Assaleh, K., & Shanableh, T. (2010). Robust polynomial classifier using L 1-norm minimization. Applied Intelligence, 33(3), 330-339.
The final publication is available at Springer via http://dx.doi.org/ DOI: 10.1007/s10489-009-0169-8

Robust Polynomial Classifier Using *L¹-norm* minimization

K. Assaleh[1] and T. Shanableh[2]
Department of Electrical Engineering [1]
Department of Computer Science and Engineering[2]
American University of Sharjah[1,2]
kassaleh@aus.edu

Abstarct

In this paper we present a robust polynomial classifier based on *L¹-norm* minimization. We do so by reformulating the classifier training process as a linear programming problem. Due to the inherent insensitivity of the *L¹-norm* to influential observations, class models obtained via *L¹-norm* minimization are much more robust than their counterparts obtained by the classical least squares minimization (L^2-*norm*). For validation purposes, we apply this method to two recognition problems: character recognition and sign language recognition. Both are examined under different signal to noise ratio (SNR) values of the test data. Results show that *L¹-norm* minimization provides superior recognition rates over *L²-norm* minimization when the training data contains influential observations especially if the test dataset is noisy.

Keywords:

Polynomial classifier, Multivariate regression, Pattern classification.

1. Introduction

Robust classification is one of the most challenging problems in pattern recognition and its applications. In most pattern recognition applications, near perfect recognition rates can be obtained when the training and test data are acquired using same or similar data acquisition devices under the same or similar environment. Nevertheless, significant reductions in recognition rates are often experienced when the training and test data are mismatched. In general, mismatch between training and test conditions can be due to variability in background noise level and type, and in data source. A prime example is speech recognition where perfect recognition rates can be obtained when both training and test data are collected under the same or similar clean environment. However, when the test is done under noisy conditions or even under clean conditions using a different recording apparatus, recognition rates can drop significantly. Similar examples can be given on image recognition when lighting conditions are changed or when the imaging sensors are varied [1-4].

The problem of robustness can be tackled at the feature extraction level, classification level, or both levels. At the feature extraction (frontend) level, various signal processing techniques and transformations are applied to undo the effect of the signal mismatch between training and test conditions. These techniques vary from one application to another. However, they often include some sort of filtering for either denoising or normalization to combat noise effect and variability of data acquisition devices or transmissions channel effects [4-7].

At the classifier (backend) level, the focus would be on designing a classifier whose parameters exhibit low sensitivity to variations in the test environment for a given class of data, and at the same time maintain good separability across the different classes. For example, in speech recognition, where Hidden Markov Models (HMMs) are the most commonly used classifier, the focus has been on robust statistics and model adaptation and compensation techniques [8, 9].

Perhaps one of the simplest supervised classification methods is based on linear discriminant functions whereby a sequence of feature vectors is linearly mapped into a sequence of class labels. Any multi-class classification problem can be reduced to multiple two-class classification problems. A two-class classifier maps a feature vector into one of two class labels (often assumed as 1 and 0, or 1 and -1). Linear discriminant functions work very well with linearly separable data. However, they fall short when the data is not linearly separable. It should be noted that linearly separable data in clean

conditions can become linearly nonseparable under noisy conditions. As a solution to this problem, many nonlinear classification methods were introduced in the past few decades including neural and statistical classifiers. Amongst the neural classifiers falls the polynomial classifier [10-12] which can be thought of as a network which accepts feature vectors, maps them to a higher dimensional space through a polynomial function and passes the expanded vectors through a single layer network. The weights of this network are obtained through the minimization of the L^2-*norm* of the error between the output of the network and the desired outputs for the training data. This is done explicitly through the 'pseudo-inverse' method. The use of the L^2-*norm* has been the standard practice due to its mathematical tractability which offers a computationally attractive non iterative solution. However, the main problem associated with the L^2-*norm* is its sensitivity to outliers (influential observations) in the training data. This problem can lead to poor recognition rates when the test data is contaminated with noise. In this paper we present a solution to this problem by using the L^1-*norm* as the criteria for solving for the polynomial classifier weights. The hope is that the L^1 based weights are more robust and hence perform better under noisy test conditions. We reformulate the problem of determining the polynomial classifier weights as a linear programming problem. We also show results on character recognition and sign language recognition under noisy conditions where L^1 based recognition results are far more superior to those based on L^2-*norm*.

The paper is organized as follows. Section 2 describes the theory of polynomial classifiers and their training using the L^2-*norm*. It also presents the formulation of the L^1 based training and describes the recognition phase in the polynomial classifier. The application scenarios of character recognition and sign language recognition are presented in Section 3. Section 4 gives a detailed insight justifying the superior results presented in the experimental results of the application scenarios. Finally, concluding remarks are given in Section 5.

2. Polynomial classifiers

A Polynomial classifier is a supervised classifier that is capable of learning complex patterns that could be linearly inseparable. Polynomial classifier have been successfully used in various applications of pattern recognition including speech and speaker recognition [10-12] and biomedical signal separation [13].

