Studying the Effectiveness of Using Linear Subspace Techniques to Improve SVM Classifiers in Facial Image Classification

Nikolaos Tsapanos
Department of Informatics
Aristotle University of Thessaloniki
Informatics and Telematics
Institute, CERTH
Email: niktsap@aiia.csd.auth.gr

Nikolaos Nikolaidis
Department of Informatics
Aristotle University of Thessaloniki
Informatics and Telematics
Institute, CERTH
Email: nikolaid@aiia.csd.auth.gr

Ioannis Pitas
Department of Informatics
Aristotle University of Thessaloniki
Informatics and Telematics
Institute, CERTH
Email: pitas@aiia.csd.auth.gr

***Abstract*—In this paper we investigate the potential benefits of combining, within a classification task, a discriminant linear subspace feature extraction technique, namely Discriminant Non-negative Matrix Factorization (Discriminant NMF or DNMF), with a Support Vector Machine (SVM) classifier. The aim was to investigate whether this combination provides better classification results compared to a template matching method operating on the DNMF space or on the raw data and an SVM classifier operating on the raw data, when applied on the frontal facial pose recognition problem. The latter is a two-class problem (frontal and non-frontal facial images). DNMF is based on a supervised training procedure and works by imposing additional criteria on the NMF objective function that aim at increasing class seperability in the lower dimensionality space. Results on face images extracted from the XM2VTS dataset show that feeding the DNMF subspace data into the SVM is the approach that provides the best results.**

I. Introduction

Over the past decades, a lot of research has been carried out on human face related computer vision and machine learning tasks, such as face detection and tracking, facial features detection, face recognition and facial expression recognition. While face detection systems that work for various, non-frontal face poses have been developed [1], face recognition and facial expression recognition techniques have been designed to work on frontal or nearly frontal face images [2], [3]. Thus, an issue that arises in practical situations is that a face or facial expression classifier trained with frontal face images will not be able to meaningfully operate in non-frontal face images. Therefore, the problem of frontal face pose recognition needs to be solved, so that frontal face images can be detected and used as input in face recognition or facial expression recognition systems.

Head pose estimation techniques [4], where the orientation of the human head is estimated by determining the value of the yaw, roll and pitch angles, can be used to determine if a facial image is suitable for use by a frontal face or facial expression recognition system. In this paper, however, we view the frontal face pose recogntion problem in a simplified a 2-class framework. Rather than determining the head pose angles, we simply consider an image of a face to be either frontal or non-frontal. Four different appearance based approaches are investigated for this task. The first approach is a simple template matching approach. The second approach involves feeding the raw image data (intensity) to a Support Vector Machine (SVM). The third approach uses Discriminant Non-negative Matrix Factorization (Discriminant NMF or DNMF) to reduce the dimensionality of the data and subsequently involves template matching. The final approach uses the DNMF transformed data along with an SVM. Our goal is to verify our expectation that combining DNMF with an SVM provides the best overall performance.

The paper is organized as follows: Section II briefly describes the DNMF algorithm, Section III goes over the principles of the SVM classifier, while section IV briefly describes template matching. Experimental data are presented in section V and section VI concludes the paper.

II. Discriminant Non-negative Matrix Factorization

Discriminant Non-negative Matrix Factorization, as its name implies, is an attempt to incorporate discriminant information into the NMF algorithm. NMF decomposes a set of input vectors into a set of basis vectors and a set of coefficient vectors. When the dimensionality of the coefficients' space is smaller than the dimensionallity of the original vectors' space, NMF becomes a subspace technique that can be used as a preprocessor to data that will be used in a classifier.

Whereas NMF optimizes a data reconstruction criterion, DNMF allows for greater reconstruction errors, in order to make the classes more easily separable [5].

Suppose that the data set matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_n]$, where $\mathbf{x}_i$ is an individual data column vector is to be written in the form of:

$$\mathbf{X} = \mathbf{BH}$$

where $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_d]$ is a matrix containing the basis vectors $\mathbf{b}_i$ in column format, d is the dimension of the

projection space and $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_D]^T$ is a matrix containing the coefficient vectors, D being the dimension of the original data space.

