2006

Two-dimensional penalized signal regression for hand written digit recognition

Qing Tang
Louisiana State University and Agricultural and Mechanical College, qtang1@lsu.edu

TWO-DIMENSIONAL PENALIZED SIGNAL REGRESSION FOR HAND WRITTEN DIGIT RECOGNITION

A Thesis
Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Applied Statistics

in

The Department of Experimental Statistics

by
Qing Tang
B.S., Wuhan University of Science & Technology, 1992
August 2006

ACKNOWLEDGEMENTS

I express my sincere thanks to my advisor, Dr. Brian Marx for his guidance, advice, patience and encouragement. His insights and suggestions have always been valuable and beneficial during the course of this research. During the two years, his kind offer of research assistantship has been so supportive and helpful.

I give special thanks to Dr. John Pine, the Chair of the Department of Geography and Anthropology. It is from the cooperation of Dr. Marx and him that the assistantship was originated.

I would like to thank other committee members, Dr. James Geaghan and Dr. Kevin McCarter who were enthusiastic to serve on the examining committee and gave valuable suggestions.

Furthermore, my thanks are extended to all other faculty and staff of the Department of Experimental Statistics who have offered help and been kind to me and made the past several years enjoyable and memorable.

Finally, I express gratitude to my family members for their love and encouragement. Without their support, this work would not have been completed. I would like to share my happiness with all my family members.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Many attempts have been made to achieve successful recognition of handwritten digits. We report our results of using a statistical method on handwritten digit recognition. A digitized handwritten numeral can be represented by an image with grayscales. The image includes features that are mapped into two-dimensional space with row and column coordinates. Based on this structure, two-dimensional penalized signal logistic regression (PSR) is applied to the recognition of handwritten digits.

The data set is taken from the USPS zip code database that contains 7219 training images and 2007 test images. All the images have been deslanted and normalized into 16 x 16 pixels with various grayscales. The PSR method constructs a coefficient surface using a rich two-dimensional tensor product *B*-splines basis, so that the surface is more flexible than needed. We then penalize roughness of the coefficient surface with difference penalties on each coefficient associate with the rows and columns of the tensor product *B*-splines. The optimal penalty weights are found in several minutes of iterative operations. A competitive overall recognition error rate of 8.97% on the test data set was achieved.

We will also review an artificial neural network approach for comparison. By using PSR, it requires neither long learning time nor large memory resources. Another advantage of the PSR method is that our results are obtained on the original USPS data set without any further image preprocessing. We also found that PSR algorithm was very capable to cope with high diversity and variation that were two major features of handwritten digits.

CHAPTER 1. INTRODUCTION

Computer technology has dramatically increased the efficiency of office and business, and we rely on electronics more than ever before. Overwhelming amounts of handwritten documents are produced, and how to process this information efficiently has become a bottleneck of office automation and electronic business. Even though handwritings by different people vary, computers are more powerful than ever before, and machine recognition of handwriting has become possible.

Because of lack of handwriting data samples, reliable recognition algorithm, and the need for powerful computers, there has been very limited progress made in handwriting recognition. Bell Lab group has been the pioneer in this area for years. According to their published reports, the core of their technology is the so-called classifier algorithm, and great achievements have been attained. They have evaluated various kinds of classifier algorithms from memory-based classifiers to networked neural architectures. It was found that the recognition accuracy is dependent on how "good" and how large the databases are and which classifiers they used. This indicates the inadequateness of the classifier algorithm technology. Even though the accuracy of recognition can be improved by training the algorithm with a given sample, the recognizer will be still short of memory when we use it for recognizing different samples obtained from different groups of people. Also, the efficiency of the algorithm is influenced by the training time and run time. It is therefore difficult to say this so-called classifier algorithm is a kind of universal handwriting recognition technology.

Behaviors of various digit samples obey the statistical principles. Therefore, it is possible to develop a model based on statistical methods to extract the individual patterns from a group of handwriting samples and recognize them. In this thesis, we will introduce our method and demonstrate its advantages as an excellent candidate of practical handwriting digit recognition algorithm. The algorithm we have developed is based on the successful logistic PSR algorithm. This algorithm cost surprisingly much less time on the overall recognition process even on commonly used PC platforms. Most other algorithms require working environments of at least workstations of large scale computing system. All of our tests were accomplished on PC platforms, which indicates the wider and more practical application potentials of our method. Along with the merits of timesaving and low-resource-demanding, our method also achieved high recognition rates among the top ranks of all algorithms on the USPS zip code data set. Noticeably, we used the original images from the USPS data set without any preprocessing including further thinning, binarization or segmentations

CHAPTER 2. LITERATURE REVIEW

2.1 A REVIEW OF RECOGNITION ALGORITHMS

Pattern recognition is one of the most important abilities of human beings. Relying on this ability, human beings extract useful information about their surroundings. Today, as the digital computer technology has been largely used and developed, to simulate this unique human ability with automated machines becomes more and more realistic. Much significant work has been done in the area of machine pattern recognition. Simply, pattern recognition includes the following three procedures:

1. Pattern acquisition with transducers or receptors.
2. Feature extraction that gets the most useful information from the pool of input data and get rid of the irrelevant information.
3. Classification of the extracted features with specially designed algorithms that are usually task-oriented or general-purpose-based in some cases.

Automatic recognition of printed and written text was one of the first goals of early research in pattern recognition. Optical character reader (OCR) was a very successful technique that based on statistical pattern classification methods. OCR technique has been widely used for printed text recognition and achieved recognition rates higher than 90%. Because of the high degree of variation in shape and size of hand written characters especially hand written digits, OCR is completely not suitable for hand writing recognition. Later, Artificial Neural Network was proposed to solve this tough problem.

Methods of character recognition may be divided into two approaches: pattern matching and structural analysis (Dimauro, Impedovo and Pirlo, 1992). Pattern matching can be performed both directly on the bitmap of the images and also in the feature space. Common feature types for pattern matching are the number of writing strokes, the orientation of strokes contained in the input image and the relation between strokes. Structural analysis approaches are recently considered more useful for handwritten character recognition where strong shape distortions are generally present. Nevertheless, the adaptive nature of the neural networks has been stressed by many authors who emphasized the possibility to use with success neural networks for pattern matching in handwriting recognition.

Handwriting recognition techniques can be classified into two major groups, the online and offline recognition. Online recognition processes the data input such as handwritten characters, digits and signatures with special devices or pen that directly attached to the system. Offline recognition systems process the input data that have been attained with optical scanner from paper documents. Compared to online recognition, offline recognition could be more difficult in that much useful information can be lost such as the handwriting duration, writing sequence, number of strokes and writing direction. For this reason, successful offline recognition has significant meaning to the people in the area to deal with static and existing handwritten matter. Several techniques have been used for offline recognition; such as statistical, structural and neural network approaches. It has been reported that neural networks have achieved good results in handwritten character recognition due to their ability to learn and generalize. Neural network approaches have been applied for online and offline segmented character and numeral recognition, and online and offline cursive words recognition.

In a brief review of handwriting recognition, Senior (1992) summarized the work done for an online system by Morasso et al. who describes a Kohone self-organized network used to classify strokes into similar forms and build what the suthors refer to as a “graphotopic map”. Stroke information from pairs of letters is fed into a backpropagation network that is trained to identify the digraph. A 97% accuracy rate was achieved. A second set of Kohonen networks was created, grouping consecutive strokes into characters and creating an allograph lexicon of possible stroke sequences which are labeled with their character names. Work done on segmented character recognition has been also summarized. Elliman et al. use features, such as end-point, junction, curve and loop, each of which is associated with a numerical quantity, such as curvature or length, before being decoded in a neural network either backpropagation or adaptive feedback classifier. Senior achieved 85.5% recognition rate by applying the method of recurrent error propagation networks to the task of offline cursive script recognition. Proposed enhancements in preprocessing and durational modeling are claimed to have significantly improved the system performance.

Nellis et al. (1991) used global features created by separately examining the left, right, top and bottom edges of each character (Nellis and Stonham, 1991). This information is then fed into Aleksander's discriminator neural network. Hepp uses a similar set of morphological features that are fed into a backpropagation neural network (Hepp, 1991). Fukushima used the neo-cognition to recognize hand written characters (Fukushima, 1992). After learning, the neo-cognition recognizes input patters with little effect from deformation, changes in size or shifts in position. A new learning algorithm " selective attention model" has the ability to segment patterns as well as recognizing them. Dimauro et al. review the approach proposed by Wang et al., which is based on a specific feature vector called "histogram" and a neural network for the classification stage (Dimauro, Impedovo and Pirlo, 1992). A "histogram" is a feature vector extracted directly from the bitmap of image. The feature vector is then used as then input for a multilayer neural network classifier.

Analysis of feedforward neural networks for handwritten character recognition was performed by Starzyk and Ansari (1992). Two methods for fast training based on the Parzen window estimates of defining the vector space for different classes were used. Accuracy similar to the accuracy of backpropagation is obtained with less training time and simpler network architecture.