A polynomial classifier is a parameterized nonlinear map which nonlinearly expands a sequence of input vectors to a higher dimension and maps them to a desired output

sequence. Consider a K-class pattern recognition problem whose feature vectors are M-dimensional. Each class, i, is represented by a sequence of N_i column vectors $\mathbf{X}_i = [\mathbf{x}_{i,1}\ \mathbf{x}_{i,2}\ \cdots, \mathbf{x}_{i,N_i}]$. In this case, identification requires the decision between K hypotheses $\{H_i\}$. Given an observation feature vector $\mathbf{x} \in R^M$, the Bayes decision rule [14] for this problem is

$$i^{opt} = \arg\max_i p(H_i \mid \mathbf{x}) \qquad i = 1, 2, \cdots, K \tag{1}$$

A common method for solving equation (1) is to approximate an ideal output on a set of training data with a network. That is, if $\{f_i(\mathbf{x})\}$ are discriminant functions [11-22], then we train $f_i(\mathbf{X}_i) = \mathbf{1}_{N_i}$ and $f_{j\neq i}(\mathbf{X}_{j\neq i}) = \mathbf{0}_{N_{j\neq i}}$, where $i, j = 1, 2, \cdots, K$, and $\mathbf{1}_{N_k}$ is a sequence of N_k ones, and $\mathbf{0}_{N_k}$ is a sequence of N_k zeros. If f_i is optimized over all possible functions such that

$$f_i^{opt} = \arg\min_{f_i} E_{\mathbf{x},H}\left\{\left|f_i(\mathbf{x}) - y_i(\mathbf{x}, H)\right|^P\right\} \qquad 1 \le P \le 2, \tag{2}$$

then the solution entails that $f_i^{opt} = p(H_i \mid \mathbf{x})$ [15]. In equation (2), $E_{\mathbf{x},H}$ is the expectation operator over the joint distribution of $\mathbf{x}$ and all hypotheses, and $y_i(\mathbf{x},H)$ is the ideal output for H_i. Thus, the optimization problem gives the functions necessary for the hypothesis test in equation (1). If the discriminant function in (2) is allowed to vary only over a given class (in our case polynomials with a limited degree), then the optimization problem of equation (2) gives an *approximation* of the *a posteriori* probabilities [15]. Using the resulting polynomial approximation in equation (1) thus gives an approximation to the ideal Bayes rule.

Training a Q^{th} order polynomial classifier consists of two main parts. Part one is expanding the training feature vectors via polynomial expansion. The purpose of this expansion is to improve the separation of the different classes in the expanded vector space. Ideally, we aim to have this expansion make all the classes linearly separable. Part two is linearly mapping the polynomial-expanded vectors to an ideal output sequence by minimizing an objective criterion. The mapping parameters represent the weights of the polynomial classifier. These weights are often referred to as the class models.

2.1 Polynomial Expansion

Polynomial expansion of an M-dimensional feature vector $\mathbf{x} = [x_1 \ x_2 \ ... \ x_M]$ is achieved by combining the vector elements with multipliers to form a set of basis functions, **p(x).** The elements of **p(x)** are the monomials of the form

$\prod_{j=1}^{M} x_j^{k_j}$, where k_j is a positive integer, and $0 \le \sum_{j=1}^{M} k_j \le Q$.

Therefore, the Q^{th} order polynomial expansion of an M-dimensional vector **x** generates an M_Q -dimensional vector **p(x)**. M_Q is a function of both M and Q and can be expressed as

$$M_Q = 1 + QM + \sum_{l=2}^{Q} C(M,l) \tag{3}$$

where $C(M,l) = \begin{pmatrix} M \\ l \end{pmatrix}$ is the number of distinct subsets of l elements that can be made out of a set of M elements.

Therefore, for class i the sequence of feature vectors $\mathbf{X}_i = [\mathbf{x}_{i,1} \ \mathbf{x}_{i,2} \cdots, \mathbf{x}_{i,N_i}]^T$ is expanded into

$$\mathbf{V}_i = [\mathbf{p}(\mathbf{x}_{i,1}) \quad \mathbf{p}(\mathbf{x}_{i,2}) \quad \cdots \quad \mathbf{p}(\mathbf{x}_{i,N_i})]^T \tag{4}$$

Notice that while $\mathbf{X}_i$ is a $N_i \times M$ matrix, $\mathbf{V}_i$ is a $N_i \times M_Q$ matrix.

Expanding all the training feature vectors results in a global matrix for all K classes obtained by concatenating all the individual $\mathbf{V}_i$ matrices such that

$\mathbf{V} = [\mathbf{V}_1 \quad \mathbf{V}_2 \quad \cdots \quad \mathbf{V}_K]^T$.

2.2 Solving for the classifier weights

For each class i, the training problem reduces to finding an optimum weight vector. This weight vector is obtained by minimizing the distance between the ideal output vector $\mathbf{y}_i$ and a linear combination of the polynomial expansion of the training feature vectors $\mathbf{V}\ \mathbf{w}_i$ such that

$$\mathbf{w}_i^{opt} = \arg\min_{\mathbf{w}_i} \left\| \mathbf{V}\ \mathbf{w}_i - \mathbf{y}_i \right\|_p \tag{5}$$

The ideal output for the i^{th} class, $\mathbf{y}_i$, is a column vector comprised of ones and zeros such as $\mathbf{y}_i = [\ \mathbf{0}_{N_1}, \mathbf{0}_{N_2}, \ldots, \mathbf{0}_{N_{i-1}}, \mathbf{1}_{N_i}, \mathbf{0}_{N_{i+1}}, \ldots, \mathbf{0}_{N_k}\ ]^{\text{T}}$

