Simultaneous Feature Selection and Classifier Training via Linear Programming: A Case Study for Face Expression Recognition

Guodong Guo and Charles R. Dyer
Computer Sciences Department
University of Wisconsin-Madison
Madison, WI 53706
{gdguo, dyer}@cs.wisc.edu

Abstract

A linear programming technique is introduced that jointly performs feature selection and classifier training so that a subset of features is optimally selected together with the classifier. Because traditional classification methods in computer vision have used a two-step approach: feature selection followed by classifier training, feature selection has often been ad hoc, using heuristics or requiring a time-consuming forward and backward search process. Moreover, it is difficult to determine which features to use and how many features to use when these two steps are separated. The linear programming technique used in this paper, which we call feature selection via linear programming (FSLP), can determine the number of features and which features to use in the resulting classification function based on recent results in optimization. We analyze why FSLP can avoid the curse of dimensionality *problem based on margin analysis. As one demonstration of the performance of this FSLP technique for computer vision tasks, we apply it to the problem of face expression recognition. Recognition accuracy is compared with results using Support Vector Machines, the AdaBoost algorithm, and a Bayes classifier.*

1. Introduction

The goal of feature selection in computer vision and pattern recognition problems is to preprocess data to obtain a small set of the most important properties while retaining the optimal salient characteristics of the data. The benefits of feature selection are not only to reduce recognition time by reducing the amount of data that needs to be analyzed, but also, in many cases, to produce better classification accuracy due to finite sample size effects [9].

Most feature selection methods involve evaluating different feature subsets using some criterion such as probability of error [9]. One difficulty with this approach when applied to real problems with large feature dimensionality, is the high computational complexity involved in searching the exponential space of feature subsets. Several heuristic techniques have been developed to circumvent this problem, for example using the branch and bound algorithm [6] with the assumption that the feature evaluation criterion is monotonic. Greedy algorithms such as sequential forward and backward search [6] are also commonly used. These algorithms are obviously limited by the monotonicity assumption. Sequential floating search [18] can provide better results but at the cost of higher search complexity. Jain and Zongker [9] evaluated different search algorithms for feature subset selection and found that the sequential forward floating selection (SFFS) algorithm proposed by Pudil *et al.* [18] performed best. However, SFFS is very time consuming when the number of features is large. For example, Vailaya [21] used the SFFS method to select 67 features from 600 for a two-class problem and reported that SFFS required 12 days of computation time.

Another issue associated with feature selection methods is the *curse of dimensionality*, i.e., the problem of feature selection when the number of features is large but the number of samples is small [9]. This situation is common in many computer vision tasks such as object recognition because there are often less than tens of training samples (images) for each object, but there are hundreds of candidate features extracted from each image.

Yet another difficult problem is determining how many features to select for a given data set. Traditional feature selection methods do not address this problem and require the user to choose the number of features. Consequently, this parameter is usually set without a sound basis.

Recently, a new approach to feature selection was proposed in the machine learning community called *Feature Selection via Concave Minimization* (FSV) [3]. The basic idea is to jointly combine feature selection with the classi-

1063-6919/03 $17.00 © 2003 IEEE

fier training process using a linear programming technique. The results of this method are (1) the number of features to use, (2) which features to use, and (3) the classification function. Thus this method gives a complete and optimal solution.

In order to evaluate how useful this method may be for problems in computer vision and pattern recognition, this paper investigates its performance using the face expression recognition problem as a testbed. 612 features were extracted from each face image in a database and we will evaluate if a small subset of these features can be automatically selected without losing discrimination accuracy. Success with this task will encourage future use in other object recognition problems as well as other applications including perceptual user interfaces, human behavior understanding, and interactive computer games.

The feature selection via linear programming (FSLP) formulation is presented in next section. We analyze why this formulation can avoid the *curse of dimensionality* problem in Section 3. Then we describe the face expression recognition problem and the feature extraction method used in Section 4. The FSLP method is experimentally evaluated in Section 5 and results are compared with Support Vector Machines, AdaBoost, and a Bayes classifier.