Some methods based on neural networks require long time to learn. Once the learning procedures are accomplished, their response to the feeding is extremely fast and reliable. Even though, fast response and high reliability are important considerations for automated recognition systems, long learning or training procedure and requirements of large training samples are critical for building practical recognition systems.

The feeding data to the recognition systems are usually a pool of irrelevant information containing many features. Features can be symbolic or numerical matter that bears useful or irrelevant information about the recognition tasks. The most important thing to do for the recognition systems is to be able to extract the relevant features from the input data. It is easy to understand that it will be impossible to fulfill the recognition task if too many features are presented if they are not all relevant. So, how to effectively and accurately extract relevant features form the feeding is the first step to do for handwriting recognition to be practical. To be relevant for the extracted features is not enough to accomplish the tasks though. Feature extraction is a procedure similar to the judgment flow by human beings. The practical judgment criteria can be variant depends on the task itself. But, in general, the features should be relevant that they can provide continuous information flow for the following recognition procedures. Also, they should be feasible for computation. They should be good enough to be classified to produce as few as possible mis-classification errors. The feature judgment should be able to retain as much as possible the valuable and crucial information to help solve the recognition problem.

Following the feature extraction, it is the feature classification process. Usually, the pattern recognition problem is considered as a classification problem. So that it is simple to benchmark different recognition methods by testing their respective classifiers or algorithms.

As for the neural network methods, many kinds of classifiers have been proposed and tested with various data representations, such as Baseline linear classifier (Duda and Hart, 1973), Baseline nearest neighbor classifier, LeNet 1 (Le Cun, et al., 1990), LeNet4 (Le Cun, et al. 1991) Boosted LeNet 4 (Schapire, 1990, Drucker, Schapire and Simard, 1993), Tangent Distance Classifier (TDC) (Simard, Le Cun and Denker, 1993) and Tangent Vector Combining Local representations (Keysers, Paredes, Ney and Vidal, 2002, Keysers et al., 2000, Dahmen et al., 2001). Many neural network based methods have been tested with the USPS zip code data set and achieved remarkable results. Among these methods, the ones that have gained the most attention are: backpropagation algorithm by Le Cun et al. (1989 & 1990), deformable prototype matching algorithm by Hastie, et al. (1994, 1997) and simplified genetic algorithm by Parkins et al. (2002 & 2004). The ultimate goal for the USPS zip code recognition should be close to that by human, which is about 2.5% of error rate.

2.2 BACKPROPAGATION NETWORK

Le Cun et al reported their results on handwritten zip code recognition by using backpropagation network in 1989. The backpropagation network is multilayered (input layer, hidden layer and output layer) and arranged in a feedforward architecture with the ability to send back the error between the actual output and the target output values. Each layers contains interconnected elements that behave like soft linear classifiers. Each element computes a weighted sum of the inputs and transforms the sum through a nonlinear squashing function. The learning is fulfilled by iterations of modifying the weights on each connection so as to minimize an objective function that was popularly the mean square error between the actual output and the desired output. Backpropagation was used to calculate the gradient of the objective function.

The network had four hidden layers named as H1, H2, H3, and H4. H1 and H3 are shared-weight feature extractors and H2 and H4 are averaging /subsampling layers. The 16x16 inputs were enlarged to 28x28 to avoid boundary problems. Constraints were added to the network architecture so that the system can be fed with images rather than feature vectors and the learning time can be relatively short due to the imposed a priori knowledge on the system. The data set they used was the well-known USPS zip codes that contain totally 9298 digits digitized and preprocessed from handwritten zip codes appeared on mails at the Buffalo NY post office. Their network was trained for 23 passes through the training set. The performance of their backpropagation network was measured both on the training and test set. Noticeably, in order to achieve 1% error rate on the non-rejected data for the test set, a rejection rate up to 12.1% was required. If the reject rate was counted to the overall error rate, the performance of their algorithm would be far from the amazing 1% error rate. Most misclassifications were thought to be due to erroneous segmentation of the images.

The authors believed other misclassifications were caused by ambiguous patterns, low-resolution effects or writing styles not present in the training set. Actually, all of the reasons mentioned above are factors beyond the control of any recognition algorithms. They are predetermined by the preprocessing of the raw data. For this reason, such kinds of images that cause the misclassifications should be rejected from the testing set and the performance of recognition algorithms is measured by recognizing the cleaned data set. But, on the other side, the images in the cleaned data set could be very "easy" to recognize for most of the algorithms developed lately. Thus the performance differences among those algorithms on the cleaned data set would be very small. If that is true, it will be very insignificant to compare the performance of various recognition methods on "cleaned" data set. So, we believe that an good algorithm should recognize more digits from those data rejected as erroneous samples. According to this idea, we count the error rate for the same data set that is not cleaned at all for each performance measurement. So, the over all error rate of Le Cun's test in 1989 should be around 13.1% on the whole data set without any rejection.

Le Cun et al. also reported another result by the same algorithm on the same data set that was supplemented with 3349 printed digits in 1990. The added printed digits were generated by a stochastic model from 35 fonts. The error rate was 1% for 9% reject rate. So the error rate for the whole data set should be around 10%, which was improved by about 3% compared to the previous result. Considering the fact that 3349 printed digits had been added to the same sample as the one used previously, the improvement is barely contributed by the printed digits since the author found no error were made on the printed characters. This time, the learning was finished after 20 passes through the training set which contained 2549 printed digits. Even though the authors thought that the learning time for their network was short due to the redundant nature and constraints imposed on the network, the training procedure was necessary.

2.3 DEFORMABLE PROTOTYPES AND MATCHING STATISTICS

Hastie et al. have developed methods for handwritten USPS zip code digits recognition. Their main idea reflected the application of statistics in hand written digit recognition. They made one or multiple prototypes for each digit, which were represented by piecewise-linear curves. The input digit images were preprocessed and converted into point-set representations of the images. The classification step involves an iterative fitting by least squares, allowing a free affined transformation in R^2. The classification step is flexible and uses all the statistics for matching as the inputs of the algorithm. They had totally 13 prototypes used in this experiments to represent the ten numerals. Some digits have more than one prototypes because they may appear in more than one valid writing forms as that for digit "2" that has two commonly used written forms. The authors believed that if they used more prototypes, they could be able to improve the misclassification rate by nearly 2% on the USPS test data.

The preprocessing is to further process the images obtained from the normalized data set by AT&T group. This time, the images are converted to binary point sets. They had developed different strategies to get point-set binaries for images already in binary and those with grayscales. The prototypes are made deformable so that they can fit the image point-set as closely as possible. Their matching statistics included the measures of minimum distance, coverage, aspect ratio, rotation and shear. Some mismatched situations in which the sum-of-square and coverage measures are in sufficient have also been considered and zigzag and prototype walk strategies are used to avoid paradoxical results.

Their results on recognizing the USPS zip codes reported in 1994 were recorded as 91% recognition rate with 2% error rate at a 7% reject rate and 92% recognition rate with 3% error rate at 5% reject.

2.4 GENETIC ALGORITHM

Parkins and Nandi (2004) reported their results on the USPS zip code data set in 2002 and 2004 by using a Genetic Algorithm with reduced complexity of classifier and feature set. Initially, Genetic Algorithm was developed as a machine-learning model that derives its behavior similar to the evolution mechanisms happen in nature. The learning machine resembles the collection of a population of DNA molecules that carry information regarding to evolution. Numerous combinations of the four basic building blocks of DNA exist. Evolution occurs when one of the new valid optimal combinations of the four nucleotides is resulted. Genetic algorithm is a method that uses similar mechanism to search the optimal solution in the candidate solution subsets to the classification problems.

The genetic algorithm used by the authors was from the GAlib C++ library. The genetic algorithm evaluates a population of test genomes and selects some of them to step to the next generation. A genome encoding scheme were devised to map all the solutions on to a genome. In the experiment reported, a fixed subset of features was selected from a global pool. Each available feature is assigned a unique number and the genome represents the list of the feature numbers contained in the subset being evaluated. In order to compare the results of the performance of a neural network and that of the feature selected set. The result for the former is about 88% and 90% for feature selected sets is saturated at 90%. From the results listed above, we found the best performance of the recognition methods discussed here is about 90%.

Comparing Hastie's results with those of Le Cun's and Parkins', it is very interesting to find that Hastie's results slightly outperform that of Parkins' by 1%-2% (feature selected sets). But, Le Cun's results are much inferior to both by 3-4% in error rate. Comparing the work done by Parkins and Hastie, we still find that Hastie's method involved a preprocessing step to convert the raw images into point-set binaries. Obviously, this is a very effective way to remove some erroneous ink, confusing grayscale pixels and continuous strokes like OCR method prefers binary images than grayscale or colorful images so that it could help improve performance of their method. By reviewing the previous work in handwritten zip code recognition, we believe that 90% classification rate is good enough as the benchmark point for different algorithms.

