Feature Extraction for Regression Problems and an Example Application for Pose Estimation of a Face

Nojun Kwak[1], Sang-Il Choi[2], and Chong-Ho Choi[2]

[1] Division of Electrical & Computer Engineering, Ajou University, San5, Woncheon-dong, Yeongtong-gu, Suwon, Gyeonggi-Do, 443-749 KOREA,
nojunk@ieee.org,
WWW home page: http://ajou.ac.kr/∼nojunk

[2] School of Electrical Engineering and Computer Science, Seoul National University, #047, San 56-1, Sillim-dong, Gwanak-gu, Seoul 151-744, Korea
{kara|chchoi}@csl.snu.ac.kr

Abstract. In this paper, we propose a new feature extraction method for regression problems. It is a modified version of linear discriminant analysis (LDA) which is a very successful feature extraction method for classification problems. In the proposed method, the between class and the within class scatter matrices in LDA are modified so that they fit in regression problems. The samples with small differences in the target values are used to constitute the within class scatter matrix while the ones with large differences in the target values are used for the between class scatter matrix. We have applied the proposed method in estimating the head pose and compared the performance with the conventional feature extraction methods.

Key words: Regression, Feature extraction, Dimensionality reduction, LDA.

1 Introduction

Regression, which is a process of estimating a real-value function based on a finite set of noisy samples, is one of the classical problems in statistics, machine learning and pattern recognition societies. When dealing with classification problems, regression problems can be classified as supervised learning, where a set data consisting of pairs of input objects and desired outputs are given. The input objects and the desired outputs are usually called the *input variables* and the *target variables*, respectively.

It is well known that reducing the number of input variables through dimensionality reduction techniques such as feature selection or feature extraction is desirable. Reducing the dimensionality of the feature space may improve the learning process by considering only the most important data representation, possibly with elements retaining the maximum information of the original data and better generalization capabilities [1]. Dimensionality reduction is quite desirable not only in the aspect of the number of required data, but also in terms of data storage and computational complexity.

In this paper, we focus on the linear feature extraction methods for regression problems to reduce the dimensionality of input space.

Many studies have been performed to solve the feature extraction problems among which the principal component analysis (PCA) [2] and the independent component analysis (ICA) [3] have been widely used. Although PCA is one of the most popular and widely used methods, which is very useful in reducing the dimension of a feature space to a manageable size, it can still be improved for supervised learning problems since it is an unsupervised learning method that does not make use of the target information. Likewise, ICA, which is another unsupervised learning method, leaves much room for improvement to be used for supervised learning problems. Unlike PCA and ICA, linear discriminant analysis (LDA) [4] was originally developed for supervised learning, especially to find the optimal linear discriminating functions for classification problems.

Although many feature extraction methods have been developed for classification problems, relatively little attention has been given to feature extraction for regression problems in the machine learning society.

On the other hand, in statistics, several algorithms have been developed for dimensionality reduction in regression problems, among which the classical multivariate linear regression (MLR) [5] can be a starting point. Although MLR is optimal in the sense of least squared error, it has the limitation that it can produce only one feature. To overcome this limitation, a local linear dimensionality reduction method based on the nearest neighbor scheme has been proposed [6]. Sliced inverse regression (SIR) [7] and principal hessian directions (PHD) [8] are also very popular dimensionality reduction techniques for regression problems in statistics.

In this paper, we propose a new feature extraction method for regression problems. It is a generalization of LDA to regression problems which tries to maximize the ratio of distances of samples with large differences in target value and those with small differences in target value. The experimental results show that the proposed method performs well for many regression problems. In addition, because it only needs to solve the eigenvalue decomposition problem, it is relatively faster than iterative methods such as ICA-FX [9].

The paper is organized as follows. In Section II, we briefly overview the conventional feature extraction methods for regression problems. A new feature extraction method is presented in Section III and the experimental results are shown in Section IV. Finally, the conclusions and future works follow in Section V.