Equation (5) indicates that weight vector is obtained by minimizing the *L^p-norm* of the error vector $\mathbf{e}_i = \mathbf{V}\mathbf{w}_i - \mathbf{y}_i$.

a. Solution based on L^2-norm

For the special case of p=2, we arrive at the well-known *L^2-regression* problem. That is, finding $\mathbf{w}_i^{opt}$ that attains the minimum of the *L^2-norm* of the error sequence $\mathbf{e}_i$. Or equivalently, minimizing the square of the *L^2-norm* such as

$$\mathbf{w}_i^{opt} = \arg\min_{\mathbf{w}_i} \|\mathbf{V}\mathbf{w}_i - \mathbf{y}_i\|_2^2 \tag{6}$$

Fortunately, for this particular formulation with the *L^2-norm* there is an explicit formula for the solution $\mathbf{w}_i^{opt}$. This solution can be obtained by applying the normal equations method [16] such as

$$\mathbf{V}^T\mathbf{V}\mathbf{w}_i^{opt} = \mathbf{V}^T\mathbf{y}_i \tag{7}$$

By incorporating equation (5), equation (6) can be rearranged as

$$\sum_{j=1}^{K} \mathbf{V}_j^T\mathbf{V}_j\mathbf{w}_i^{opt} = \mathbf{V}_i^T\mathbf{1}_i \tag{8}$$

If we define $\mathbf{R}_j = \mathbf{V}_j^T\mathbf{V}_j$, $\mathbf{R} = \sum_{j=1}^{K}\mathbf{R}_j$, and $\mathbf{v}_i = \mathbf{V}_i^T\mathbf{1}_i$ then equation (8) yields an explicit solution for $\mathbf{w}_i^{opt}$ expressed as

$$\mathbf{w}_i^{opt} = \mathbf{R}^{-1}\mathbf{v}_i \tag{9}$$

The set $\{\mathbf{w}_i^{opt}\}$ represents the weights of the K polynomial networks which we refer to as the class models.

In [10], Campbell and Assaleh discuss the computational aspects of solving for $\mathbf{w}_i^{opt}$ and they present a fast method for training polynomial networks by exploiting the redundancy of the $\mathbf{R}_j$ matrices. They also discuss in details the computational and storage advantages of their training method.

b. Solution based on L^1-norm

As we indicated earlier, minimizing the *L^2-norm* of the error signal yields an explicit formula of the solution $\mathbf{w}_i^{opt}$. This, of course, is a highly desirable characteristic that

simplifies the computational cost. However, solutions based on *L^2-norm* are known to be problematic when the data that contains influential observations is encountered. *L^2-norm* solutions in such cases tends to be biased towards such influential observations.

L^1-norm is known to provide a more robust solution than that obtained by the *L^2-norm* [17]. This is analogous to the fact that the median gives a more robust estimate of the central tendency of a collection of data points than the arithmetic mean. This is so because *L^1-regression* is less sensitive to outliers/ influential observations than least squares regression (i.e. *L^2-regression*) [18].

L^1-regression is achieved by minimizing the *L^1-norm of* the error vector $\mathbf{V}\mathbf{w}_i - \mathbf{y}_i$. Therefore, the problem is to find $\mathbf{w}_i^{opt}$ in

$$\mathbf{w}_i^{opt} = \arg\min_{\mathbf{w}_i} \left\| \mathbf{V}\mathbf{w}_i - \mathbf{y}_i \right\|_1 \tag{10}$$

Unfortunately, there is no explicit formula for the solution to the *L^1-regression* problem. However, the problem can be reformulated and solved as a linear programming problem. If we denote the elements of the matrix $\mathbf{V}$ as $v_{m\,n}$, the elements of the vector $\mathbf{y}_i$ as $y_{i,m}$, and the elements of the vector $\mathbf{w}_i$ as $w_{i,n}$, it is easy to see that the *L^1-regression* problem:

$$\text{minimize} \sum_m \left| y_{i\,m} - \sum_n v_{m\,n} w_{i\,n} \right|, \tag{11}$$

can be rewritten as

$$\text{minimize} \sum_m t_m$$

$$\text{subject to } t_m - | \, y_{i\,m} - \sum_n v_{m\,n} w_{i\,n} \, | = 0, \quad i = 1, 2, \cdots, m, \tag{12}$$

which is equivalent to the following linear programming problem:

$$\text{minimize} \sum_m t_m$$

$$\text{subject to } -t_m \le y_{i\,m} - \sum_n v_{m\,n} w_{i\,n} \le t_m, \quad i = 1, 2, \cdots, m. \tag{13}$$

Hence, to solve the *L^1-regression* problem it suffices to solve the linear programming problem stated in equation (13) which reduces to solving a system of linear inequalities. A thorough theoretical and computational analysis on the solution of *L^1* inequalities can be found in [17, 19].