CHAPTER 3. DATA DESCRIPTION

The data base used for hand written recognition consist of 9298 segmented digits that were scanned and digitized from hand written zip codes that appeared on U.S. mail passing through the Buffalo, NY post office. All the zip codes were written by many different people in different places. The writers wrote the zip codes in a great variety of size, with different writing styles, and instruments, with widely varying amounts of care. The data set includes 7291 segmented digits that are used for training samples and 2007 are used for testing the generalization performance of recognition algorithms.

3.1 FEATURE OF THE DATA

As we know, the USPS zip codes have 5 digits whose combinations indicate the localizations of US postal services. Figure 3.1 shows images that are obtained by scanning the hand written zip codes on envelopes and stored in the form of standard images format with gray scales. It is known that handwritings from different writers are quite different so that they are widely used as identity authorization for bank check, personal document signature, etc. It is also very interesting that different samples from the same writer can be very different due to varied locations, duration and preference of writing.

The digits were machine segmented from zip code strings by an automatic algorithm. Figure 3.2 shows the segmented zip codes are grouped by digits. In this example, some digits still contain some unnecessary segments. Some hand written digits largely deflect from the standard writing rule. Some digits severely tilted to right hand.

Hand-written characters are more difficult to interpret than typewritten characters because of the great variety not only between different writers but also between different samples from the same writer. The matter can be further complicated because different letters can have very similar shapes the segmented characters sometime included extraneous ink and sometimes omitted critical fragments. These segmentation errors often resulted in characters that were unrecognizable or appeared mislabeled. For example, the top horizontal dash of digit 5 could make it appears to be 3 in handwriting, and thus it could be mislabeled. Such kinds of digits comprise approximately 2% of the test data set. These characters limited the attainable accuracy. On the other hand, these kinds of digits present great challenge to current recognition algorithms and demand the necessity of more intelligent recognition algorithms. In order to keep the objectivity, we have kept such digits in our data set. Therefore, it contributed the largest portion of the resulting error rate or misclassification rate.

Another important feature of this data set is that it contains numerous examples that are ambiguous, unclassifiable, or even misclassified (LeCun et al., 1990).

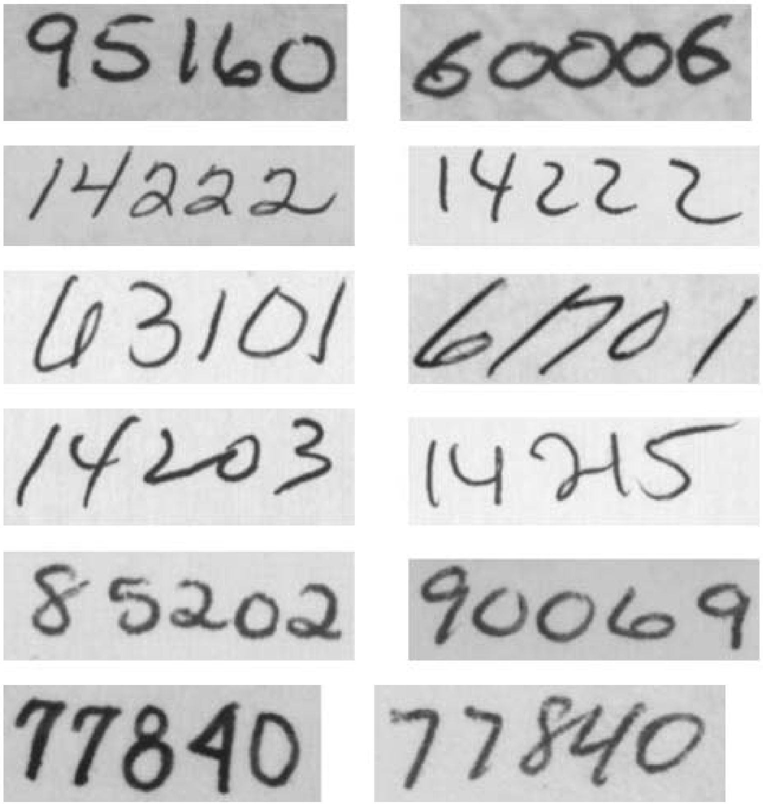

Figure 3.1 Samples of original zip codes

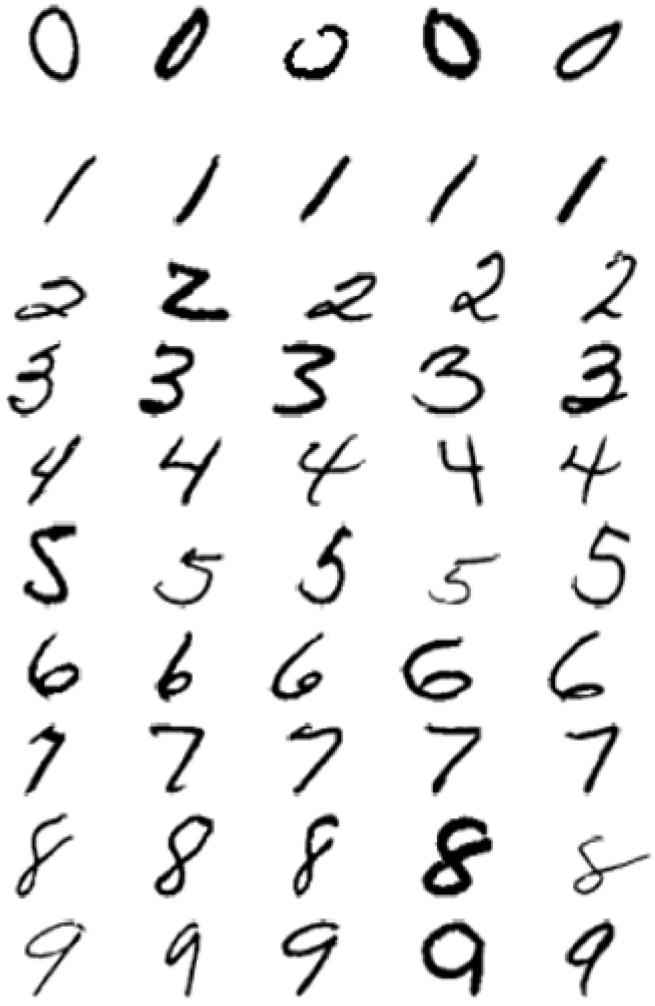

Figure 3.2 Samples of segmented zip code digits

3.2 PREPROCESSING THE DATA

The original scanned digits are binary and of different sizes and orientations; the images here have been deslanted and size normalized, resulting in 16 x 16 grayscale digit images (Le Cun et al., 1990). The normalization of the handwritten digits automatically scanned from envelopes was done by the U.S. Postal Service. Figure 2.3 shows the samples of normalized images from the test data set.

As each normalized image has 16×16 pixels, an image can be read as a 16×16 matrix. We unfolded each image by row and change it to a 1×256 matrix. Then we added the label that identifies the corresponding image before the first column and made it a 1×257 matrix. In this way, we converted the entire training dataset to a 7291×257 matrix and the validation dataset to a 2007×257 matrix. For these two matrices, the first column (y-predictor) are labels for digits from 0 to 9; column 2 to 257(x-regressors) are pixel grayscales. Each pixel has its grayscale represented by a decimal between –1 to 1, for which –1 stands for completely white and +1 completely dark.

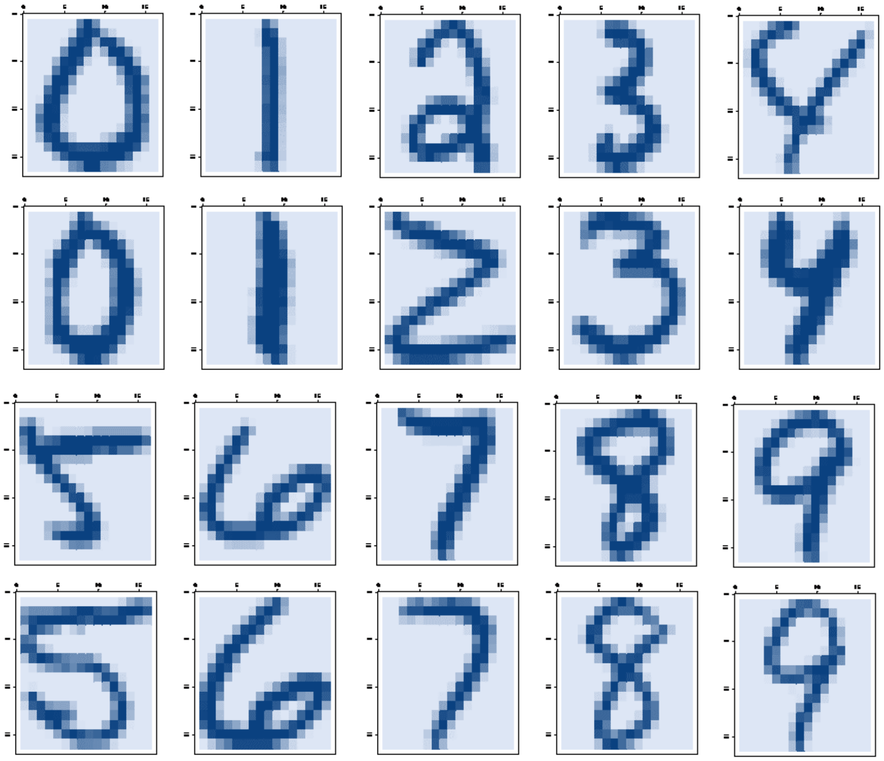

Figure 3.3 Sample of normalized images from test data set

3.3 DISTRIBUTION OF THE DATA

The USPS zip code dataset contains 7219 training images and 2007 test images. Table 3.1 shows the distribution of the data set. In both training and test dataset, all digits do not appear with equal frequency.