2 Conventional Methods: Linear Feature Extraction for Regression

Consider a set of predictor/response [3] pairs $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$ where $\boldsymbol{x}_i \in \Re^{d\times 1}$, $\boldsymbol{y}_i \in \Re^{t\times 1}$ and n denotes the number of given predictor/response pairs. Here, d is the number of input variables and t is the number of target variables which will be equal to 1 in most problems. [4].

[3] Note that instead of the terms *predictor* and *response*, *input* and *target* can be used without notification.

[4] From now on, we will assume $t = 1$ and instead of the vector form $\boldsymbol{y}$, the scalar form y will be used without notification.

In this regression setting, we want to find a set of linear transformations of $\boldsymbol{x}$ that can constitute sufficient statistics for target vector $\boldsymbol{y}$. This transformation can be denoted as $f_i = \boldsymbol{w}_i^T \boldsymbol{x}$, where f_i is the i-th new feature and $\boldsymbol{w}_i \in \Re^{d\times 1}$ is the corresponding coefficient or weight vector.

In this section, we introduce several conventional methods for this purpose.

2.1 Sliced Inverse Regression (SIR)

The following is the standard SIR algorithm. For simplicity, let us assume that $t = 1$ and the covariance matrix S_x of input variables $\boldsymbol{x}$ is $d \times d$ identity matrix.

Step 1. Sort the data y_i in increasing order.
Step 2. Divide the ordered data set into L slices to make the slice size as equally as possible. Let n_l be the number of examples in slice l.
Step 3. Within each slice, compute the sample mean of $\boldsymbol{x}$, $\bar{\boldsymbol{x}}_l = \frac{1}{n_l} \sum_{i\in \text{slice } l} \boldsymbol{x}_i$.
Step 4. Compute the covariance matrix for the slice means of $\boldsymbol{x}$, weighted by the slice sizes.

$$S_\eta = \frac{1}{n} \sum_{l=1}^{L} n_l (\bar{\boldsymbol{x}}_l - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_l - \bar{\boldsymbol{x}})^T \tag{1}$$

Here, $\bar{\boldsymbol{x}}$ denotes the sample mean of $\boldsymbol{x}$ such that $\bar{\boldsymbol{x}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i$.
Step 5. Find the k-th SIR direction $\boldsymbol{w}_k$ by conducting the eigenvalue decomposition of S_η.

$$S_\eta \boldsymbol{w}_k = \lambda_k \boldsymbol{w}_k, \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \tag{2}$$

Note the similarity of SIR to PCA. SIR takes L points each of which is the sample mean of n_l points in each slice l and then performs the PCA to these L points. However, the difference is that in generating the L points, $\boldsymbol{x}$s that are associated with similar y values are averaged out to capture the relationship between the input $\boldsymbol{x}$ and the target y.

2.2 Principal Hessian Directions (PHD)

As in SIR, let us assume that $t = 1$ and let $f(\boldsymbol{x})$ be the regression function $E(Y|\boldsymbol{x})$. Here, $E(\cdot)$ denotes expectation. Consider the Hessian matrix $H(\boldsymbol{x})$ of $f(\boldsymbol{x})$ whose (i, j) component is as follows:

$$H_{ij}(x) = \frac{\partial^2}{\partial x_i \partial x_j} f(x), \tag{3}$$

where x_k is the k-th component of the vector $\boldsymbol{x}$.

Hessian matrices are important in studying multivariate nonlinear functions and PHD focuses on the utilization of the properties of Hessian matrices for dimensionality reduction. In the PHD algorithm, the principal Hessian directions $\boldsymbol{w}_k$s ($k = 1, \cdots, d$) are obtained by solving the following eigenvalue decomposition problem:

$$S_{yxx} \boldsymbol{w}_k = \lambda_k \boldsymbol{w}_k, \quad |\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_d| \tag{4}$$

where S_{yxx} can be estimated by

$$S_{yxx} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T. \tag{5}$$

Because the PHD is based on the Hessian matrix, it performs poorly on the problems where targets are linearly related to the input variables.