2. Linear Programming Formulation

In the early 1960s, the linear programming (LP) technique [13] was used to address the pattern separation problem. Later, a robust LP technique was proposed to deal with linear inseparability [2]. Recently, the LP framework has been extended to cope with the feature selection problem [3]. We briefly describe this new LP formulation below.

Given two sets of points $\mathcal{A}$ and $\mathcal{B}$ in R^n, we seek a linear function such that $f(x) > 0$ if $x \in \mathcal{A}$, and $f(x) \leq 0$ if $x \in \mathcal{B}$. This function is given by $f(x) = w'x - \gamma$, and determines a plane $w'x = \gamma$ with normal $w \in R^n$ that separates points $\mathcal{A}$ from $\mathcal{B}$. Let the set of m points, $\mathcal{A}$, be represented by a matrix $A \in R^{m \times n}$ and the set of k points, $\mathcal{B}$, be represented by a matrix $B \in R^{k \times n}$. After normalization, we want to satisfy

$$Aw \geq e\gamma + e, \;\; Bw \leq e\gamma - e \tag{1}$$

where e is a vector of all 1s with appropriate dimension. Practically, because of overlap between the two classes, one has to minimize some norm of the average error in (1) [2]:

$$\begin{aligned} \min_{w,\gamma} f(w,\gamma) = min_{w,\gamma} \quad & \tfrac{1}{m} \parallel (-Aw + e\gamma + e)_+ \parallel_1 \\ & + \tfrac{1}{k} \parallel (Bw - e\gamma + e)_+ \parallel_1 \end{aligned} \tag{2}$$

where x_+ denotes the vector with components $\max\{0, x_i\}$. There are two main reasons for choosing the 1-norm in Eq. (2): (i) it is easy to formulate as a linear program (see (3) below) with theoretical properties that make it computationally efficient [2], and (ii) the 1-norm is less sensitive to outliers such as those occurring when the underlying data distributions have pronounced tails [3].

Eq. (2) can be modeled as a so-called robust linear programming (RLP) problem [2]:

$$\begin{aligned} \min_{w,\gamma,y,z} \quad & \tfrac{e'y}{m} + \tfrac{e'z}{k} \\ subject\ to \quad & -Aw + e\gamma + e \leq y, \\ & Bw - e\gamma + e \leq z, \\ & y \geq 0, z \geq 0. \end{aligned} \tag{3}$$

which minimizes the average sum of misclassification errors of the points to two bounding planes, $x'w = \gamma + 1$ and $x'w = \gamma - 1$, where "$'$" represents transpose.

Problem (3) solves the classification problem without considering the feature selection problem. In [3] a feature selection strategy was integrated into the objective function in order to simultaneously select a subset of the features. Feature selection is defined by suppressing as many components of the normal vector w to the separating plane P as needed to obtain an acceptable discrimination between the sets $\mathcal{A}$ and $\mathcal{B}$. To accomplish this, they introduced an extra term into the objective function of (3), reformulating it as

$$\begin{aligned} \min_{w,\gamma,y,z} \quad & (1-\lambda)\left(\tfrac{e'y}{m} + \tfrac{e'z}{k}\right) + \lambda e'|w|_* \\ subject\ to \quad & -Aw + e\gamma + e \leq y, \\ & Bw - e\gamma + e \leq z, \\ & y \geq 0, z \geq 0. \end{aligned} \tag{4}$$

where $|w|_* \in R^n$ has components equal to 1 if the corresponding components of w are nonzero, and has components equal to 0 if the corresponding components of w are 0. So, $e'|w|_*$ is actually a count of the nonzero elements in the vector w. This is the key to integrating feature selection with the classifier training process. As a result, Problem (4) balances the error in discrimination between two sets $\mathcal{A}$ and $\mathcal{B}$, $\frac{e'y}{m} + \frac{e'z}{k}$, and the number of nonzero elements of w, $e'|w|_*$. Moreover, if an element of w is 0, the corresponding feature is removed. Thus only the features corresponding to nonzero components in the normal w are selected after linear programming optimization.