Table 3.1: Data distribution in training and test dataset

	0	1	2	3	4	5	6	7	8	9	Total
Train	1194	1005	731	658	652	556	664	645	542	644	7291
Test	359	264	198	166	200	160	170	147	166	177	2007

CHAPTER 4. TWO-DIMENTIONAL (2D) PENALIZED LOGISTIC SIGNAL REGRESSION (PSR)

An image fitted into a 16×16 pixels can be read as a two-dimensional 16×16 $(p \times \breve{p})$ matrix. We unfold this matrix by row and obtain a vector of 1×256. Since each pixel can be read as a regressor, we get a total of 256 regressors. The label of an image digit d=0, 1, 2,…,9 will be the scalar response *(y*).* When we focus on a certain digit *(d)*, the response variable y can be regarded as binary indicators. When $y^*=d$, we record y=1, when $y^*\neq d$, we record y=0. So the main idea of our method is to apply logistic regression approach to the dataset ten times.

Further more, in order to make the coefficient surface to be smooth, a two-dimensional penalized logistic regression model is applied to the data. We calculate the probability of an image to be each digit *"d"* by the 2D logistic PSR. Totally, ten probabilities are obtained for each image. The highest possibility will be chosen as the classification of an image.

4.1 LOGISTIC REGRESSION

Logistic regression is a very important and popular statistical method for binary response variable *(y)* and quantitative explanatory variable *(X)* to predict the probability of a certain event occurring. In our handwritten digit recognition problem, when we work on a certain digit *(d)*, multiple response variable *(y*)* can be classified into two categories, "true" or "false", that is when $y^*=d$, say "true"; or when $y^*\neq d$, say "false". For each digit *(d)*, we construct a logistic regression model from the training data set with 256 coefficients (β_1 to β_{256}). The model has a multiple linear regression form, we can express the predictor as:

$$\log\left(\frac{\pi_d(x)}{1-\pi_d(x)}\right) = X\beta_d = \eta_d ,$$

where $\pi_d(x)$ is the "true" probability when X takes value x and β_d is the vector coefficients (β_1 to

β_{256}). The logit function $\log\left(\frac{\pi_d(x)}{1-\pi_d(x)}\right)$ is the log-odds of a “true” as opposed to a “false” and $X\beta_d = \eta_d$. We are actually modeling the log-odds as a function of the regressors. The probability $\pi_d(x)$ can be solved by the inverse function as

$$\pi_d(x) = \frac{\exp(\eta_d)}{1+\exp(\eta_d)},$$

where the logit or log-odds will range from negative infinity to positive infinity and the probability $\pi_d(x)$ will range from 0 to 1.

When a new image *(i)* in test data set comes, its pixels will be weighted by the coefficient β_d of digit *(d)*. The sum of this weighted pixels is the scalar that is used for the logistic equation for digit *(d)* to calculate the probability $\hat{\pi}_d(x_i)$ of this image *(i)* to be digit *“d”*

$$\hat{\pi}_d(x_i) = \frac{\exp(\hat{\eta}_d)}{1+\exp(\hat{\eta}_d)}.$$

Based on this idea, for each digit *(d)*, we construct a logistic regression model from training data and totally 10 logistic models are obtained while every model has different coefficients. We then apply these models to each image in test data and predict its possibility to be a digit 0, 1, 2,…,9. Among these 10 possibilities, the highest one will be chosen as an image’s classification.

4.2 2D LOGISTIC PENALIZED SIGNAL REGRESSION

For each digit *(d)*, its logistic regression model has 256 coefficients. These coefficients construct a 16×16 surface which reflects the images’ 16×16 pixel locations. Based on the logistic regression method we discussed, we further assume the coefficient surface to be smooth and take the image’s spatial structure into consideration for reliable digit prediction. A two-dimensional logistic penalized signal regression model is proposed to the zip code data.

Marx and Eilers (2005) provided a practical solution for the 2D signal PSR regressors and forcing their estimated coefficients to be smooth (using P-splines). There are two steps towards such smoothness: (a) we purposely overfit the coefficient surface (not the signal) using two-dimensional tensor product *B*-spline, making the surface more flexible than needed. (b) We penalize estimation of the surface using difference penalties on each of the rows and columns of the tensor product *B*-spline coefficients.

4.2.1 Tensor Product Mean Model

Since each image is located into a two- dimensional 16×16 pixels, we index axes as v and $\breve{v}$. We can locate these pixels (regressors) as:

$$(v_1, \breve{v}_1), (v_1, \breve{v}_2), \dots, (v_{16}, \breve{v}_{16}) \ ,$$

when regressor image digitized on plane $(v, \breve{v})$, we have

$$\eta = \sum_{j=1}^{16} \sum_{k=1}^{16} x_{jk} \underbrace{\beta(v_j, \breve{v}_k)} = \beta_{11}x_{11} + \beta_{12}x_{12} + \dots + \beta_{1616}x_{1616} \, .$$

The tensor products *B*-spline exits in the $v \times \breve{v}$ plane. We choose to divide the domain v_1 to v_{16} ($\breve{v}_1$ to $\breve{v}_{16}$) into 10 equal intervals using 11 interior knots. Taking each boundary into consideration and choosing the first order penalty, we have a complete basis with total 13 *B*-splines on the axis. The full tensor basis has dimension of 13×13 $(n \times \breve{n})$. Figure 4.1 displays nine tensor product *B*-spline, which is a portion of a full basis.

These equally spaced indexing knots are placed on $v(\breve{v})$ to yield a regularly-spaced grid, carving out the plane into subsquare. The *r*th-*s*th single tensor product is presented as $B_r(v)\breve{B}_s(\breve{v})$. $r = 1, 2, \dots, 13$, $s = 1, 2, \dots, 13$. The coefficient surface β can be reexpressed through a multiple regression using tensor products,

$$\beta(v_j,\breve{v}_k)=\sum_{r=1}^{13}\sum_{s=1}^{13}B_r(v_j)\breve{B}_s(\breve{v}_k)\gamma_{rs},$$

where γ_{rs} is the unknown amplitude of the tensor products. Weighing each tensor product by its γ and summing produces a smooth coefficient surface. The coefficients drive the surface, changing the γs changes the surface. The tensor products also provide a reduction in coefficients estimation through smoothness.

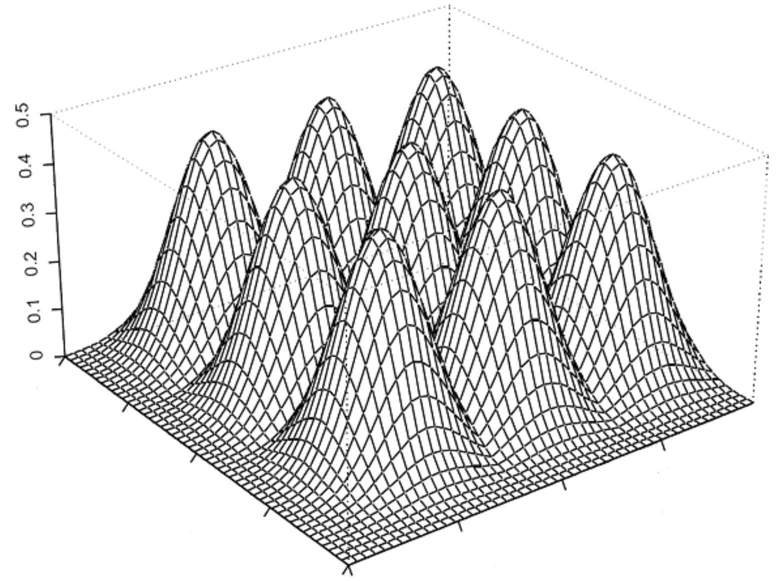

Figure 4.1 Nine cubic *B*-spline tensor products, a portion of a full basis

4.2.2 2D Penalized Logistic Signal Regression Model

As our main idea is apply a logistic regression model to the zip code data, we can obtain the model from the 7291 training images as