2.3 Linear Discriminant Analysis (LDA)

Unlike the methods previously described in this section, LDA focuses on the classification problem where instead of a continuous target variable y, a discrete class identifier $c \in \{1, \cdots, N_c\}$ is used. Here, N_c is the number of classes.

In LDA, we try to optimize the following Fisher's criterion such that the ratio of the between-covariance matrix $S_b = \frac{1}{n}\sum_{c=1}^{N_c} n_c(\bar{\boldsymbol{x}}_c - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_c - \bar{\boldsymbol{x}})^T$ and the within-covariance matrix $S_w = \frac{1}{n}\sum_{c=1}^{N_c}\sum_{i\in\{\text{class}=c\}}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_c)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_c)^T$ is maximized.

$$W = \arg\max_{W} \frac{|W^T S_b W|}{|W^T S_w W|} \tag{6}$$

Here, $\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i$ is the total mean of the samples, n_c is the number of samples belonging to the class c and $\bar{\boldsymbol{x}}_c = \frac{1}{n_c}\sum_{i\in\{\text{class}=c\}}\boldsymbol{x}_i$ is the mean of the samples belonging to the class c.

The optimization problem in (6) is equivalent to the following generalized eigenvalue problem,

$$S_b\boldsymbol{w}_k = \lambda_k S_w \boldsymbol{w}_k \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d, \tag{7}$$

where $\boldsymbol{w}_1$ is the most discriminant component, $\boldsymbol{w}_2$ is the second, and so on.

3 The Proposed Method: LDA for regression

In the classification problems, LDA has been a very successful method for dimensionality reduction and many variants have been also developed. As described in the previous section, the gist of LDA lies in maximizing Fisher's criterion which tries to maximize the between-class scatter while minimizing the within-class scatter.

In this section, we extend this idea to the regression problems and a new feature extraction algorithm for regression is proposed. From now on, the new method will be referred to as *LDA-r*.

Unlike the classification problems, it is difficult to define the between-class scatter and within-class scatter matrices in regression problems because the target variable is continuous. The simple idea that the samples with small differences in the target values are considered as belonging to the same class, while the ones with large differences are considered as belonging to different classes, is used to define the between-class and within-class scatter matrices. The followings are the modified within-class

and between-class scatter matrices for LDA-r:

$$S_{wr} = \frac{1}{n_w} \sum_{(i,j)\in A_w} f(y_i - y_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \tag{8}$$

$$S_{br} = \frac{1}{n_b} \sum_{(i,j)\in A_b} f(y_i - y_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T. \tag{9}$$

Here, $A_w = \{(i,j) : |y_i - y_j| < \tau, \quad i,j \in \{1,\cdots,n\}, \quad i \neq j\}$, $A_b = \{(i,j) : |y_i - y_j| \geq \tau, \quad i,j \in \{1,\cdots,n\}, \quad i \neq j\}$ and $n_w = |A_w|$ and $n_b = |A_b|$. The function $f(\cdot)$ is a weight function positive values. Note that $n_w + n_b = \frac{n(n-1)}{2}$.

Using this modified scatter matrices, the Fisher's criterion can be rewritten for regression problems as

$$W = \arg\max_W \frac{|W^T S_{br} W|}{|W^T S_{wr} W|}. \tag{10}$$

As stated earlier, maximizing the above Fisher's criterion is equivalent to solving the generalized eigenvalue problem:

$$S_{br}\boldsymbol{w}_k = \lambda_k S_{wr}\boldsymbol{w}_k \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \tag{11}$$

which is again equivalent to the following eigenvalue decomposition problem:

$$S_{wr}^{-1} S_{br}\boldsymbol{w}_k = \lambda_k \boldsymbol{w}_k \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \tag{12}$$

where $\boldsymbol{w}_1$ is the most important component, $\boldsymbol{w}_2$ is the second, and so on.