In [3] a method called *Feature Selection via Concave Minimization* (FSV) was developed to deal with the last term in the objective function of (4). They first introduced a variable v to eliminate the absolute value in the last term by replacing $e'|w|_*$ with $e'v_*$ and adding a constraint $-v \leq w \leq v$, which models the vector $|w|$. Because the step function $e'v_*$ is discontinuous, they used a concave

exponential to approximate it, $v_* \approx t(v,\alpha) = e - \varepsilon^{-\alpha v}$, in order to get a smooth solution. This required introduction of an additional parameter, α. Alternatively, instead of computing the concave exponential approximation, we will use a simple term $e's$ with only one parameter, μ. This produces the final formulation, which we call *Feature Selection via Linear Programming* (FSLP):

$$\begin{array}{rl} \min\limits_{w,\gamma,y,z} & \left(\frac{e'y}{m} + \frac{e'z}{k}\right) + \mu e's \\ subject\ to & -Aw + e\gamma - y \leq -e, \\ & Bw - e\gamma - z \leq -e, \\ & -s \leq w \leq s, \\ & y, z \geq 0. \end{array} \tag{5}$$

Our FSLP formulation in (5) is slightly different from the FSV method in that FSLP is simpler to optimize and is easier to analyze in relation to the margin, which we do in Section 3. It should be noted that the normal of the separating hyperplane w in (5) has a small number of non-zero components (about 18) and a large number of 0 components (594) in our experiments. The features corresponding to the 0 components in the normal vector can be discarded, and only those with non-zero components are used. As a result, no user-specified parameter is required to tell the system how many features to use.

3. Avoiding the Curse of Dimensionality

In [3] the authors did not address the issue of the *curse of dimensionality*. Instead, they focused on developing the FSV method to get a smooth solution, which is not explicitly connected with the margin analysis as we do here. Also, their experiments used data sets in which the number of examples was much larger than the number of feature dimensions. Here we will show that our FSLP method is actually related to margin maximization, which makes it possible to avoid the *curse of dimensionality* problem [9].

Consider the last term, $e's$, in the objective function of (5), where s is the absolute value of the normal w due to the constraint $-s \leq w \leq s$. To minimize the objective function in (5) requires minimizing the term $e's$ too. Since

$$e's = \sum_i s_i = \sum_i |w_i| = \| w \|_1 \tag{6}$$

this means minimizing $\| w \|_1$, which is the 1-norm of the normal w. Because minimizing $\| w \|_1$ is equivalent to maximizing $\frac{1}{\|w\|_1}$, the objective function in (5) maximizes $\frac{1}{\|w\|_1}$.

Recall from Eq. (1) there are two bounding hyperplanes, $P1 : w'x - \gamma = 1$ and $P2 : w'x - \gamma = -1$. The discriminating hyperplane P is midway between these two hyperplanes, i.e., $w'x - \gamma = 0$. The distance of any point x to the hyperplane P is defined as $d(x;P) = \frac{|w'x-\gamma|}{\|w\|_2}$. From Eq. (1) $|w'x - \gamma| \geq 1$, so any point, x, that is outside the two bounding hyperplanes, $P1$ and $P2$, satisfies $d(x;P) \geq \frac{1}{\|w\|_2}$.

The minimum distance between the two bounding hyperplanes is $\frac{2}{\|w\|_2}$, which is defined as the margin, similar to that used in developing SVMs [22]. We know that the p-norm is non-increasing monotonic for $p \in [1,\infty]$, so $\| w \|_1 \geq \| w \|_2, \forall w \in R^n$, which is equivalent to

$$\frac{1}{\| w \|_1} \leq \frac{1}{\| w \|_2}. \tag{7}$$

Also, the p-norm $\| w \|_p$ is convex on R^n, $\forall p \in [1,\infty]$ [19]. So, by maximizing $\frac{1}{\|w\|_1}$, we approximately maximize $\frac{2}{\|w\|_2}$. As a result, the last term, $e's$, in the objective function of (5) has the effect of maximizing the margin.