In order to estimate the quality of this factorization, NMF uses the Kullback-Leibler divergence

$$D_{KL}(\mathbf{x}||\mathbf{y}) = \sum_i (x_i \log \frac{x_i}{y_i} + y_i - x_i)$$

to evaluate how different the reconstructed data ($\mathbf{BH}$) are from the original data ($\mathbf{X}$). Thus, the goal of the NMF algorithm is to minimize

$$D_{KL}(\mathbf{X}||\mathbf{BH}) \tag{1}$$

with respect to $\mathbf{B}$ and $\mathbf{H}$. This is done through an iterative process that estimates the basis and coefficient vectors in $\mathbf{B}$ and $\mathbf{H}$. More details can be found in [6].

In order to improve the separability of the projected data, it is reasonable to require that the centers of each class (as defined by the class mean) in the projected space are further apart, while all the data of the same class are more closely clustered together. Similar to the Linear Discriminant Analysis (LDA) [7], DNMF accomplishes this by taking into account the between class scatter matrix ($\mathbf{S}_b$) and the within class scatter matrix ($\mathbf{S}_w$).

In a data set $\mathbf{X}$ of C classes, with N_c samples $\mathbf{x}_i^{(c)}$ for each class, the between class scatter matrix is given by:

$$\mathbf{S}_b = \sum_{c=1}^{C} N_c(\hat{\mathbf{x}}^{(c)} - \hat{\mathbf{x}})(\hat{\mathbf{x}}^{(c)} - \hat{\mathbf{x}})^T$$

where $\hat{\mathbf{x}}^{(c)}$ is the mean of class c and $\hat{\mathbf{x}}$ is the mean of the entire data set. The trace of this matrix provides an estimate on how far apart the classes are. The within class scatter matrix for class c is given by:

$$\mathbf{S}_w^{(c)} = \sum_{i=1}^{N_c} (\mathbf{x}_i^{(c)} - \hat{\mathbf{x}}^{(c)})(\mathbf{x}_i^{(c)} - \hat{\mathbf{x}}^{(c)})^T$$

while the overall scatter matrix is $\mathbf{S}_w = \sum_{c=1}^{C} \mathbf{S}_w^{(c)}$. The trace of this matrix provides an estimate on how far apart the data within each class are. DNMF enriches the objective function of the NMF method by including the traces of these matrices in (1), forming the new objective function:

$$D_{KL}(\mathbf{X}||\mathbf{BH}) - \kappa_1 * tr(\mathbf{S}_b) + \kappa_2 * tr(\mathbf{S}_w) \tag{2}$$

The estimation of $\mathbf{B}$ and $\mathbf{H}$ is again performed by an iterative optimization process using a random initialization. Once the basis vectors in $\mathbf{B}$ have been estimated using a set of training data, a test vector $\mathbf{t}$ can be projected to the DNMF space by multiplying with $\mathbf{B}$ from the right:

$$\mathbf{t}_d = \mathbf{tB}$$

III. Support Vector Machines

Support Vector Machines [8] originated from the attempt to optimally separate two classes, C_1 and C_2, that are linearly separable with a single hyperplane. A hyperplane is defined by the equation $< \mathbf{w}, \mathbf{x} > +b = 0$, where $\mathbf{w}$ is the hyperplane's normal vector and $<,>$ denotes the dot product between two vectors.

When two classes C_1 and C_2 are linearly separable, then we can find a normal $\mathbf{w}$ and a constant b, so that $< \mathbf{w}, \mathbf{x} > +b \geq 1$, if $\mathbf{x} \in C_1$ and $< \mathbf{w}, \mathbf{x} > +b \leq -1$, if $\mathbf{x} \in C_2$. For the points of C_1 that satisfy the equality $< \mathbf{w}, \mathbf{x} > +b = 1$, it is easy to see that their distance from the separating hyperplane is $\frac{1}{||\mathbf{w}||}$ (likewise for the points of C_2). The sum of the minimum distances ($\frac{2}{||\mathbf{w}||}$) from the points of each category to the separating hyperplane is called a margin. A reasonable assumption for a hyperplane that optimally separates the two classes is that maximizes the margin.