$$\log\left(\frac{\pi_d(x)}{1-\pi_d(x)}\right)=X\beta_d=\eta_d \ ,$$

where η_d is a 7291×1 matrix, X is a 7291×256 matrix of the image grayscale, β_d is the 256×1 coefficient vector for digit *"d"*. This equation can be re-expressed by using tensor product *B*-spline as

$$\log\left(\frac{\pi_d(x)}{1-\pi_d(x)}\right)=\eta_d=\mathrm{XT}^*\gamma_d==\mathrm{M}\gamma_d\,,$$

where $\mathrm{T}^*\gamma_d$ becomes the 256 by 1 coefficient vector. γ_d is the height of the tensor products for digit "*d*" and $\mathrm{M}=\mathrm{XT}^*$.Solving for $\pi_d(x)$, we have

$$\pi_d(x)=\frac{\exp(\mathrm{M}\gamma_d)}{1+\exp(\mathrm{M}\gamma_d)}.$$

The estimation γ_d is to maximize the log-likelihood function $L(\gamma_d)$ since when we focus on a certain digit *"d"*, the response variable *(y)* is changed to binomial responses. Each image response can be regarded an independent Bernoulli observations, it should be either a digit *"d"* or not a digit "*d*".

$$Y_i \sim BIN(N=1,\pi_{di})$$

the likelihood function for the training data will be

$$\begin{aligned}L(\gamma_d)&=\pi_{d1}{}^{Y_1}(1-\pi_{d1})^{1-Y_1}\pi_{d2}{}^{Y_2}(1-\pi_{d2})^{1-Y_2}\ldots\pi_{d7291}{}^{Y_{7291}}(1-\pi_{d7291})^{1-Y_{7291}}\\&=\prod_{i=1}^{7291}\pi_{di}{}^{Y_i}(1-\pi_{di})^{1-Y_i}\end{aligned}$$

the log-likelihood

$$\begin{aligned}l(\gamma_d)&=\log[L(x_d)]=\log\prod_{i=1}^{7291}\pi_{di}{}^{Y_i}(1-\pi_{di})^{1-Y_i}\\&=\sum_{i=1}^{7291}[Y_i\log(\pi_{di})+(1-Y_i)\log(1-\pi_{di})\end{aligned}$$

then plug in

$$\hat{\pi}_{di}(x)=\frac{\exp(\mathrm{M}\hat{\gamma}_d)}{1+\exp(\mathrm{M}\hat{\gamma}_d)}.$$

We see that $l(\gamma_d)$ is non-linear in γ, the method of solving is used to maximizing $l(\gamma_d)$.

4.2.3 Penalty On Tensor Product Coefficients

In the *P*-spline regression model, we impose discrete roughness or differences penalties on γ. A separate difference penalty is assigned to each of its rows and its columns. The penalties have structure to effectively break the linkage in the penalty from row to row or from column to column. The objective function is to maximize the penalized binomial log-likelihood function

$$l^*(\gamma) = l(\gamma) \text{ - Row Penalty - Column Penalty}$$

$$l^*(\gamma) = l(\gamma) - \lambda_1 \gamma' P'_1 P_1 \gamma - \lambda_2 \gamma' P'_2 P_2 \gamma \ .$$

The penalty has two parts, which includes difference penalties on rows and difference penalties on columns. The weighs λ_1 and λ_2 regularize the penalties, one associated with rows and one associated with column of Γ, where Γ donates for the unknown coefficient matrix of γ_{rs}. The *P*-spline solution for the coefficients γ is

$$\hat{\gamma}_{t+1} = (\mathrm{M}' \hat{W}_t \mathrm{M} + \lambda_1 P'_1 P_1 + \lambda_2 P'_2 P_2)^{-1} \mathrm{M}' \hat{W}_t \hat{z}_t \,,$$

where weight matrix W = diag{*π*(*x*)/[1-*π*(*x*)]} and *z* is the working vector.

The weight of penalty is the same for each row, the same for each column, but are allowed to differ from rows to columns. Figure 4.2 displays the coefficient surface with strong row penalty using a second order penalty on each row and column with large λ_1 and λ_2. Figure 4.3 displays a strong column penalty coefficient surface. In these two figures, the height of the mounds are the coefficient *γ*s. The row (column) penalty minimizes the row (column) difference and forces the *γ*s in each row (column) to follow a linear relation. Since *β* is reexpressed through *γ* and its location, changing *γ*, changes *β.*

4.2.4 Order of Penalty

For small $n = \breve{n} = 3$ in figure 4.1, we name the knots from left to right and from bottom to top as $\gamma_1, \gamma_2, \gamma_3, ..., \gamma_9$. A first order penalty matrix D looks like

$$D_1 = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$$

Each row (column) of Γ gets such a banded D matrix. The complete column penalty has the contrast structure

$$P_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \otimes D_1 = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix},$$

$$P_1 \times \gamma = (\gamma_2 - \gamma_1 \quad \gamma_3 - \gamma_2 \quad \gamma_5 - \gamma_4 \quad \gamma_6 - \gamma_5 \quad \gamma_8 - \gamma_7 \quad \gamma_9 - \gamma_8)'.$$

The row penalty tries to minimize the adjacent row difference.

And the corresponding complete row penalty has the form

$$P_2 = D_1 \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix},$$

$$P_2 \times \gamma = (\gamma_4 - \gamma_1 \quad \gamma_5 - \gamma_2 \quad \gamma_6 - \gamma_3 \quad \gamma_7 - \gamma_4 \quad \gamma_8 - \gamma_5 \quad \gamma_9 - \gamma_6)'.$$

the column penalty tries to minimize the adjacent column difference.

4.2.5 Choosing of Optimal Penalty Parameters

Akaike Information Criteria (AIC) method is widely used for model selection. The lower the AIC is, the better the model is. We search optimal penalty parameters (λ_1, λ_2) by grid search to minimize AIC

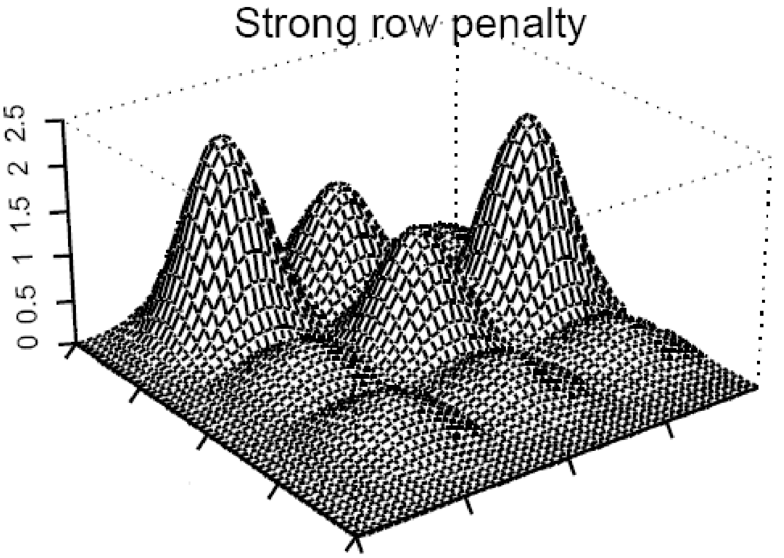

Figure 4.2 Nine cubic *B*-spline tensor products with a strong liner row penalty

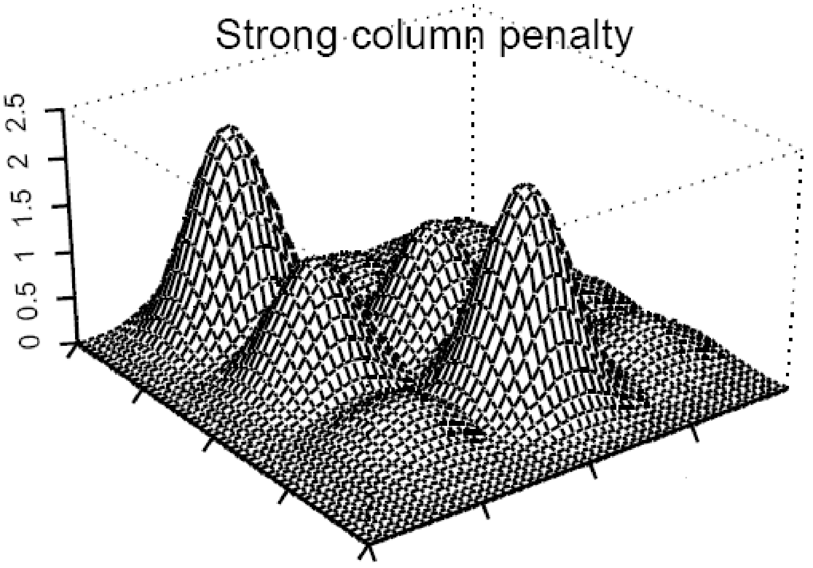

Figure 4.3 Nine cubic *B*-spline tensor products with a strong liner column penalty

$$\text{AIC} = \text{deviance}\,(y;\hat{\gamma}) + 2\text{trace(H)}$$

Upon convergence

$$\text{trace}(H) = \text{trace}\ \{\text{M}'\hat{W}\text{M}(\text{M}'\hat{W}\text{M} + \lambda_1 P'_1\, P_1 + \lambda_2 P'_2\, P_2)^{-1}\}.$$