2.3 Identification

In the identification stage we are given a sequence of N_c feature vectors $\mathbf{X}_c$ and we are required to determine its class c as one of the enrolled classes in the set $\{1,2,\cdots,K\}$. This is done by two steps: first, expand $\mathbf{X}_c$ into its polynomial basis terms $\mathbf{V}_c = [\mathbf{p}(\mathbf{x}_{c,1}) \quad \mathbf{p}(\mathbf{x}_{c,2}) \quad \cdots \quad \mathbf{p}(\mathbf{x}_{c,N_c})]^T$, and second, evaluate the output sequences against all K models $\{\mathbf{w}_i^{opt}\}$ to obtain a set of score sequences $\{\mathbf{s}_i\}$ such as

$$\mathbf{s}_i = \mathbf{V}_c \mathbf{w}_i^{opt} . \tag{14}$$

The elements of the score sequence $\mathbf{s}_i$ represent the individual scores of each feature vector in the vector sequence $\mathbf{X}_c$. The class of the sequence $\mathbf{X}_c$ is determined by maximizing $\{g(\mathbf{s}_i)\}$ such as

$$c = \arg\max_i (g(\mathbf{s}_i)) \tag{15}$$

where g is a function that outputs a statistic of the sequence $\mathbf{s}_i$ such as the mean or the median. In our case we chose g to compute the mean of $\mathbf{s}_i$ such as

$$g(\mathbf{s}_i) = \frac{1}{N_c} \sum_{j=1}^{N_c} s_{i,j} \tag{16}$$

To summarize, the overall operations of the polynomial classifier are illustrated in Figure 1. Note that this work will be examining the use of both the *L^1-norm* and the *L^2-norm* minimization in the 'Parameter Estimation' block as mentioned above.

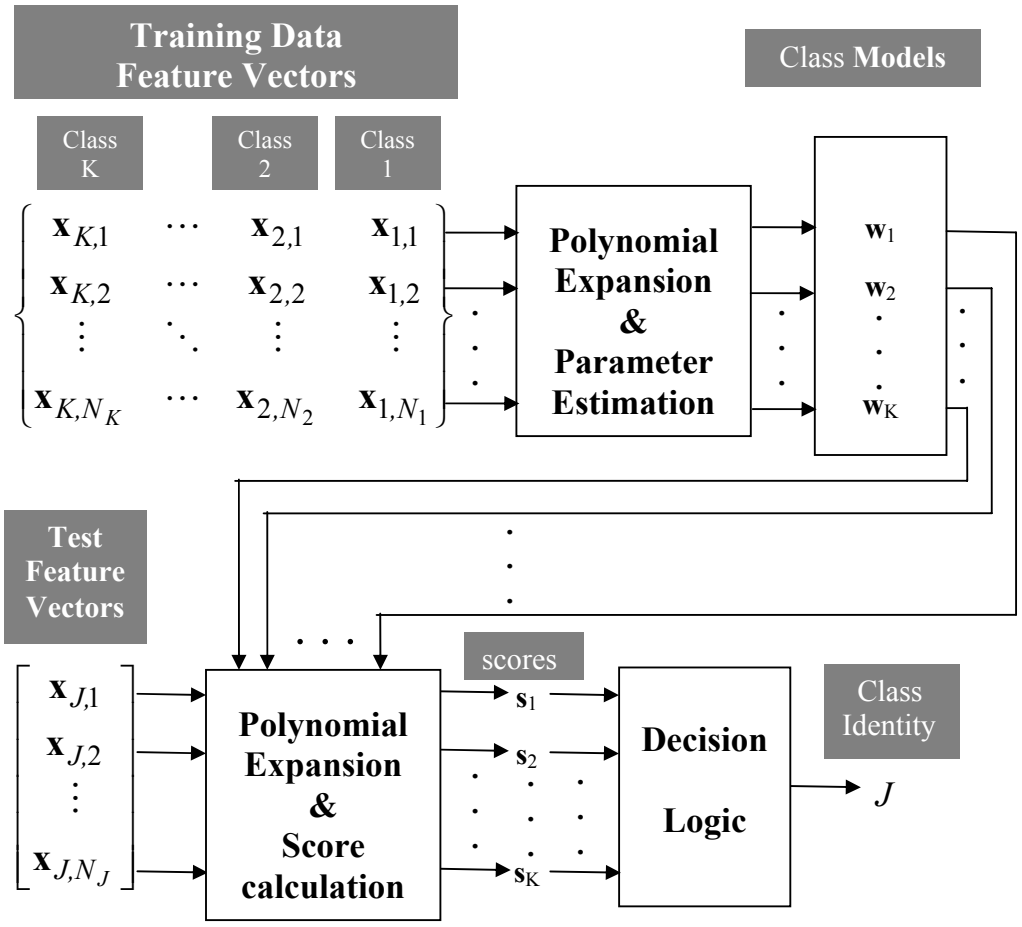


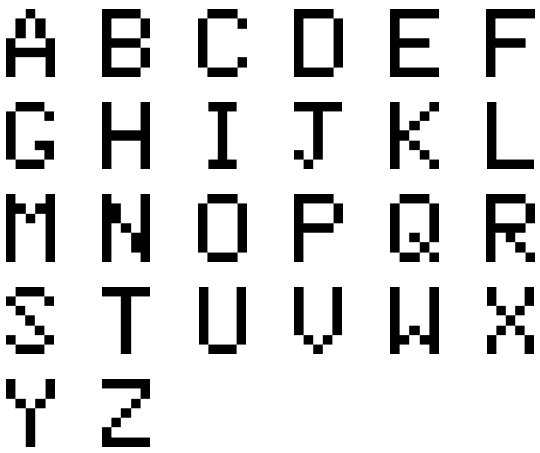
Figure 1. Illustration of the overall operations of the polynomial classifier.