Maximizing the margin can often circumvent the *curse of dimensionality* problem, as seen in Support Vector Machines, which can classify data in very high-dimensional feature spaces [22]. Our FSLP method has a similar advantage because it incorporates a feature selection process based on margin size.

In fact, when $\mu = 0$ the last term in the objective function of (5) disappears. In this case classification performance worsens (we do not describe this case in Section 5 formally) because the remaining two terms do not have the property of maximizing the margin. So, the last term, $e's$, has two effects: (i) feature selection, and (ii) margin maximization.

Because the *curse of dimensionality* problem occurs in so many computer vision tasks, our analysis that FSLP circumvents this problem is an important new result. Further demonstration of this property is shown empirically in Section 5.

4. Face Expression Recognition

Face expression recognition is an active research area in computer vision [11] [12] [25] [24]. See [16] for a recent review of methods for face expression recognition.

In this paper we investigate face expression recognition from static images using Gabor filters for facial feature extraction. Several researchers [11] [12] [25] [24] have demonstrated the advantages of using Gabor wavelet coefficients [5] to code facial expressions.

A two-dimensional Gabor function, $g(x,y)$, and its Fourier transform, $G(u,v)$, can be written as

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi jWx\right] \tag{8}$$

$$G(u,v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\} \tag{9}$$

where W is the frequency of a sinusoidal plane wave along the x-axis, and σ_x and σ_y are the space constants of the Gaussian envelope along the x and y axes, respectively. $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$. Filtering a signal with this basis provides a localized frequency characterization. Filters with arbitrary orientation can be obtained by a rotation of the x-y coordinate system.

In our experiments, each input face image was convolved with 18 Gabor filters (3 scales and 6 orientations), resulting in 18 filtered images. The amplitudes of each filtered image at selected fiducial points were used as feature vectors. Thus, for each face image, the extracted feature vector was length 612 (34x3x6) when 34 fiducial points were used. Typical positions of the fiducial points are shown in Figure 1.

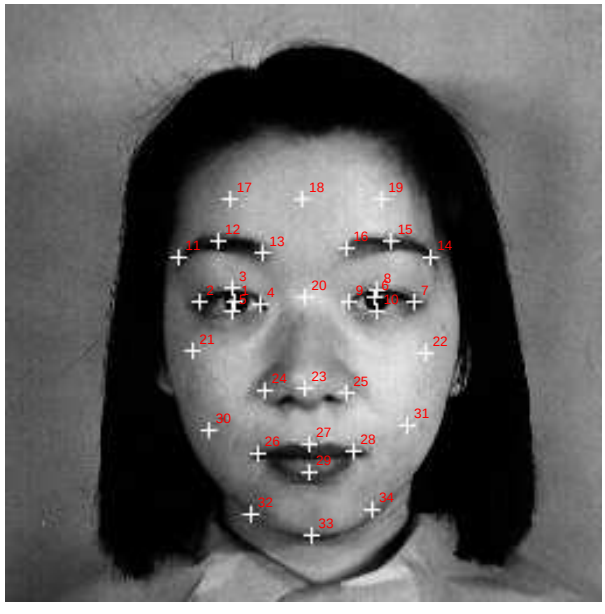

Figure 1. 34 fiducial points on a face image.

5. Experimental Evaluation

5.1. Face Expression Database

The face expression database [11] used in our experiments contains 213 images of 10 Japanese women. Each person has two to four images for each of seven expressions: neutral, happy, sad, surprise, anger, disgust, and fear. Each image size is 256 x 256 pixels. A few examples are shown in Figure 2. For more information on the database such as image collection, data description, and human ranking, see [11]. This database was also used in [12] [25] [24].

5.2. Experimental Results

Our experimental procedure used 10-fold cross-validation because the database contains only 213 images. That is, the database was divided randomly into ten roughly equal-sized parts, from which the data from nine parts were used for training the classifiers and the last part was used for testing. We repeated this procedure ten times so that each part was used once as the test set.