CHAPTER 5. CONCLUSION

An innovative technique on handwriting digit recognition has been developed based the 2D logistic PSR model. The new method has been tested on the USPS zip code dataset and a 8.97% recognition error rate was achieved. Our approach is to construct a coefficient surface using a rich 2D tensor product B-Spline basis and then purposely overfitting the constructed coefficient surface, while assuming smooth surface. Sensible constraints were imposed to further smoothen the surface, where penalties are added to rows and columns on tensor product coefficients to suppress the roughness of the surface. The optimal penalty weight can be found within several minutes on common PC platform. A careful review on different methods and approaches has been given in this work and the performance of our method is competitive on the same data set up to now. Compared to other methods, our method does not require learning procedure, preprocessing of the input images and expensive and powerful computing platform. A prospective remark on this work has been made.

The following Figure 5.1 to Figure 5.10 show the estimated coefficient surface for digit 0, 1, 2,…,9 respectively. These coefficient surfaces are obtained from training data with optimal penalty weights (λs) which minimize the AIC.

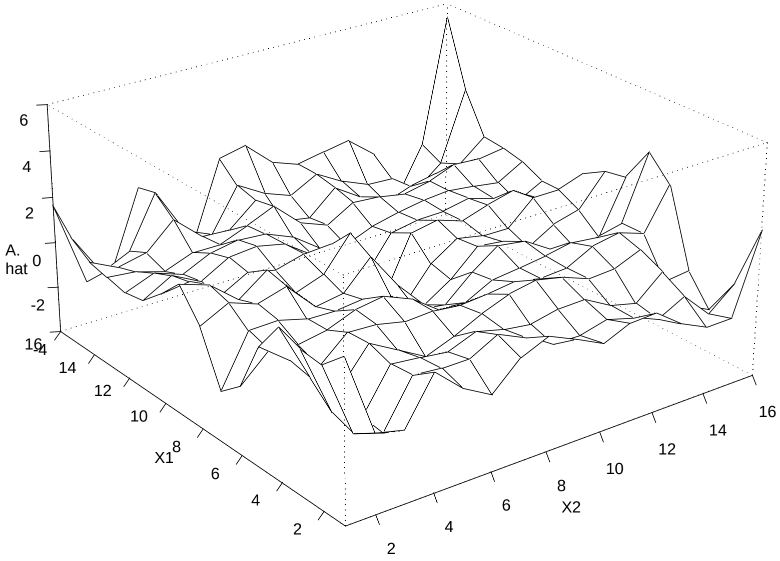

Figure 5.1 Estimated coefficient surface for digit “0” with λ_1=10^{-2}, λ_2=10^{-4}

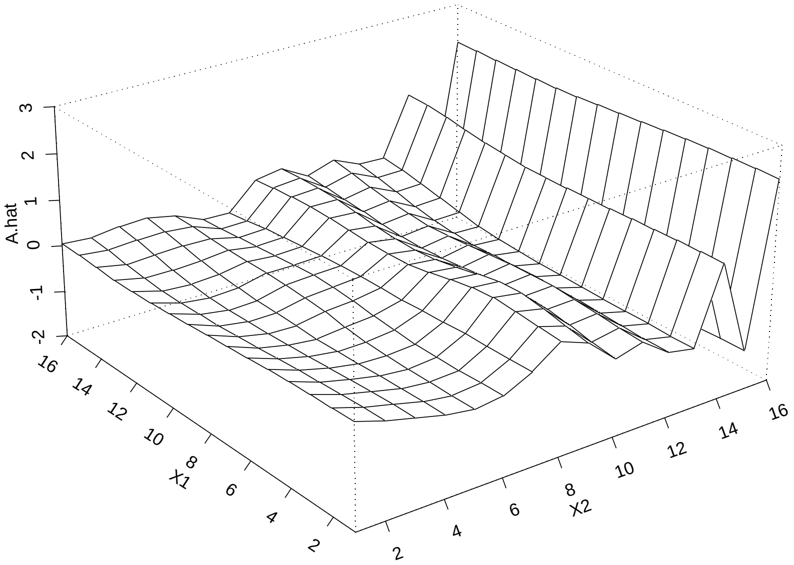

Figure 5.2 Estimated coefficient surface for digit “1” with λ_1=10, λ_2=10^{-6}

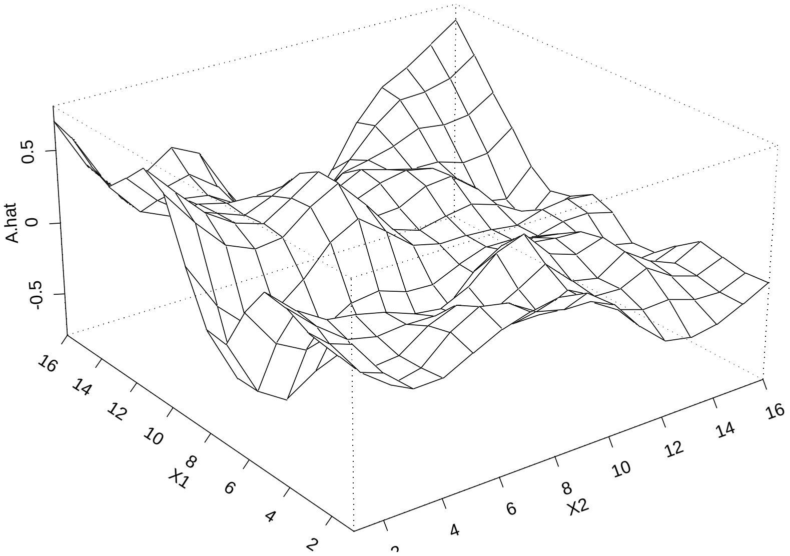

Figure 5.3 Estimated coefficient surface for digit “2” with λ_1=1, λ_2=1

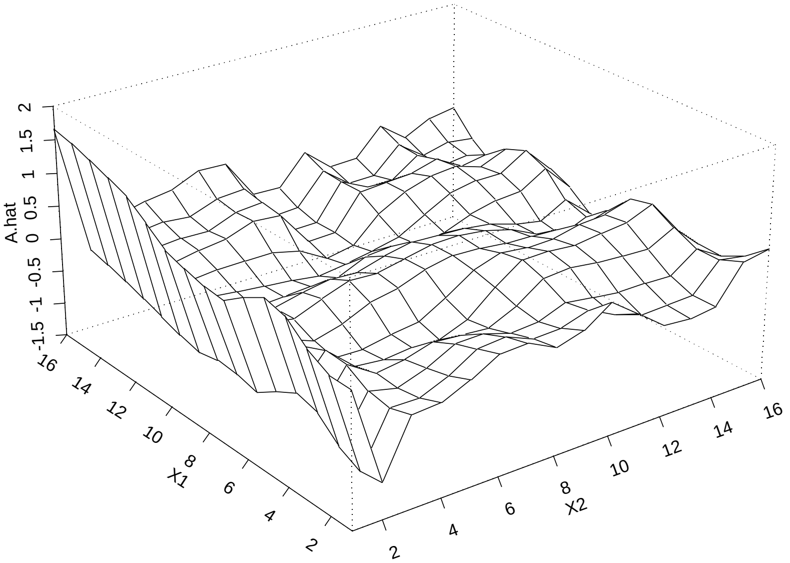

Figure 5.4 Estimated coefficient surface for digit “3” with λ_1=1, λ_2=10^{-4}

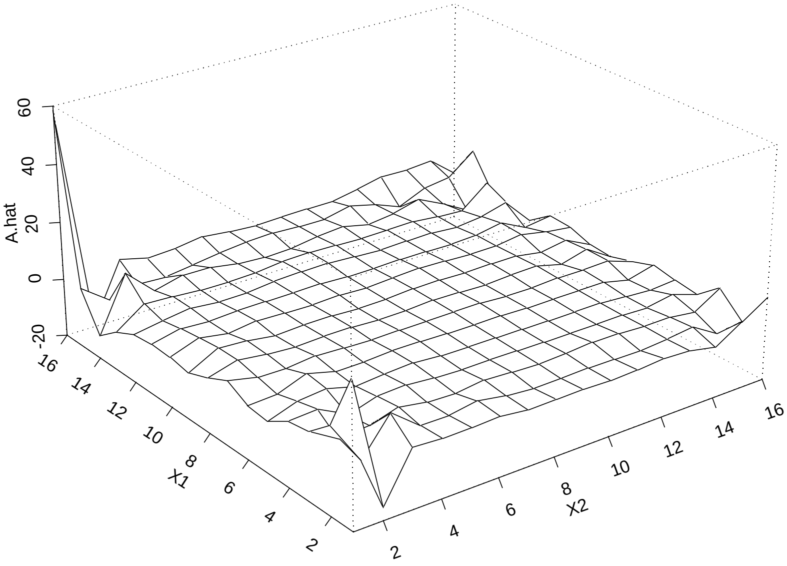

Figure 5.5 Estimated coefficient surface for digit “4”with λ_1=10^{-2}, λ_2=10^{-4}