3. Application scenarios

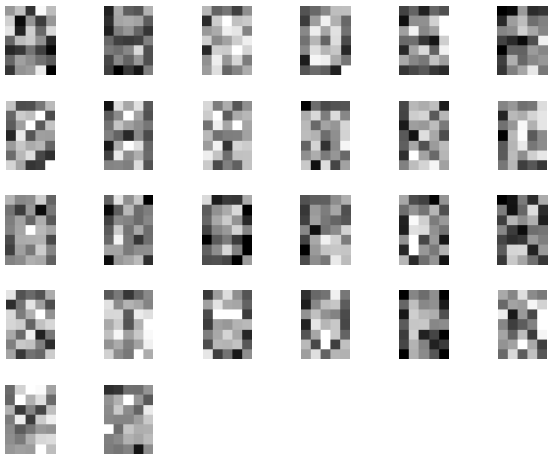
3.1 Optical character recognition

To validate the superiority of the proposed technique over the classical L^2-norm minimization, we compare the performance of the two methods on optical character recognition (OCR) under noisy test conditions. The data is simply obtained by representing each of the 26 English alphabets by a 7x5 bit map as shown in Figure 2(a). As a simple feature extraction method, each 7x5 bit map is unfolded into a 35-dimensional binary feature vector. The feature vectors of the 26 classes are linearly separable, and can be perfectly classified by a simple linear classifier. However, the challenge is when the bit maps are contaminated with noise whereby the vector elements start to be confusable. An example is shown in Figure 2(b) where the bitmaps of the

characters have undergone a noisy channel that imposes an additive Gaussian noise with the same strength of the signal itself (i.e. a 0dB SNR). To asses the performance of the proposed method and compare it to the classical L^2-*norm* based method, we have applied an additive Gaussian noise to the clean set of bitmaps of the characters to obtain SNRs between 20 dB and -10 dB in steps of 5 dB. The noisy data is then used as test data for different polynomial classifiers (i.e. with different polynomial orders) that were trained on the clean dataset. We have experimented with the 1st and 2nd orders using L^2-*norm* minimization and we obtained the performance as shown in Figure 3. The figure clearly shows that the recognition rate deteriorates as the SNR decreases. The figure also shows that 2nd order polynomial performs better than simple 1st order polynomial. First order polynomial classifiers are similar to linear discriminant analysis and they only work well when the classes are linearly separable. The figure also shows the results of experimenting with the polynomial classifier using L^1-*norm* minimization for 1st and 2nd order polynomials. The figure shows the recognition rates obtained for different SNR testing conditions; it vividly shows the superiority of L^1-*norm* minimization over the L^2-*norm* minimization.



(a) OCR clean data



(b) OCR data with Gaussian noise at 0dB SNR

Figure 2. Clean and noisy data used in the character recognition application scenario.

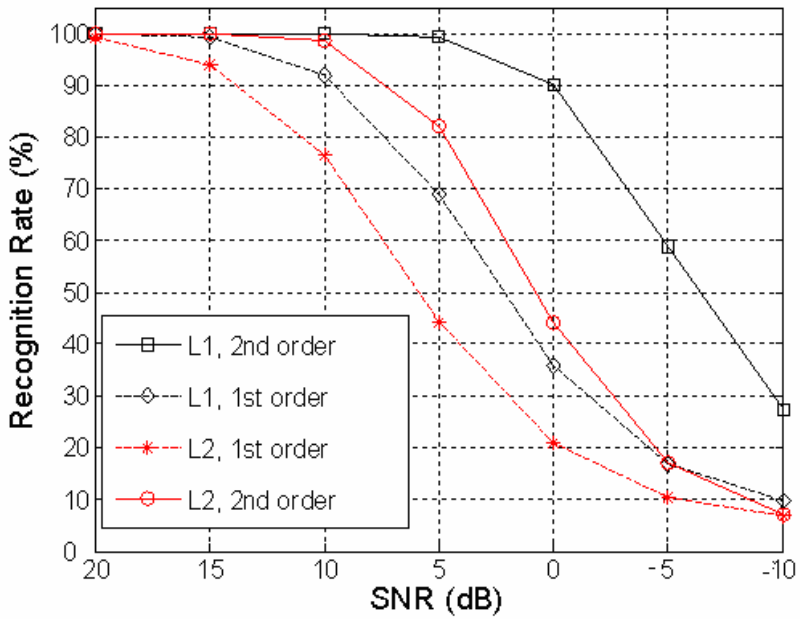


Figure 3. Recognition rates using *L^1-norm* and *L^2-norm* minimizations.