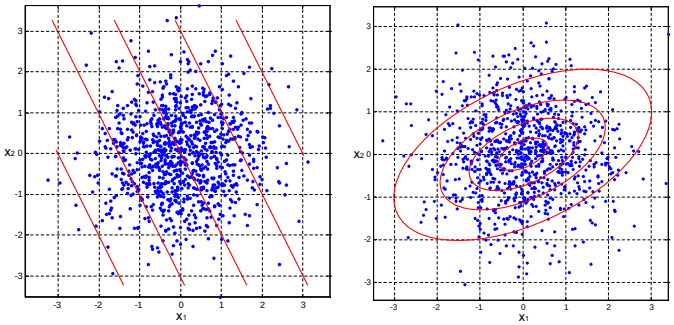
In modifying LDA for regression problems, we could have segmented the given dataset into several virtual classes based on the target values with fixed boundaries and applied the conventional LDA for classification problems. Although this method is simple, the results can be highly dependent on how to segment boundaries and the number of virtual classes. In addition, this approach may not take into account the different levels of similarity among different classes. Therefore, in LDA-r, soft boundaries which are different from one sample to another are used.

Note that the threshold parameter τ plays an important role in setting the boundary. If τ is small, n_w becomes small while n_b becomes large and vise versa. The threshold τ can be represented as a multiple of the standard deviation σ_y of target variable y such that $\tau = \alpha\sigma_y$. Typical range for α is 0.1 to 1.0.

Although the weight function $f(\cdot)$ can be set as a constant, e.g., $f(x) = 1$, it is probably better to make $f(x)$ take different values for different inputs. Because $|y_i - y_j| = \tau$ sets a boundary whether the pair (i,j) should belong to A_w or A_b, the effect of (i,j)-pair which is near this boundary can be reduced by setting $f(x) \simeq 0$ for $|x| \simeq \tau$. Typical examples of $f(\cdot)$ fulfilling this requirement are $f(x) = ||x| - \tau|$ and $f(x) = \sqrt{||x| - \tau|}$.

Note that LDA-r is not invariant to transformation of input features and susceptible to scaling of input features as in LDA. Therefore, it is desirable to preprocess the given dataset by applying PCA which is often called the sphering process [2].

The computational complexity of LDA-r can be decomposed into two parts. The first part is related to obtaining the covariance matrices shown in (9) and it is proportional to the square of the number of examples, i.e., $O(n^2)$. The second part is related



(a) Linear Target (b) Quadratic Target

Fig. 1. One thousand random points drawn from $N(0, I_2)$. The slanted lines and ellipses in red are the contour map which connects the points that have the same y value.

to solving the eigenvalue decomposition problem in (11) and it is typically proportional to the cubic of the input dimension, i.e., $O(d^3)$.

Comparing this to the complexity of LDA, because the second part is common in LDA and LDA-r, we see that LDA-r is somewhat more computationally complex than LDA which requires $O(n)$ operations in obtaining the scatter matrices. However, for a large n, a subset of samples can be selected in computing the scatter matrices to reduce the computational complexity.

4 Experimental Results

4.1 Linear and Quadratic Targets

Consider two independent input features x_1 and x_2 which have normal distribution with zero mean and variance of 1. In addition, suppose the target output variable y has the following relationships with the input $\boldsymbol{x}$:

$$\text{Linear:} \quad y = 2x_1 + x_2 \tag{13}$$

$$\text{Quadratic:} \quad y = 4(x_1 - 2x_2)^2 + (2x_1 + x_2)^2. \tag{14}$$

In Fig. 1(a) and (b), we have plotted 1,000 samples each. In each figure, a contour map was drawn in red which connects the points that have the same y value (slanted lines for the linear case, and ellipsoids for the quadratic case). For these empirical data, we have applied SIR, PHD, and LDA-r.

Linear target: For the linear case, the optimal feature is $f = 2x_1 + x_2$ which corresponds to the optimal weight vector $\boldsymbol{w}^* = [2, 1]^T$.