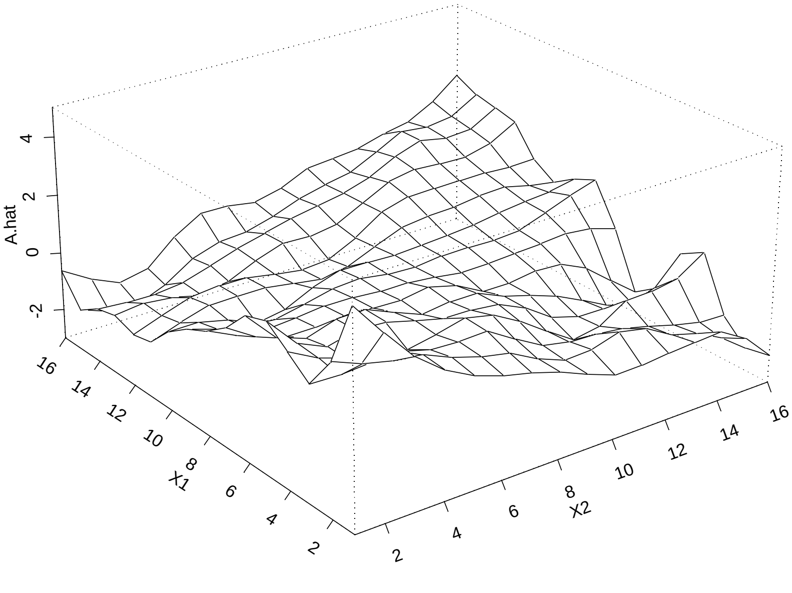

Figure 5.6 Estimated coefficient surface for digit “5” with λ_1=10^{-3}, λ_2=10^{-1}

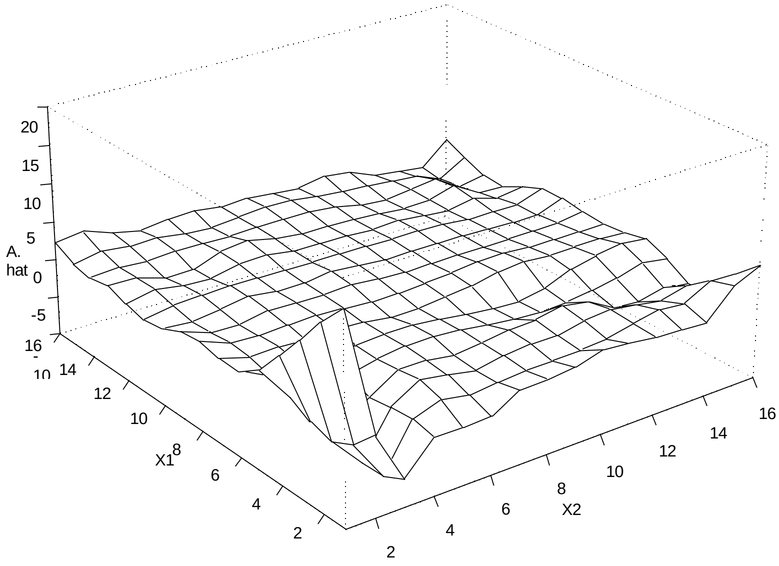

Figure 5.7 Estimated coefficient surface for digit “6”with λ_1=10^{-2}, λ_2=10^{-7}

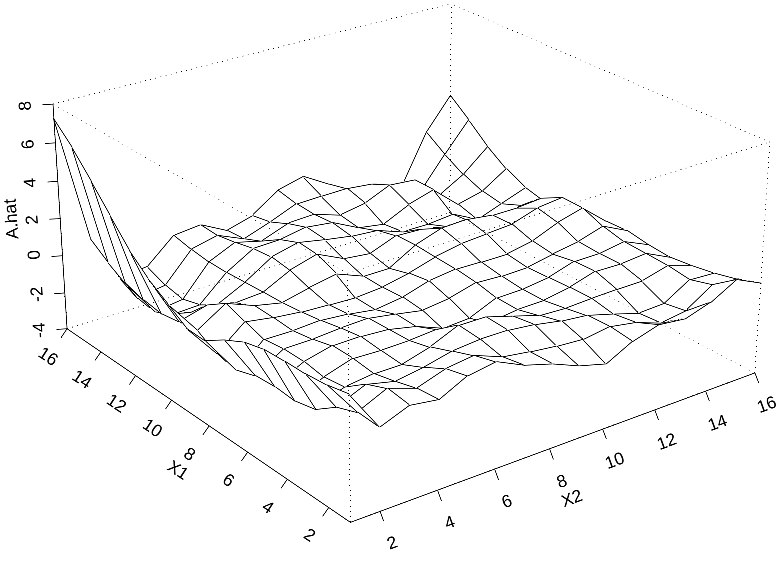

Figure 5.8 Estimated coefficient surface for digit “7” with λ_1=10^{-1}, λ_2=10^{-4}

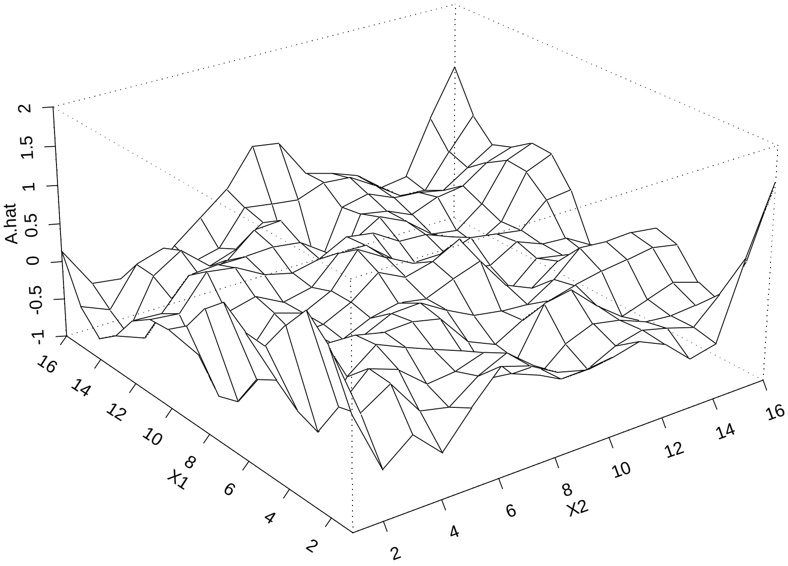

Figure 5.9 Estimated coefficient surface for digit “8” with λ_1=10^{-1}, λ_2=10^{-3}

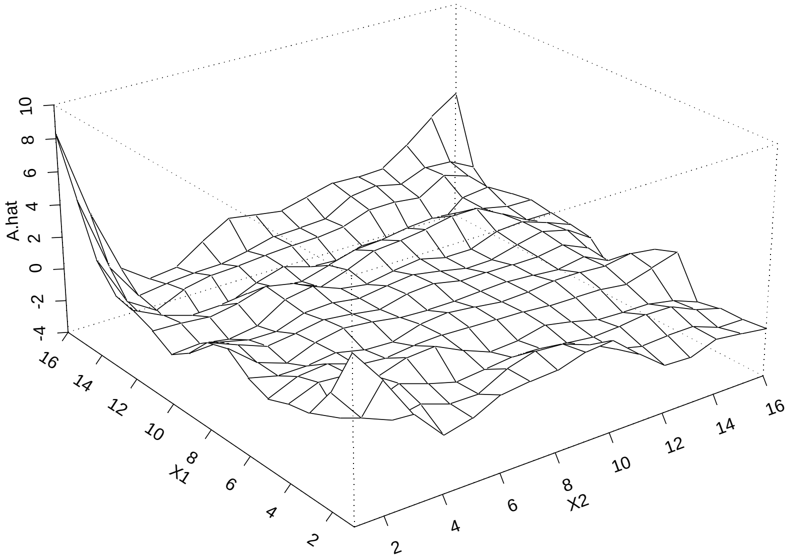

Figure 5.10 Estimated coefficient surface for digit “9” with λ_1=10^{-3}, λ_2=10^{-2}

CHAPTER 6. DISCUSSION

Automatic handwriting recognition has been an active research area for years. During the recent years, many methods have been developed or improved to obtain higher recognition rate on various data sets, such as MNIST, NIST and USPS zip code database. Especially, the USPS dataset has been a very popular benchmark since the first use by the well-known AT&T group led by Le Cun. Many groups have obtained recognition rates on this data set. Based on our review on the previous work done by other groups, the recognition rate on the same data seems a good benchmark of those various methods, including neural networks, genetic algorithm and statistical method. The USPS zip code data is open and free to every one. And this made it so popular in the handwriting recognition community.

