A UNION OF INCOHERENT SPACES MODEL FOR CLASSIFICATION

K. Schnass and P. Vandergheynst

Signal Processing Laboratory (LTS2)
Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.
pierre.vandergheynst@epfl.ch

ABSTRACT

We present a new and computationally efficient scheme for classifying signals into a fixed number of known classes. We model classes as subspaces in which the corresponding data is well represented by a dictionary of features. In order to ensure low misclassification, the subspaces should be incoherent so that features of a given class cannot represent efficiently signals from another. We propose a simple iterative strategy to learn dictionaries which are are the same time good for approximating within a class and also discriminant. Preliminary tests on a standard face images database show competitive results.

Index Terms— classification, feature selection, subspace learning, Grassmannian manifolds, dictionary learning, alternate projections

1. INTRODUCTION

A powerful and general approach to the problem of classifying signals into classes is to first select informative features and then work out the classification in feature space. The role of the features is usually two-fold: they should reduce the dimensionality of the problem and they should make it possible to use very simple classification schemes, like nearest neighbour or nearest subspace [1, 2]. However it is often impossible to construct features a priori and one must then resort to learning them from training data.

Assume we have N already labelled training signals $y \in \mathbb{R}^d$ belonging to c classes, where each class i contains n_i elements, i.e. $\sum_i n_i = N$. We denote the j-th signal in class i as y_i^j, $i = 1 \dots c, j = 1 \dots n_i$. For each class i we collect all its training signals as columns in the $d \times n_i$ class matrix Y_i, i.e. $Y_i = (y_i^1 \dots y_i^{n_i})$, and these class matrices in turn are combined into a big $d \times N$ data matrix $Y = (Y_1 \dots Y_c) = (y_1^1 \dots y_1^{n_1} \dots y_c^1 \dots y_c^{n_c})$. Given a new signal y_{new} the goal is to decide which class it belongs to with the help of the already labelled training signals.

The scope of this paper is to recast the classification problem into finding a set of subspaces that each model a class where classification is achieved by finding the closest subspace to a test signal in an euclidean sense. Note that idea that each class should have its own representative system, learned from the training data can already be found in [8]. There frames or dictionaries for texture classification are learned, such that each provides a sparse representation for its texture class. The new texture then gets the label of the texture frame providing the sparsest representation. In [9], the same basic idea is used but the learning is guided by the principle that the dictionaries should also be discriminant, while in [10] both learning principles are combined, i.e. the dictionaries should be discriminant and approximative.

The main difficulty, and the contribution of this paper, is thus to learn an optimal set of subspaces from training data in a computationally efficient way. We provide results using the examples of classifying face images into classes corresponding to identities, an example for which ample comparative data is available [4, 5, 6].

2. CLASS MODEL

To every class i, we will assign a set of s_i vectors f_i^j, $j = 1 \dots s_i$, which are collected as columns in the matrix $F_i = (f_i^1 \dots f_i^{s_i})$. Every element y_i in class i can thus be written as a combination of these class specific features with coefficients x_i and some residual r_i, orthogonal to the feature span,

$$y_i = F_i x_i + r_i, \qquad r_i^k \perp sp(F_i). \tag{1}$$

To each class is thus assigned a subspace spanned by the feature vectors. In nearest subspace classification, the features of a test signal are projected onto all class subspaces and the subspace/class that carries the most energy will be selected. Naturally, the features of a class should thus represent very well all the elements of that class (the projection has high energy), while at the same time they shouldn't represent well the elements of any other class. Suppose for simplicity that the feature sets form orthonormal systems, i.e. $F_i^\star F_i = I_s$, they should thus satisfy :

$$\frac{\|F_j^\star y_i\|_2}{\|F_i^\star y_i\|_2} < 1, \forall j \neq i. \tag{2}$$

Let us justify qualitatively the choice of the 2-norm above. To choose a good p-norm for the classification, we bound the norm ratio we need to be small :

$$\frac{\|F_j^\star y_i\|_p}{\|F_i^\star y_i\|_p} \leqslant \frac{\|F_j^\star F_i x_i\|_p}{\|x_i\|_p} + \frac{\|F_j^\star r_i\|_p}{\|x_i\|_p}. \tag{3}$$

Since in most cases we do not have information about the distribution of the coefficients x_i, the first term on the right hand side can be as big as $\|F_j^\star F_i\|_{p,p} = \max_{\|x\|_p=1} \|F_j^\star F_i\|_p$. Taking into account the orthogonality of the features in the matrices F_i, we see that for $p = 2$ this term can only be equal to one if two classes overlap, meaning that there is a signal whose features in its own class can be represented by features in a different class. For $p = 1/\infty$, however, the corresponding term is equal to the maximum absolute column/row sum of the $F_j^\star F_i$ and it can be easily seen that this can be larger than one, even if for no signal the features in its own class can be fully represented by features in a different class. Similar results hold for all other $p \neq 2$, thus making $p = 2$ the best choice in this case. Observe also that $p = 2$ corresponds to measuring the energy captured by the features of a class. Thus if the features are well chosen also the second term in inequality (3) can be expected to be small.