Considering that the area between the neighboring slanted lines can be considered as a slice in SIR, there will be significant differences in the mean values $\bar{\boldsymbol{x}}_l (l = 1, \cdots, L)$ of each slices and we expect the SIR will work well for this problem. As expected, SIR

produced $\boldsymbol{w} = [0.89, 0.45]^T$ which is very close to the optimal value $\boldsymbol{w}^*$. The number of slices was set to $L = 10$ in this case.

Regarding PHD, because y is linear with respect to $\boldsymbol{x}$, all the elements in the Hessian matrix of this problem are zeros and we can expect PHD can not solve this problem. As a matter of fact, for the empirical data shown in Fig. 1(a), PHD produced $\boldsymbol{w} = [0.88, -0.51]$ which is far from $\boldsymbol{w}^*$.

The reason PHD fails to this problem lies in the form of the weight function. In PHD, the weight function is just the deviation from the target mean $\bar{y}$. Therefore, the points in the lower left part in Fig. 1(a) have negative weights ($y_i - \bar{y} < 0$) and the other points which are located in the upper right part have positive weights ($y_j - \bar{y} > 0$). As a result, contributions of any two points which are symmetric with respect to the center cancel out each other in the formation of S_{yxx} and the eigenvalues of S_{yxx} become very small resulting in poor performance of PHD.

For this example, LDA-r is also applied with weight function $f(x) = \sqrt{||x| - \tau|}$ and $\alpha = 0.3$. LDA-r resulted in $\boldsymbol{w} = [0.89, 0.45]^T$ which is very close to the optimal weight. Note that in LDA-r, the scatter matrices are all positive semi-definite.

Quadratic target: As shown in Fig. 1(b), for a fixed y, (x_1, x_2) constitutes an ellipsoid whose major axis is in the direction of $(2, 1)$ and the minor axis is in $(-1, 2)$.

If we are to extract only one feature among the set of linear combinations of input variables x_1 and x_2, the major axis is the best projection which corresponds to a feature $f = x_1 - 2x_2$, i.e., $\boldsymbol{w}^* = [1, -2]^T$.

As expected, SIR does not work well for this example because all the mean values of the different slices are near zero and a random direction which is highly dependent on a specific data will be chosen. For the empirical data shown in Fig. 1(b), SIR with $L = 10$ extracted the first weight vector $\boldsymbol{w} = [-0.84, 0.52]^T$ which is far from the optimal value $\boldsymbol{w}^* = [1, -2]^T$.

Unlike SIR, PHD works well for this problem because y is quadratic with respect to $\boldsymbol{x}$ and the principal Hessian directions are easily calculated. Calculating the Hessian matrix, it becomes $H = \begin{bmatrix} 16 & -12 \\ -12 & 34 \end{bmatrix}$ and the principle Hessian direction is $[1, -2]^T$ as expected. For the empirical data shown in Fig. 1(b), the PHD algorithm resulted in $\boldsymbol{w} = [0.44, -0.90]^T$ which is very close to the optimal value.

For this example, LDA-r is also applied with weight function $f(x) = \sqrt{||x| - \tau|}$ with $\alpha = 0.3$. LDA-r resulted in $\boldsymbol{w} = [0.44, -0.90]^T$ which is the optimal vector.

4.2 Pose Estimation

In this part, the proposed algorithm is applied to a pose estimation problem, by taking it as a regression problem, and the proposed algorithm are compared to some of other conventional methods.

In face recognition systems, pose variation in a face image significantly degrades the accuracy of face recognition. Therefore, it is important to estimate the pose of a face image and classify the estimated pose into the correct pose class before the recognition

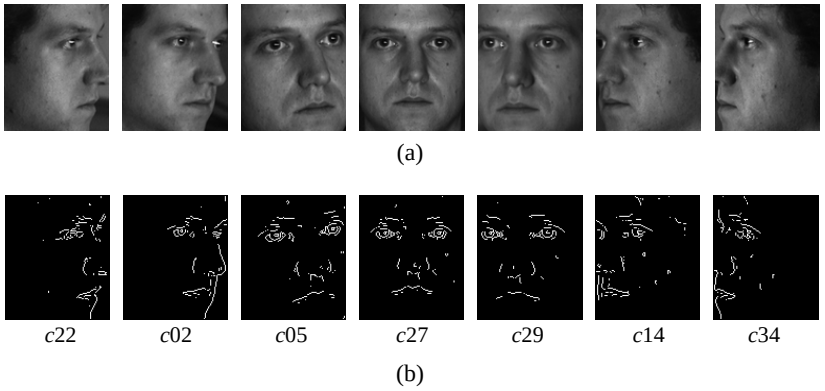


Fig. 2. Edge images for different poses: (a) images under various poses; (b) corresponding edge images.