Also, this database has reasonable size, not as big as the NIST or MNIST databases. As we know, most of the current recognition algorithms are time and resource consuming, as most of the neural network methods we have reviewed in other sections. The digit images in the USPS database are quite enough for a good performance evaluation for most of the methods we have introduced. In the review section of this work, we listed the recognition rate for Le Cun, et al., Hastie et al. and Parkins et al. as 87%-89% with backpropagation neural network, 91%-92% with statistical method and 88%-90% with genetic algorithm respectively. It seems the performance competition is a very close tie. But, considering the fact that the USPS data set contains several thousand images, the difference is large enough for the purpose of method evaluation. According to the test results, we found the performance of Hastie’s statistical method works the best on the hand written zip code digit recognition and Le Cun’s backpropagation neural network had achieved a score below 90%.

By applying the PSR model, our method has achieved a recognition rate above 91% (the overall error rate is 8.97% including misclassification) that is very close the error rate obtained by Hastie's model. Our model was completely tested on regular PC on Windows XP platform that is famous of its large resource dependence. Every run of the test was finished in just several minutes. And no any learning or further image preprocessing procedure was required. All of these merits of our method can be beneficial when it's applied to hand writing recognition practice. The results we obtained are just for initial benchmark test. There are many ways to further improve the performance of our method. Embedding a fast learning procedure in our method can significantly eliminate many of the misclassifications we have had without any learning and self-adapting capability with the method itself. The learning process can effectively tune up the algorithm and make it immune to many kinds of situations, such as continuous strokes, alternative writing styles and skewing.

Also, we can refine the iteration step length so that more iterations are needed but higher recognition rate can be obtained. We also would like to discuss about some misclassification situations our model has encountered. Some typical examples are shown in the following Figure 6.1.

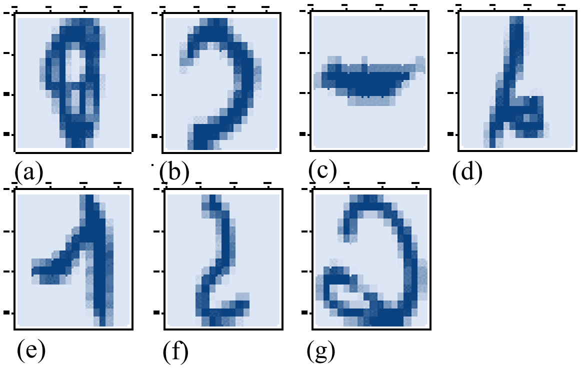

Figure 6.1 Examples of the misclassifications by PSR model.

In Figure 6.1(a) the image is labeled as digit 0, but it's predicted as digit 8 due to the erroneous fragment in the middle of the image. (b) The actual image is digit 0, but it's classified as digit 3 due to the unclosed stroke. (c) The actual image is labeled as digit 0, but it's classified as digit 4. (d) The actual image is labeled as digit 1, but it's classified as digit 6 due to the ambiguous dot caused by extra ink. (e) The actual image is labeled as digit 1, but it's classified as digit 3 due to the exaggeratedly long head. (f) The actual image is labeled as digit 2, but it's classified as digit 8 due to the short starting stroke. (g) The actual image is labeled as digit 2, but it's classified as digit 0.

The examples shown in the above figure indicate some extremely tough situations in the handwriting practice. The images in (e), (f) and (g) are listed here as examples that can be classified finally by improving the model itself or giving more learning time. But the images shown in (a), (b), (c) and (d) are examples that totally cannot be recognized either by human or machine.

Compared to the error rate of 2.5% that human have achieved on the same database, artificial handwriting recognition methods are having a long way to go and that will encourage developing more other new methods.

REFERENCES

Agresti A. (1996). An Introduction to Categorical Data Analysis, *John Wiley& Sons,Inc.*

Baig, A.F. (2004). Spatial-temporal artificial neurons applied to online cursive handwritten character recognition, *ESANN'2004 proceedings - European Symposium on Artificial Neural Networks Bruges (Belgium)*, 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 561-566

Bottou, L., Cortes, C., Denker, J.S., Drucker, H., Guyon, I., Jackel, L.D., LeCun, Y., Muller, U.A., Sackinger, E., Simard, P. and Vapnik, V. (1994). Comparison of classifier methods: a case study in handwritten digit recognition, *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Conference B: Computer Vision & Image Processing.*, (Jerusalem), pp. 77-82.

Breiman, L. (1994). Neural networks: a review from statistical perspective: Comment, *Statistical Science* ,vol. 9, No. 1., pp. 38-42.

Eilers, P.H.C. and Marx, B.D. (2003). Multivariate Calibration with Temperature Interaction Using two-dimensional penalized signal regression. *Chemometrics and Intelligent Laboratory Systems*, 66, pp. 159-174.

Eilers, P.H.C. and Marx, B.D. (2002). Generalized linear additive smooth structures. *Journal of Computational and Graphical Statistics*, vol. 11, No.4, pp.758-783.

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with B-splines and penalties (with comments and rejoinder). *Statistical Science,* 11, pp. 89-121.

Hastie,T., and Simard, P.Y. (1998). Metrics and models for handwritten character recognition, *Statistical Science,* 13.

Hastie,T., Buja, A. and Tibshirani, R. (1995). Penalized discriminant analysis, *the Annals of Statistics,* vol. 23, No. 1., pp. 73-102

Hinton, G.E., Dayan, P. and Revow M. (1997). Modelling the manifolds of images of handwritten digits, *IEEE trans. on Neural Networks,* vol.8, pp. 65-74.

Jain, A.K., Fellow, IEEE, and Zongker, D.(1997). Representation and recognition of handwritten digits using deformable templates, *IEEE Trans. on Pattern Analysis and machine Interll.*, vol. 19, No. 12.

Keysers, D., Dahmen, J., Theiner, T., and Ney, H. (2000). Experiments with an extended tangent distance, *Proceedings 15th International Conference on Pattern Recognition*, (Barcelona, Spain).

LeCun, Y., Jackel, L.D., Bottou, L., Cortes, C., Denker, J.S., Drucker, H., Guyon, I., Muller, U.A., Sackinger, E., Simard, P., and Vapnik, V. (1995). Learning algorithms for classification: a comparison on handwritten digit recognition, *Neural Networks: The Statistical Mechanics Perspective*, (J. H. Oh, C. Kwon, and S. Cho, eds.), pp. 261-276.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., and Jackel, L.D. (1989). backpropagation applied to handwritten zip code recognition, *Neural Computation*, vol. 1, no. 4, pp. 541-551.

Marx, B.D., and Eliers, P.H.C. (2005). Multidimensional penalized signal regression, *Technometrics*, vol. 47, No. 1, pp. 13-22.

Mayraz, G. and Hinton, G.E. (2002). Recognizing handwritten digits using hierarchical products of experts, *IEEE Trans. on Pattern Analysis and machine Interll.*, Vol. 24, No.2.

Parkins, A.D. and Nandi, A.K. (2004). Genetic programming techniques for hand written digit recognition," *Signal processing,* vol. 84, pp.2345-2365.

Scholkopf, B., Simard, P., Smola, A. and Vapnik, V. (1998) Prior knowledge in support vector kernels, *Advances in Neural Inf. Proc. Systems*, vol. 10, pp. 640-646.

Simard, P., Le Cun, Y. and Denker, J. (1993). Effcient pattern recognition using a new transformation distance, *Advances in Neural Information Processing Systems*, vol. 5, pp. 50-58.

Senior, A. W. (1992) Off-line handwriting recognition: A review and experiments, *Technical Report*, Cambridge University Engineering Department.

VITA

The author, Qing Tang, was born in Yiyang, Hunan province, People's Republic of China, on September 26th, 1970. She had finished her elementary and middle school education in Wuhan, Hubei Province, People's Republic of China. She was admitted to the Wuhan University of Science and Technology in September 1988 and awarded her bachelor's degree in electrical engineering in July 1992. After graduation, she started her engineer career in the area of electrical engineering in Wuhan Iron and Steel (Group) Company. In August 2004, she was admitted to the Department of Experimental Statistics in Louisiana State University to pursue her master's degree. She will receive the degree of Master of Applied Statistics in August 2006.