Experimentally we found that the parameter μ in (5) is best set to a small value, and we used $\mu = 0.00001$ in all experiments. To solve this 7-expression classification problem we used a simple binary tree tournament scheme with pairwise comparisons.

Table 1. The performance of FSLP compared to linear SVM (L-SVM) and GRBF non-linear SVM (NL-SVM) using 10-fold cross-validation. The average number of selected features (Ave. #) for each pairwise classifier and the total number of selected features (Total #) used for all pairs are shown in addition to the number of errors out of 21 test examples in each run.

Test	Ave. #	Total #	FSLP	L-SVM	NL-SVM
Set 1	16.8	82	3	2	1
Set 2	17.0	84	2	2	2
Set 3	17.1	90	1	1	2
Set 4	16.4	92	3	3	3
Set 5	16.0	83	1	2	2
Set 6	19.1	102	2	2	2
Set 7	16.9	85	2	2	2
Set 8	17.2	91	1	0	0
Set 9	17.5	91	2	1	2
Set 10	17.4	89	2	1	1
Ave.	17.1	88.9	1.9	1.6	1.7

Experimental results of the FSLP method are shown in Table 1. Feature selection was performed for each pair of classes, resulting in a total of 21 pairs for the 7-expression classification problem. The second column in Table 1 shows the number of selected features on average over the 21 pairwise classifiers, ranging from 16.0 to 19.1 for the ten runs. The average number of selected features over the ten runs was 17.1. Thus a very sparse set of features was automatically selected out of the 612 features extracted from each face image. This demonstrates that FSLP can significantly reduce the number of feature dimensions, and without any user interaction.

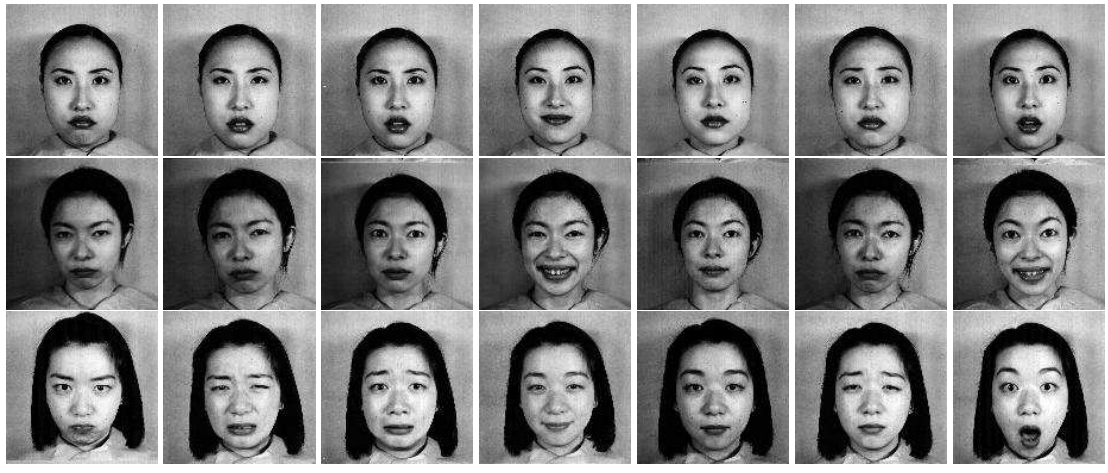

Figure 2. Some images in the face expression database. From left to right, the expressions are angry, disgust, fear, happy, neutral, sad, and surprise.