procedure. Given face images with pose variation, an image can be assigned to a pose class by a classification method using a feature extraction method.

However, unlike general classification problems, since pose classes can be placed sequentially from left profiles to right profiles in the pose space, there is an order relationship between classes, which can be represented in distance, and the distance between classes can be used as a measure of class similarity. For example, consider a pose estimation problem which consists of three pose classes 'front (0°)', 'half profile (45°)' and 'profile (90°)'. In this problem, 'profile' images are more closer to 'half profile' images than 'front' images. If a classifier misclassifies a 'profile' image, it would be better to classify it into a 'half profile' than a 'front' image. Thus, we can make use of the order relationship between classes for feature extraction. In this sense, these types of classification problems are similar to regression problems. If each of the pose classes is assigned a numerical target value, the pose estimation problem may be regarded as a regression problem and the feature extraction methods can be used to extract useful features in discriminating the pose of a face image.

We evaluate the performance of pose estimation on the CMU-PIE database [10]. The CMU-PIE database contains more than 40,000 facial images of 68 individuals, 21 illumination conditions, 12 poses and four different expressions. Among them, we selected the images of 65 individuals with seven pose indices ($c22$, $c02$, $c05$, $c27$, $c29$, $c14$, $c34$). Each face was cropped to include only the face and rotated on the basis of the distance among the manually selected points on an image, and then rescaled to a size of 120×100 (see Fig. 2(a)). Three images under different illumination variation for each of the 65 individuals in each pose class were used as a training set while the other 8190 (65x18x7) images were used for testing. We first divided the pose space into seven pose classes from left profile to right profile and built a feature space for each pose class using feature extraction methods explained in the previous section. In order to estimate a pose of a face image, each of the seven pose classes was assigned a numerical target value from 1 (left profile) to 7 (right profile).

Table 1. Error rate in pose classification on face images(%)

Method	$c22$	$c02$	$c05$	$c27$	$c29$	$c14$	$c34$	Overall
PHD (1200)	28.80	44.62	28.89	1.37	1.88	5.98	3.76	12.36
SIR (1200)	29.74	44.87	27.95	1.71	2.22	7.61	3.25	16.76
LDA (6)	9.66	0	0	4.53	9.49	8.38	12.48	6.34
LDA-r(200)	7.61	0.09	0	2.56	2.82	4.87	7.18	3.59

Table 2. Error rate in pose classification on edge images(%)

Method	$c22$	$c02$	$c05$	$c27$	$c29$	$c14$	$c34$	Overall
PHD (1200)	9.91	5.04	1.97	2.65	2.65	5.73	4.87	4.69
SIR (1200)	9.32	4.87	1.97	2.65	2.65	5.38	4.44	4.47
LDA (6)	1.03	1.03	0.17	0.26	0.26	1.97	2.56	1.04
LDA-r(200)	0.94	0.94	0	0.35	0.09	1.03	3.23	0.80

In the experiment below, each of the pixels was used as an input feature constituting a 12,000 dimensional input space and the methods presented in the previous section were used to extract features for estimating the pose. As can be seen, this problem is a typical example of the SSS problem whose input dimension d (12,000) is much larger than the number of training examples n (1,365). To resolve this SSS problem, in all the feature extraction methods, we have preprocessed the dataset with PCA and reduced the dimension of input space into $n-1$. For the proposed method, the weight function $f(x) = \sqrt{||x| - \tau|}$ and α was set to 0.1. With these extracted features, the one nearest neighborhood rule was used as a classifier with the Euclidean distance ($L2$) as the distance metric.