3.2 Online video-based recognition of isolated Arabic sign language gestures.

The performance of the proposed method of *L^1-norm* minimization is also illustrated through an online recognition system of video-based isolated gestures of Arabic sign

language. This section starts with a description of the sign language dataset used in the experiments followed by feature extraction and experimental results.

a. Dataset description

Arabic Sign Language does not yet have a standard database that can be purchased or publicly accessed. Therefore, we decided to use our own ArSL database which we describe in greater details in [20]. The dataset contains 23 Arabic gestured words/phrases collected from 3 different signers. Each of the three signers was asked to repeat each gesture 50 times over three different sessions resulting in a total of 150 repetitions of the 23 gestures which corresponds to 3450 video segments. The signer was videotaped without imposing any restriction on clothing or image background.

b. Sign language feature extraction

We adopt one of the feature extraction techniques that we have previously proposed in [16]. For completeness this section provides a summary of the adopted technique. It was shown that the motion information in a video-based gesture is extracted from the temporal domain of the input image sequence through successive image differencing. Let $I_{g,i}^{(j)}$ denote image index *j* of the i^{th} repetition of a gesture at index *g*. The image formed from the Accummulated Differences (ADs) can be computed by:

$$AD_{g,j} = \sum_{j=1}^{n-1} \partial_j \left(\left| I_{g,j}^{(j)} - I_{g,i}^{(j-1)} \right| \right) \tag{17}$$

Where *n* is the total number of images in the i^{th} repetition of a gesture at index *g*. ∂_j is a binary threshold function of the j^{th} frame.

Radon transformation is applied to the resultant ADs image. As such, the pixel intensities of the ADs image are projected at a given angle θ using the following equation:

$$R_\theta(x) = \int_{-\infty}^{+\infty} f(x'\cos\theta - y'\sin\theta, x'\sin\theta + y'\cos\theta)dy' \tag{18}$$

Where *f* is the input image and the line integral is parallel to the *y'* axis where *x'* and *y'* are given by:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{19}$$

The projected ADs image is then coarsely represented by transforming it into the frequency domain using a 1-D DCT followed by an ideal low pass filter. It was shown in [20] that the number of DCT coefficients to retain in the ideal low pass filter can be determined empirically. Given that the ADs image is projected onto the x-axis, a DCT

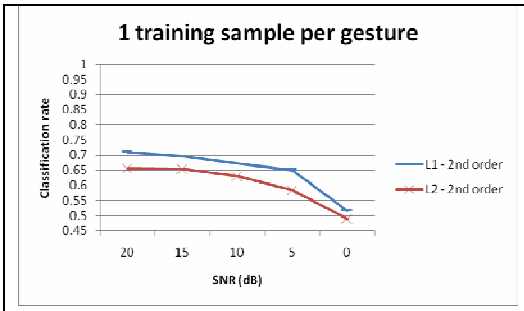
cutoff of 100 was shown to be adequate. The feature vectors entail the retained 100 DCT coefficients.

c. Experimental setup and results

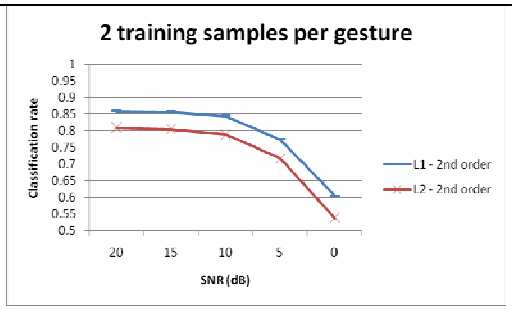
In online recognition systems training is required to be done as the user is enrolled to the system. Normally, this mode involves a few numbers of training samples per class. Hence in this work the polynomial classifiers are trained using few samples per sign language gesture. These samples are selected in a round robin fashion from the training set and the classification results are averaged and reported in this section. The dataset is comprised of three users and is spilt into 70% for training and 30% for testing.

To illustrate the robustness of the proposed L^1*-norm* minimization, we assume that the testing feature vectors are contaminated with Gaussian noise. Such a scenario might arise if the feature vectors are transmitted over wireless or mobile links to be classified at a destination end-system. Considering limited bandwidth communications, transmission of feature vectors would be a great advantage over transmitting row image sequences that represent a sign language gesture.

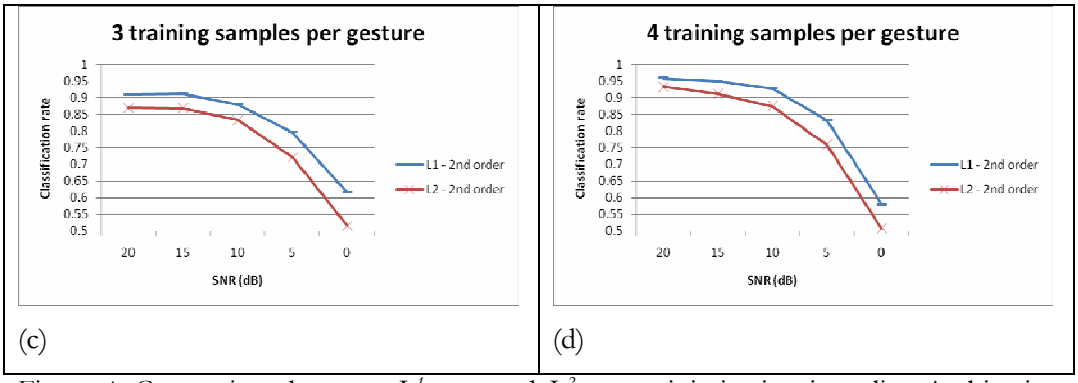
Figure 4 presents the Signal to Noise Ratio (SNR) versus the classification rate using second order polynomial expansion with L^1*-norm* and L^2*-norm* minimization. Our experiments show that the sign language data is not linearly separable hence second order polynomial expansion is used in this experiment.