In order to find the optimal separating hyperplane, we need to minimize $||\mathbf{w}||$, or equivalently $\frac{1}{2}||\mathbf{w}||^2$. Since we also like the separation (classification) to be correct, we need to add the proper constraints. We do this by requiring that $c_i(< \mathbf{w}, \mathbf{x}_i > +b) - 1 \geq 0$, where c_i is 1, if $\mathbf{x}_i \in C_1$ and -1, if $\mathbf{x}_i \in C_2$. We add these constraints to the optimization problem using non-negative Lagrange multipliers α_i. Thus the Langangian becomes:

$$L_P = \frac{1}{2}||\mathbf{w}||^2 - \sum_i \alpha_i c_i(< \mathbf{w}, \mathbf{x}_i > +b) + \sum_i \alpha_i \tag{3}$$

Requiring that the derivative of L_P with respect to $\mathbf{w}$ and b vanishes, we obtain that:

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \tag{4}$$

$$\sum_i \alpha_i y_i = 0$$

Replacing (4) in (3) we formulate the Wolfe Dual of L_P:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_i \alpha_i \alpha_j y_i y_j < \mathbf{x}_i, \mathbf{x}_j > \tag{5}$$

One of the Wolfe Dual's (L_D) properties is that its maximum occurs for the same values of $\mathbf{w}$, b and α_i as the minimum of L_P.

The SVM can be trained by maximizing (5) and using the values of the Lagrange multipliers to determine w according to (4). After the maximization, only a few α_i will be non-zero. The data points $\mathbf{x}_i$ whose Lagrange multipliers are non-zero are called the support vectors.

In the case where the data are not linearly separable, the SVM's parameters after training are still determined by (4), with the difference that some classification errors are allowed and fewer points become support vectors. However, this is still not enough to produce satisfactory classification results on classes that are very hard to separate. In this case, a non-linear mapping Φ is used to project the data to a higher dimensionality space, where the classes may indeed be linearly separable, or, in any case, more easily separable.

The problem that arises from this approach is that Φ is not always practical or even possible to compute. SVMs can overcome this problem by using the appropriate kernel function for the different mappings Φ. A function k is a kernel function for the mapping Φ iff:

$$k(\mathbf{x}_i, \mathbf{x}_j) = < \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j) >$$

where $\mathbf{\Phi}(\mathbf{x}_i)$ and $\mathbf{\Phi}(\mathbf{x}_i)$ are the projections of the $\mathbf{x}_i$ and $\mathbf{x}_j$ accoring to the mapping Φ.

When a vector $\mathbf{t}$ has to be classified by a trained SVM, the output of the SVM is:

$$t \in \begin{cases} C_1, \text{ if} \sum_s \alpha_s y_s k(\mathbf{t}, \mathbf{x}_s) \geq 0 \\ C_2, \text{ if} \sum_s \alpha_s y_s k(\mathbf{t}, \mathbf{x}_s) < 0 \end{cases}$$

where $\mathbf{x}_s$ are the selected support vectors.

IV. Template matching

A simple baseline template matching technique has also been used. A class template is constructed for the frontal facial class and test images are classified according to their euclidean distance from that template. The template is constructed by calculating the arithmetic mean of all the vectors of this class. There is a threshold for that distance, below which the test image is accepted as a member of the class and above which it is rejected, i.e. it is assigned to the non-frontal facial class.

V. Experiments

Our objective in this paper was two-fold. Our first objective was to determine how much of an improvement does the DNMF provide over using the raw (image intensity) data along with an SVM classifier. Our other objective was to test whether using an SVM benefits more from DNMF than a classifier that depends on data being clustered around the class center, namely template matching.

Our experiments were conducted on data obtained from the XM2VTS face database [9]. Face tracking was applied on the head rotation shot videos, that depict people that start from a frontal pose, turn their heads to their right profile, back to frontal pose then to the left profile. The images were then rescaled to a size of 30×40. There are 6862 facial images captured this way, with 2486 of them being frontal and 4376 non-frontal.