Table 1 shows the error rates of pose classification for the test images using several methods. Numbers in the parentheses are the number of features. As can be seen in Table 1, the proposed method is better than the other methods in most cases. Overall error rates of PHD and SIR ($L = 10$) are above 12%, while LDA gives an overall error rate of 6.34%. However, since the pose estimation is a classification problem where levels of similarity among different classes can be defined, LDA-r is more suitable for this problem than LDA, and we can see that the overall error rate of LDA-r is 2.75% lower than that of LDA.

On the other hand, the images such as those in Fig. 2(a) contain necessary information for pose estimation as well as other information such as the illumination condition, appearance variation, etc. In order to remove the redundant information for pose estimation, we transform a face image to an edge image by using the Sobel mask [11]. As shown in Fig. 2(b), the edge images enhance the geometrical distribution of facial feature points. Even though the edge images may be sensitive to illumination variation, the pose estimation can be reliably performed on images under illumination variation if the training set contains edge images under various illumination conditions. Subsequently, as can be seen in Table 2, the overall error rates are lower than those in Table 1. In the case of edge images, the performance difference between each feature extraction meth-

ods became smaller compare to the raw images, but we can see that the performance of LDA-r is still better than the other methods.

5 Conclusions

In this paper, we have proposed a new method for linear feature extraction for regression problems. It is a modified version of LDA. The distance information among samples are utilized in constructing the within class and between class scatter matrices.

The two examples in Section 4.1 show the advantage of the proposed method against the conventional methods such as SIR and PHD. It showed good performance on both examples, while SIR and PHD performed poorly in one of the examples. We also applied the proposed method to estimating the head pose of a face image and compared the performance to those of the conventional feature extraction methods.

The experimental result in pose estimation shows that the proposed method produces better features than the conventional methods such as SIR, PHD and LDA. The proposed method is easy to implement and is expected to be useful in finding good linear transformations for regression problems.

Acknowledgments. This work was partly supported by Samsung Electronics.

References

1. Cios, K.J., Pedrycz, W., Swiniarski, R.W.: Data Mining Methods for Knowledge Discovery - Chapter 9. Kluwer Academic Publishers (1998)
2. Joliffe, I.T.: Principal Component Analysis. Springer-Verlag (1986)
3. Bell, A.J., Sejnowski, T.J.: An Information-Maximization Approach to Blind Separation and Blind Deconvolution. Neural Computation. **7** (1995) 1129-1159
4. Fukunaga, K.: Introduction to Statistical Pattern Recognition. 2nd ed. Academic Press, New York (1990)
5. Weisberg, S.: Applied Linear Regression - Chapter 3. 2nd ed. John Wiley, New York (1985) 324
6. M. Loog: Supervised Dimensionality Reduction and Contextual Pattern Recognition in Medical Image Processing - Chapter 3. Ponsen & Looijen, Wageningen, The Netherlands. (2004)
7. Li, K. C.: Sliced Inverse Regression for Dimension Reduction (with discussioin): J. the American Statistical Association. **86** (1991) 316-342
8. Li, K. C.: On Principal Hessian Directions for Data Visualization and Dimension Reduction: Another Application of Stein's lemma: J. the American Statistical Assiciation. **87** (1992) 1025-1039
9. Kwak, N., Kim, C.: Dimensionality reduction based on ICA for regression problems: Proc. Int'l Conf. on Artificial Neural Networks (IJCNN) (2006) 1-10
10. Sim, T., Baker, S., Bsat,M.: The CMU Pose, Illumination, and Expression Database. IEEE Trans. Pattern Analysis and Machine Intelligence. **25** (2003) 1615-1618
11. Georghiades, A.S., Belhumeur, P.N.: Frome Few to Many: Illumination Cone Models for Face Recognition Under Variable Lighting and Pose. IEEE Trans. Pattern Analysis and Machine Intelligence. **23** (2001) 643-660