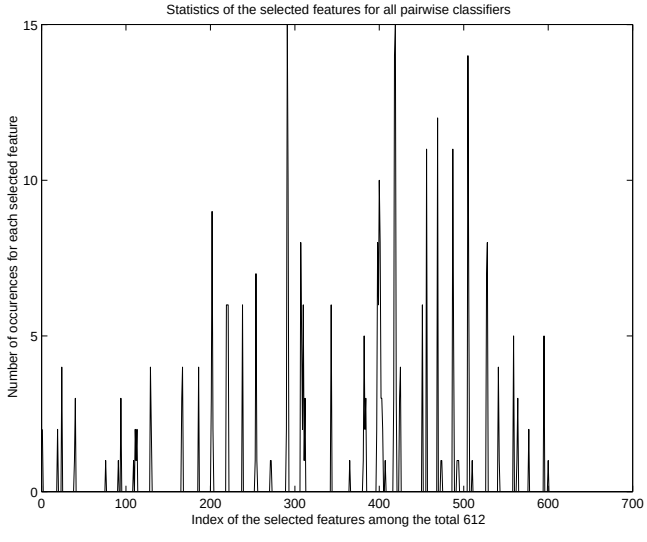

Figure 3. Histogram of the frequency of occurrence of the 612 features used in training Set 1 for all 21 pairwise FSLP classifiers.

The third column in Table 1 shows the total number of features selected by FSLP for all 21 pairwise classifiers in each test set. Because some features are useful in discriminating between one pair, say, "angry" and "happy," but not for separating another pair, say "angry" and "sad," the number of features selected for all pairs is larger than that for each pair. For instance, there were 82 selected features for 21 pairwise classifiers in Set 1. This number is still much smaller than all 612 features. On the other hand, the frequency of occurrence of the 82 features over all pairs of classes was very variable, as shown by the histogram in Figure 3.

Column 4 in Table 1 lists the number of classification errors out of 21 test examples by FSLP on each data set. The average over 10 runs was 1.9.

5.3. Comparison with SVMs

In order to verify whether the FSLP method has good performance or not in terms of recognition accuracy, we compared it with some other methods. Support Vector Machines [22] are known to give high recognition accuracy in practice, so we first compared FSLP with SVMs. The classification errors of both linear or non-linear SVMs (using all 612 features without selection) in each run are shown in columns 5 and 6 of Table 1. For the non-linear SVM, we used the GRBF kernel and experimentally set the width parameter to its best value. The maximum error of FSLP was 3 over the 10 runs, which was never larger than the errors by linear SVMs and non-linear SVMs. The average number of errors over 10 runs was very similar for FSLP, linear SVM (1.6 errors) and non-linear SVM (1.7 errors). The corresponding recognition accuracies of the three methods were 91.0%, 92.4%, and 91.9%, respectively (see Table 2), which are comparable. Notice, however, that the average number of features selected by FSLP was 17.1, much less than that used by the SVMs. Furthermore, the computation time of FSLP was fast in both the training and recognition phases, with run times of several minutes to train all 21 classifiers on a Linux machine with a 1.2 GHz Pentium processor using a Matlab implementation and CPLEX 6.6 for the standard linear programming optimization.

While the recognition accuracy of SVMs is comparable to FSLP, one major weakness of SVMs is their high computational cost, which precludes real-time applications. In addition, SVMs are formulated as a quadratic programming problem and, therefore, it is difficult to use SVMs to do feature selection directly. (Some researchers have proposed approximations to SVM for feature selection [23] [4] by first training the SVM using the whole training set, and then computing approximations to reduce the number of features. This two-step approach cannot guarantee selection of the best feature subset, however.) Finally, SVM approximations [23] [4] cannot determine automatically how many features to use. On the contrary, FSLP addresses all of these issues at once.