(a)



(b)



(c) (d)

Figure 4. Comparison between L^1-*norm* and L^2-*norm* minimization in online Arabic sign language recognition.

We experimented with 1 up to 4 training samples per gesture. The classification results are shown in Figure 4 parts (a)-(d) respectively. The figure shows that the proposed L^1-*norm* minimization outperforms its L^2-*norm* counterpart. This statement is true for an SNR range of 0 up to 20 dB. Clearly in this application, an SNR below 0 dB results in very poor classification rates which in some cases fall below 50% and therefore not reported in the figure. In parts (a)-(d) the average classification rate at an SNR of 20 dB for the L^1-*norm* is 85.88% and for the L^2-*norm* is 81.65%. Likewise the average classification rate at an SNR of 0 dB for the L^1-*norm* is 58% and for the L^2-*norm* is 51.3%.

4. Discussion

To further elaborate on the superiority of the L^1-*norm* based training as demonstrated in the previous section, we examine the spread of the training data for both applications. For visualization purposes we project the multidimensional feature set onto one-dimensional set using Fisher Linear Discriminant analysis. The distribution of the projected data is then analyzed using Boxplot diagrams as shown in Figures 5 and 6. Recall that Boxplots are used to show the spread of the data via a box that contains lines at the 25th percentile, median, and the 75th percentile values. Whiskers extend from each end of the box to the most extreme values within 1.5 times the interquartile range from the two ends of the box. Influential observations are data points with values beyond the ends of the whiskers represented by the either "+" or "◊".

For each of the 26 letters in the alphabet Figure 5 shows the spread of the out-of-class projected feature sets (each comprised of 25 values). Note that each of the in-class feature vectors is represented by one projected value shown as a "red" dash.

Clearly, Figure 5 shows that the data does include significant percentage of influential observations. Therefore, and as mentioned previously, *L^2-norm* based training would yield a biased estimate of the separating hyperplane parameters, $\mathbf{w}_i^{opt}$. Whereas the *L^1-norm* based training exhibits robustness against the influential observations yielding a more accurate separating hyperplane as evident in the experimental results shown in Figure 3.

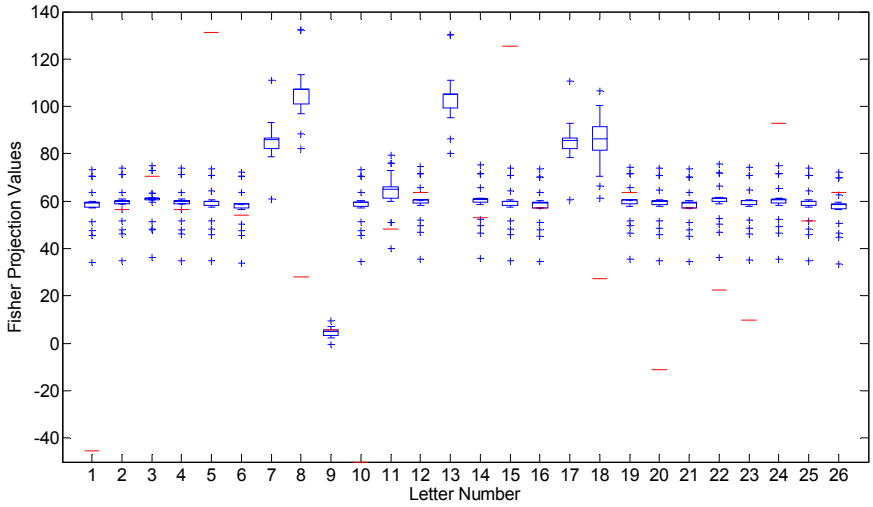


Figure 5. Data spread of the in-class (red) and out-of-class (blue) of the projected feature sets for the OCR dataset.

The spread of the sign language dataset is also examined via the Boxplots and shown in Figure 6. For each of the 23 gestures the figure shows the spread of the in-class and out-of-class projected feature sets. Each of the in-class feature sets is comprised on 105 values (35 per signer) while each of the out-of-class features sets is comprised of 2310 values.

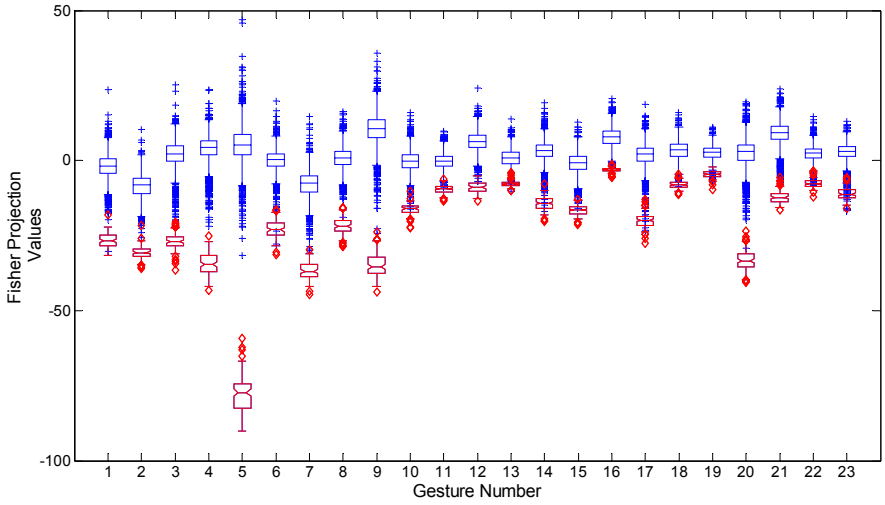


Figure 6. Data spread of the in-class (red) and out-of-class (blue) of the projected feature sets for the sign language dataset.