We first reshaped every 30×40 image into a vector with 1200 elements. We then randomly split the data vectors in half for both classes to form the raw data training and test sets. Thus, both sets consisted of 1243 frontal face images and 2188 non-frontal images with no overlaps between the sets. We then used the DNMF algorithm to reduce the dimensionality to 100 for both the training and test sets.

Our baseline template matching test was the one described in section IV. The baseline SVM test was feeding the raw data into an SVM that used a second degree polynomial kernel. We then repeated the above tests using the DNMF data sets.

The overall performance of the classifiers was judged by their Equal Error Rate (EER), the point were the missclassification percentage of one class equals the missclassification percentage of the other. As we can see from the EER results in Table I, the baseline SVM classifier is significantly better than the template matching classifier. Furthermore, DNMF, as expected, improves both of these classifiers' performance, however the margin of improvement is larger in the SVM case. The combination of DNMF and SVM provides the overall best result.

TABLE I
EER FOR ALL THE COMBINATIONS OF INPUT DATA AND CLASSIFIERS.

	Template matching	SVM
Raw data	0.1834	0.1191
DNMF	0.1689	0.0491

Since in our case, but also in other cases, the subject of the frontal face pose recognition task is to determine whether an image is a suitable input for a face recognition or facial expression recognition system that has been trained using frontal face images, it is reasonable that we should be able to favor, if needed, the non-frontal class, i.e. limit the classification errors for data that belong to the non-frontal class. Especially in the case of face recognition on a video sequence, misclassifying some of the frontal face poses as non-frontal still allows a system to determine through voting the identity of the face from the rest of the frames where a frontal face pose was correctly recognized. However, misclassifying non-frontal poses as frontal ones and then feeding them to the face recognition algorithm is sure to introduce erroneous face recognition results that can negatively affect the voting process.

In the template matching classifier, we can adjust the balance of the two classes by setting the threshold that must met in order for a facial image to be classified as frontal higher, in order to accept more images as frontal, or lower, in order to prevent non-frontal images accepted as frontal. In the SVM case, the outputs of the classifier are in the range of $[0, 1]$, if the image is classified as frontal and $[-1, 0)$ otherwise. By introducing an additional bias applied to this result, we can, again, favor either class during the classification. Figure 1 presents the varying error rates for both classes and all the classifiers tested.

VI. Conclusion

In this paper we investigated the effect that DNMF, a subspace technique, has on the performance of an SVM classifier and compared it against the effect of DNMF on a simple template matching classifier. Our primary goal was to verify that using DNMF to preprocess the data before training an SVM with them is indeed an improvement over an SVM trained with the raw data.

Our experiments verifed this improvement, while also showing that this improvement is more significant than the improvement DNMF provides to template matching, a classification method that DNMF is more obviously suited for. The SVM fed with the DNMF data proved to be the best combination, indicating that the high performance of the DNMF+SVM could not only be attributed to the fact that the SVM is an extremely

