*	*6*	*7*	*8*
5	0/0			
6	0/0	0/0		
7	0/0	0/0	0/0	
8	0/0	0/0	12/20	41/62
9	0/4	1/117	15/1456	119/5529

(*b*)

$M_1 \backslash M_2$	*5*	*6*	*7*	*8*
5	0/0			
6	0/0	0/0		
7	0/0	1/1	2/2	
8	0/0	0/0	7/11	19/23
9	0/2	1/22	7/409	107/1521

the training set and 9.9%= 199 / 2007 for the test set. They are not much different from the LDA results by 56 features. Can the three-way method help filter out a group of high quality cases?

The result is shown in Table 4.3. Here we find that for the unanimous winners, $M_1 = 9, M_2 = 0$, the error rate in the test set is reduced to below 1% (13 out of 1535 cases).

15.4.3 A combined use of different feature spaces.

As pointed out before, three-way subclassification is especially effective in isolating high quality cases- cases that are easier to predict their membership. This is achieved by a conditional error analysis which exploits the degree of unanimity among different classification decisions from subclassification. After locating the very high quality cases, we can then focus on the rest of cases and try to find better classification, perhaps even with drastically different classification methods. For example, consider the nearest neighbor classifier used by LeCun et al.. As mentioned before, it has an error rate of 2.5% for the 2007 test cases, but is computationally very demanding. If we can use it only for the non-($M_1 = 9, M_2 = 0$) cases, then the overall error rate would be still be at most around 3%. But in this way, we have allocated the total computation time more effectively without sacrificing much of the overall classification quality.

Perhaps an easier way of improvement is to combine the results from different feature spaces. We try the following path.

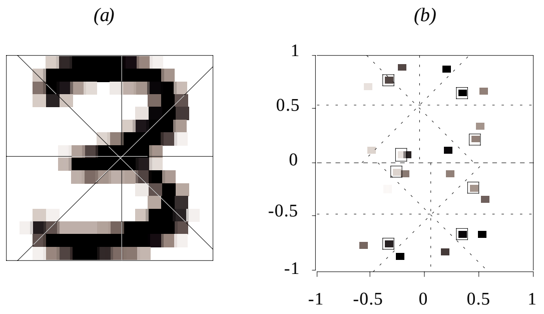


Figure 4.1: Radial Partition for Digit 3: (a). First Level Radial Partition; (b). Locations of Mass Centers with Weights for three radial partitions.

Table 4.3: Misclassification Matrix for Three-way Subclassification Using Majority Rule with the Feature69-7291/2007 data.

$M_1 \backslash M_2$	0	1	2	3	4	5	6	7	8
0	35/35								
1	9/26	7/10							
2	3/13	6/10	2/3						
3	7/17	4/6	6/9	4/4					
4	2/14	1/2	0/2	0/3	1/1				
5	4/12	2/3	3/4	0/1	0/1	1/1			
6	1/11	1/4	2/3	2/2	2/2	1/2	0/0		
7	3/18	3/5	2/4	1/2	2/4	2/4	0/1	1/1	
8	3/28	2/11	4/7	0/3	1/2	3/5	2/2	3/6	1/1
9	13/1535	5/69	1/22	1/15	1/15	0/12	1/10	3/12	4/12

(1). Apply three-way subclassification trained by 69-features to all test cases and locate the $M_1 = 9$, $M_2 = 0$ group.

(2) Apply three-way subclassification trained by 56-features to the non-$(9, 0)$ cases and locate the $M_1 = 9$, $M_2 = 0$ cases.

(3) Apply binary classification trained by 69-features to all other left-over cases.

A breakdown of the error rate is given in Table 4.4. The overall error rate is now about 5.7%, which is compatible with the result (between 5% and 6%) by the neural network approach. It is certainly a great improvement over the original LDA result which is about 10% for either feature. It is also significantly better than the 8.2% error rate obtained by Hastie, Buja, and Tibshirani (1995).

It is interesting to observe that there are a good number of tied scores in binary subclassification. In order to keep the procedure simple, we resolve these ties essentially by a random choice. Further investigation on how to handle these cases seems worthwhile.

Table 4.4: Breakdown of Error Rate for Combining Use of Different Feature Spaces.

Group	Cases	Misclassified Cases	Error Rate
(1). $M_1 = 9$, $M_2 = 0$/3-way/69-features	1535	13	0.85%
(2). $M_1 = 9$, $M_2 = 0$/3-way/55-features	160	15	9.38%
(3). Left-over/binary/69-features	312	87	27.88%
Total	2007	115	5.73%

15.5 Conclusion.

Discriminant analysis is relatively easier when the number of classes is small. In view of this tendency, subclassification appears to be a promising strategy for alleviating the complexity of many classes. In this article, we propose a three-way subclassification method and show that it can be fruitfully applied to complement binary subclassification.

Three-way subclassification is designed to exploit the degree of unanimity among various decisions during subclassification. Thus, unlike the binary situation, the majority rule by itself is not appropriate for the three-way subclassification. Instead, the information gathered from the conditional error rate table in the training set is used for grading the discriminant quality of an incoming unit **x** to be classified. If it falls into the highest grade group $(M_1(\mathbf{x}), M_2(\mathbf{x})) = (k-1, 0)$, then we have the best confidence about the classification accuracy. On the contrary, if it falls into very low grade group with small $M_1(\mathbf{x})$ or large $M_2(x)$, then we had better send it to other more powerful classifiers and hope for a better result.

In many industrial applications, tolerable error rates are usually set up by practical concerns from the economic aspects. Thus it is important to identify sub-populations that are easier to classify than others. Our method has the appeal of being able to identify such higher-grade subpopulations. For the remaining lower-grade subpopulation, we can then rely on more complicated methods to carry out the task. The ability to filter out cases that are harder to classify is a very important consideration in quality management because this helps engineers identify where the quality improvement should be focused on. Further investigation along this line of thoughts is worth pursuing.