In comparison to Figure 5 (the spread of the OCR dataset), Figure 6 shows that the sign language dataset includes a smaller percentage of influential observations. As such, the improvement in classification rates using the *L^1-norm* training as opposed to *L^2-norm* training is more advantageous in the case of the OCR dataset.

5. Conclusion

In this paper we have presented a robust polynomial classifier based on *L^1-norm* minimization. We have reformulated the classifier training process as a linear programming problem and took advantage of its inherent insensitivity to influential observations. We have showed that class models obtained via *L^1-norm* minimization are more robust than those obtained by the classical least squares minimization (*L^2-norm*). We applied this method to both character recognition and online recognition of sign language. Results show that *L^1-norm* minimization provides superior recognition rates over *L^2-norm* minimization when the training data contains influential observations. This conclusion was also verified by examining the spread of in-class and out-of-class data using Boxplots of projected feature vectors. The percentage of influential observations in the character recognition scenario was higher than that of the sign language recognition thus the use of *L^1-norm* minimization in the former scenario was more advantageous.

References

[1] C. Sanderson and S. Bengio, "Robust Features for Frontal Face Authentication in Difficult Image Conditions," Proc. IDIAP-RR 03-05, Martigny, Switzerland, 2003.

[2] D. Comaniciu and P. Meer. Robust, "Analysis of Feature Spaces: Color Image Segmentation," Proc. IEEE Conf. on Comp. Vis. and Pattern Recognition, pp. 750-755, Puerto Rico 1997.

[3] Y. Zheng, H. Li, D. Doermann, "Machine printed text and handwriting identification in noisy document images," IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 337.

[4] W.D. Addison and R.H. Glendinning, "Robust image classification," Signal Processing, 86(7), pp. 1488-1501, July 2006.

[5] Q. Zhu and A. Alwan, "Non-linear feature extraction for robust speech recognition in stationary and non-stationary noise," Computer Speech & Language, 17(4), pp. 381-402, October 2003

[6] K. Assaleh and R. Mammone, "LP-Derived Features for Speaker Identification," IEEE Trans. on Speech and Audio Processing, 2(4), pp. 630-638, 1994.

[7] J. Ming T. Hazen J. Glass and D. Reynolds, "Robust Speaker Recognition in Noisy Conditions" IEEE Trans. on Speech and Audio Processing,
15(5), pp. 1711 – 1723, July 2007.

[8] M. Gales and S. Young, "Robust continuous speech recognition using parallel model combination," IEEE Trans. on Speech and Audio Processing, 4(5), pp. 352-359, September 1996.

[9] A. Drygajlo, N. Virag and G. Cosendai, "Robust speech recognition in noise using speech enhancement based on masking properties of the auditory system and adaptive HMM," Proc. of the 4th European Conference on Speech Communication and Technology, Madrid, Spain, pp. 473-476, 1995.

[10] W. M. Campbell,K. T. Assaleh, and C. C. Broun, "Speaker recognition with polynomial classifiers," IEEE Transactions on Speech and Audio Processing , Vol. 10, No. 4, pp 205 -212, 2002.

[11] K. T. Assaleh and W. M. Campbell, "Speaker Identification Using a Polynomial-based Classifier," Proc. of the fourth International Symposium on Signal Processing and its Applications ISSPA'99, Brisbane, Australia, August 1999.

[12] W. M. Campbell and K. T. Assaleh, "Low-Complexity Small-Vocabulary Speech Recognition for Portable Devices," Proc. of the fourth International Symposium on Signal Processing and its Applications ISSPA'99, Brisbane, Australia, August 1999.

[13] K. Assaleh, and H. Al-Nashash, "A Novel Technique for the Extraction of Fetal ECG Using Polynomial Networks," IEEE Trans. on Biomedical Engineering, 52(6), pp. 1148 – 1152, June 2005.

[14] K. Fukunaga, "Introduction to Statistical Pattern Recognition," Academic Press, 1990.

[15] J. Schurmann, "Pattern Classification," John Wiley and Sons, Inc., 1996.

[16] G. Golub and C. Van Loan, "Matrix Computations," John Hopkins, 1989.

[17] J. Cadzow, "Minimum ℓ1, ℓ2, and ℓ∞ Norm Approximate Solutions to an Overdetermined System of Linear Equations," Digital Signal Processing, 12(4), pp. 524-560, October 2002.

[18] R. Vanderbei, "Linear Programming Foundations and Extensions," 2nd ed. Kluwers, 2001.

[19] A. Dax, "The ℓ1 solution of linear inequalities," Computational Statistics & Data Analysis, 50(1), pp. 40-60, January 2006.

[20] T. Shanableh, K. Assaleh, and M. Al-Rousan, "Spatio-temporal Feature Extraction Techniques for Isolated Arabic Sign Language Recognition," IEEE Trans. on Systems, Man and Cybernetics-Part B: Cybernetics, 37(3), June 2007.