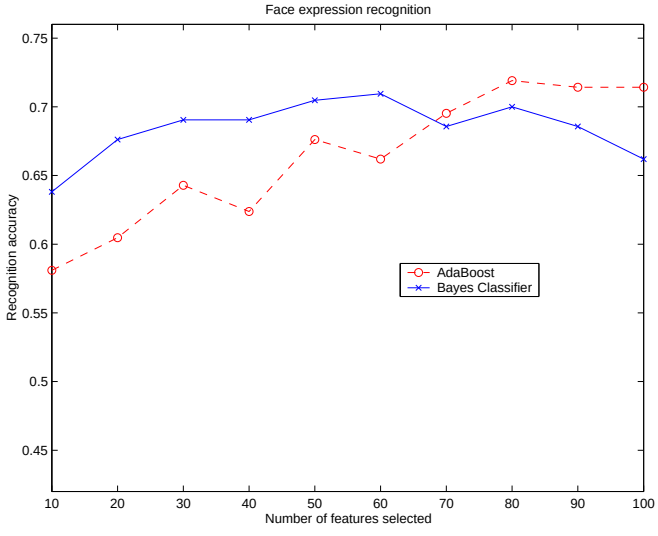

Figure 4. Recognition accuracies of a Bayes classifier and Adaboost as a function of the number of features selected.

5.4. Comparison with AdaBoost and Bayes

Because one of our main goals was an evaluation of FSLP's feature selection process, we also compared the method with some greedy and heuristic methods for feature selection. The AdaBoost method [20] uses a greedy strategy to select features in the learning phase. Greedy feature selection can also be used with a Bayes classifier by assuming feature independence and incrementally adding the most discriminating feature [10]. Figure 4 shows the recognition performance of the AdaBoost and Bayes classifiers as a function of the number of features selected. It is clear that less that 100 features are sufficient for both algorithms. The Bayes classifier reached its best performance of 71.0% with 60 features, and the performance deteriorated slightly if more features were used. The recognition accuracy of the Bayes classifier was 63.3% (shown in Table 2) when all 612 features were used. Overfitting the training data is a serious problem for the Bayes method, so feature selection is necessary for it. Nevertheless, a simple greedy method does not give Bayes much better accuracy. For the AdaBoost method, peak performance was 71.9% using 80 features (see Table 2) for each pair of classes. As shown in Figure 4, using more features slightly lowered recognition accuracy. In summary, both the AdaBoost and Bayes classifiers combined with a greedy feature selection strategy need to use a larger number of features than FSLP, and their recognition accuracies are much worse than FSLP.

5.5. Comparison with Neural Nets and LDA

We also compared the recognition performance of FSLP with other published methods [25] [24] [12] that used the same database. In [25] [24] a Neural Network was used with 90.1% recognition accuracy. When some problematic images in the database were discarded, the accuracy was 92.2%. In [12] a result of 92% using linear discriminant analysis (LDA) was reported, but they only included nine people's face images and, hence, only 193 of the 213 images were used. In conclusion, FSLP gives comparable results to Neural Network and LDA methods, but FSLP optimally selects a small number of features automatically, which is especially important for real-time applications.

6. Concluding Remarks

This paper introduced a linear programming technique called FSLP for accomplishing jointly-optimal feature selection and classifier training, and demonstrated its performance for face expression recognition. There are four main properties of this new technique that make it advantageous over existing methods: (1) FSLP can determine how many features to use automatically without any user interaction; (2) FSLP gives high recognition performance, comparable with linear SVMs, non-linear SVMs, Neural Networks, and LDA, and much better than AdaBoost and Bayes classifiers; (3) FSLP avoids the *curse of dimensionality* problem, which often occurs when the amount of training data is small [9]; and (4) FSLP feature selection is fast to compute.

Acknowledgments

The authors thank Olvi Mangasarian and Steven Wright for their help on the linear programming technique, and M. Lyons for providing the face expression database. The support of the NSF under Grant No. IIS-9988426 is gratefully acknowledged.

Table 2. Comparison of the recognition accuracy and the number of features used by the Bayes classifier without feature selection (Bayes All), Bayes with pairwise-greedy feature selection (Bayes FS), AdaBoost, linear SVM (L-SVM), non-linear SVM (NL-SVM), and FSLP.

	Bayes All	Bayes FS	AdaBoost	L-SVM	NL-SVM	FSLP
Accuracy	63.3%	71.0%	71.9%	92.4%	91.9%	91.0%
# Features	612	60	80	612	612	17.1

References

[1] M. Bartlett, P. Viola, T. Sejnowski, L. Larsen, J. Hager, and P. Ekman, Classifying facial action, in *Advances in Neural Information Processing Systems 8*, D. Touretzky etal eds., MIT Press, Cambridge, Mass., 823-829, 1996.