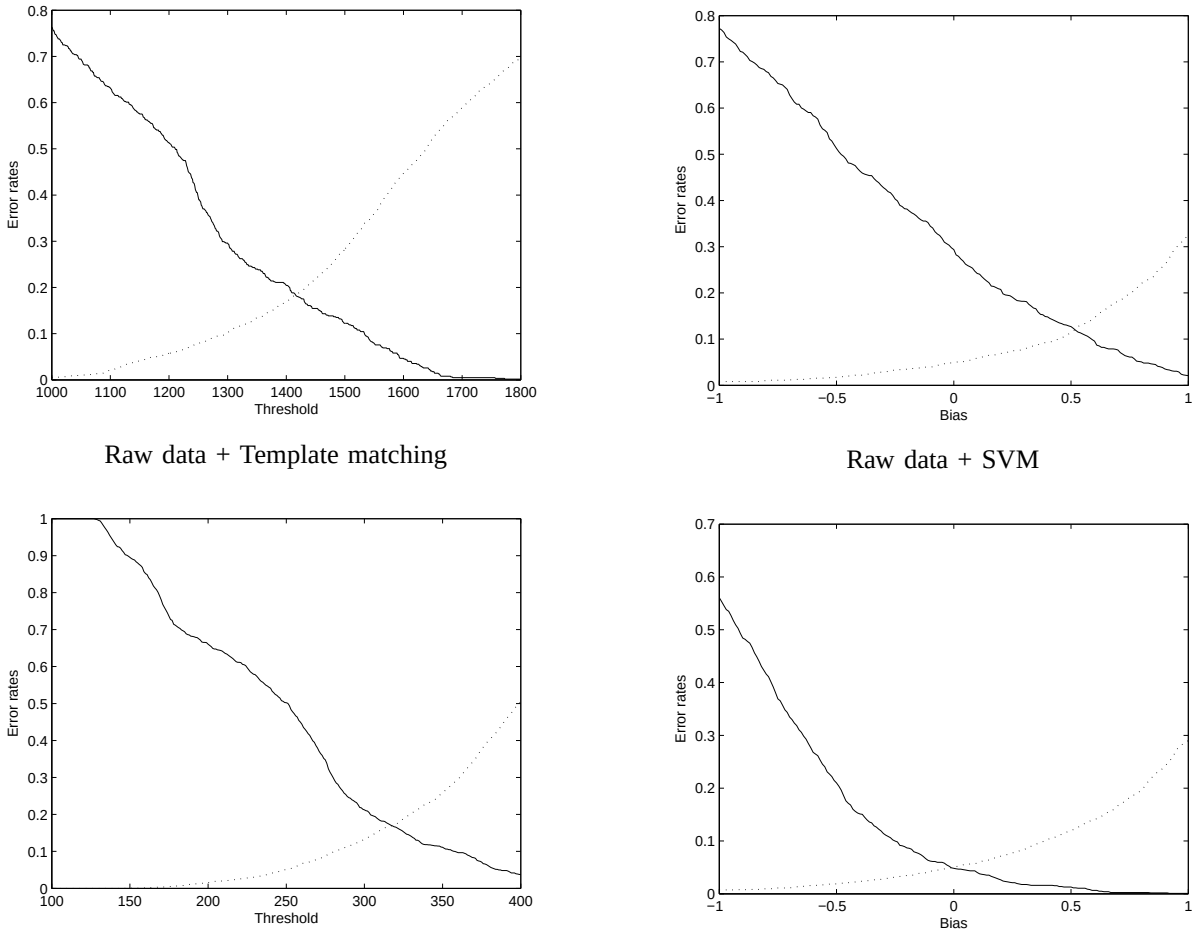

Fig. 1. Error rates for all the combinations of input data and classifiers. The frontal class error is present with a solid line, while the non-frontal class error is presented with a dotted line.

good cliassifer, as the SVM by itself had significantly lower performance. The experimental results are also an indication that other face related tasks can benefit from the combination of DNMF with SVMs, though there is no concrete evidence supporting this yet.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Communitys Seventh Framework Programme (FP7/2007-2013) under Grant agreement No. 211471 (i3DPost).

REFERENCES

[1] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, August 2002. [Online]. Available: http://dx.doi.org/10.1109/34.982883

[2] W. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips, "Face recognition: A literature survey," *ACM Computing Surveys*, pp. 399–458, 2003. [Online]. Available: http://www.face-rec.org/interesting-papers/

[3] B. Fasel, J. Luettin, B. Fasel, and J. Luettin, "Automatic facial expression analysis: A survey," *Pattern Recognition*, vol. 36, pp. 259–275, 1999.

[4] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 4, pp. 607–626, April 2008. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2008.106

[5] S. Zafeiriou, A. Tefas, and I. Pitas, "Exploiting discriminant information in elastic graph matching," pp. 768–771, 2005.

[6] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization." *Nature*, vol. 401, no. 6755, pp. 788–791, October 1999. [Online]. Available: http://dx.doi.org/10.1038/44565

[7] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals Eugen.*, vol. 7, pp. 179–188, 1936.

[8] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.

[9] K. Messer, J. Matas, J. Kittler, and K. Jonsson, "Xm2vtsdb: The extended m2vts database," in *In Second International Conference on Audio and Video-based Biometric Person Authentication*, 1999, pp. 72–77.