[2] K. P. Bennett and O. L. Mangasarian, "Robust linear programming discrimination of two linearly inseparable sets," *Optimization Methods and Software*, vol. 1, 23-34, 1992.

[3] P. S. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines," *Proc. 5th Int. Conf. Machine Learning*, 82-90, 1998.

[4] J. Brank, M. Grobelnik, N. M. Frayling, and D. Mladenic, Feature selection using linear support vector machines, Microsoft Technical Report MSR-TR-2002-63, June 2002.

[5] J. Daugman, Uncertainty relation for resolution in space, spatial frequency and orientation optimized by tow-dimensional visual cortical filters, *J. Optical Society of America A*, 1160-1169, 1985.

[6] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice Hall, Englewood Cliffs, N.J., 1982.

[7] Y. Freund and R. E. Schapire, A decision-theoretic generalization of online learning and an application to boosting. *J. Comp. and Sys. Sci.* **55**(1), 119-139, 1997.

[8] K. Fukunage, *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, New York, 1991.

[9] A. Jain and D. Zongker, Feature selection: Evaluation, application, and small sample performance, *IEEE Trans. Pattern Analysis and Machine Intelligence* **19**(2), 153-158, 1997.

[10] C. Liu and H. Wechsler, Probablistic reasoning models for face recognition, *Proc. Computer Vision and Pattern Recognition Conf.*, 827-832, 1998.

[11] M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, Coding facial expressions with gabor wavelets. *Proc. 3rd Int. Conf. Automatic Face and Gesture Recognition*, 200-205, 1998.

[12] M. J. Lyons, J. Budynek, and S. Akamatsu, Automatic Classification of single facial images, *IEEE Trans. Pattern Analysis and Machine Intelligence* **21**(12), 1357-1362, 1999.

[13] O. L. Mangasarian, Linear and nonlinear separation of patterns by linear programming, *Operations Research* **13**, 444-452, 1965.

[14] P. Mitra, C. A. Murthy, and S. K. Pal, Unsupervised feature selection using feature similarity", *IEEE Trans. Pattern Analysis and Machine Intelligence* **24**(3), 301-312, 2002.

[15] E. Osuna, R. Freund and F. Girosi, Training support vector machines: an application to face detection. *Proc. Computer Vision and Pattern Recognition Conf.*, 130-136, 1997.

[16] M. Pantie and L. J. M. Rothkrantz, Automatic analysis of facial expressions: The state of the art, *IEEE Trans. Pattern Analysis and Machine Intelligence* **22**(12), 1424 -1445, 2000.

[17] M. Pontil and A. Verri, Support vector machines for 3-D object recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence* **20**(6), 637-646, 1998.

[18] P. Pudil, J. Novovicova and J. Kittler, Floating search methods in feature selection, *Pattern Recognition Letters* **15**, 1119-1125, 1994.

[19] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, New Jersey, 1970.

[20] K. Tieu and P. Viola, Boosting image retrieval, *Proc. Computer Vision and Pattern Recognition Conf.*, vol. 1, 228-235, 2000.

[21] A. Vailaya, Semantic classification in image database, Ph.D. thesis, Michigan State University, 2000.

[22] V. N. Vapnik, *Statistical Learning Theory*, John Wiley, New York, 1998.

[23] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, Feature selection for SVMs, *Advances in Neural Information Processing Systems*, vol. 13, 668-674, 2000.

[24] Z. Zhang, Feature-based facial expression recognition: Sensitivity analysis and experiments with a multi-layer perceptron, *Int. J. Pattern Recognition and Artificial Intelligence*, **13**(6), 893-911, 1999.

[25] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu, Comparison between geometry-based and Gabor-wavelets-based facial expression recognition using multi-layer perceptron, *Proc. 3rd Int. Conf. Automatic Face and Gesture Recognition*, 454-459, 1998.