To summarize, we see that choosing $p = 2$ puts the following incoherence constraint on the feature spaces. No signal that can be constructed from features in one class should be well representable by features in another class. This constraint is the strongest we have encountered so far, which is only natural since we do not have an assumption on coefficient distribution.

3. FINDING FEATURE/SENSING MATRICES

From the analysis in the last section we can derive two types of conditions that the collection of features or subspaces F_i needs to satisfy. The first type describes how features from different classes should interact, i.e. the interplay measured in the appropriate matrix norm should be small, and the second type how the features should interact with the training data, i.e. the ratio of the response without to within class should be small. The problem with both kinds of conditions is they are not linear and difficult to handle. For instance calculating the $(2, 2)$-norm is equivalent to finding the largest singular value and can already be too computationally intensive depending on the dimensionality of the problem. We will therefore simplify the problem. Instead of requiring explicitly that the interplay between features from different classes is small, hereby avoiding to investigate what small means quantitatively, we use the intuition that this should come as free side effect from regulating the interaction with the training data, and simply ask that F is a collection of orthonormal systems F_i each of rank s. What we would actually like to do about the interaction of the features with the training data is to minimise the ratio between the response of the training data without to within class. However, a constraint involving the ratio is not linear and very hard to handle. We will therefore split it into two constraints that guarantee that the ratio is small if they are fulfilled. The first constraint is that the response within class is equal to a constant β which we choose to be the maximally achievable value given the rank of the orthonormal systems. The second constraint is that the response without class is smaller than a constant μ, whose dependence on s, d is more complicated and will be discussed later. Define the two sets $\mathcal{F}_s$ and $\mathcal{F}_\mu$ as

$$\begin{aligned} \mathcal{F}_s &:= \{F = (F_1, \dots, F_c) : \; F_i^\star F_i = I_s\} \\ \mathcal{F}_\mu &:= \{F : \; \|F_i^\star y_i^k\|_2 = \beta, \\ &\qquad\qquad \|F_j^\star y_i^k\|_2 \leqslant \mu, \, \forall k, i, j \neq i\}, \end{aligned} \tag{4}$$

then our problems could be summarised as finding a matrix in the intersection of the two sets, i.e. $F \in \mathcal{F}_s \cap \mathcal{F}_\mu$. However, since this intersection might be empty, we should rather look for a pair of matrices, each belonging to one set, with minimal distance to each other measured in some matrix norm, eg. the Frobenius norm, denoted by $\|\cdot\|_{\mathbf{2}}$[1],

$$\min \|F_s - F_\mu\|_{\mathbf{2}} \text{ s.t. } F_s \in \mathcal{F}_s, \, F_\mu \in \mathcal{F}_\mu. \tag{5}$$

One line of attack is to use an alternate projection method, i.e. we fix a maximal number of iterations, an initialisation for F_s^0 and then in each iterative step do:

- find a matrix $F_\mu^k \in \operatorname{argmin}_{F \in \mathcal{F}_\mu} \|F_s^{k-1} - F\|_{\mathbf{2}}$
- check if $\|F_s^{k-1} - F_\mu^k\|_{\mathbf{2}}$ is smaller than the distance of any previous pair and if yes store F_s^{k-1}
- find a matrix $F_s^k \in \operatorname{argmin}_{F \in \mathcal{F}_s} \|F_\mu^k - F\|_{\mathbf{2}}$
- check if $\|F_s^k - F_\mu^k\|_{\mathbf{2}}$ is smaller than the distance of any previous pair and if yes store F_s^k

If both sets are convex, the outlined algorithm is known as Projection onto Convex Sets (POCS) and guaranteed to converge. Non convexity of possibly both sets, as is the case here, results in much more complex behaviour. Instead of converging, the algorithm just creates a sequence (F_μ^k, F_s^k) with at least one accumulation point, see [3] for more details on this algorithm and its properties.

As mentioned above we choose β to be the maximally achievable value. An orthonormal system of s feature vectors can maximally take out all the energy of a signal,

$$\|F_i^\star y_i\|_2 \leqslant \|y_i\|_2. \tag{6}$$

As the signals are assumed to have unit norm, this energy is at most one and we set $\beta = 1$. From the discussion in the last section we see that the parameter μ reflects the incoherence we require between features from different classes. If

[1] We use this notation instead of the more common variant $\|\cdot\|_F$ to avoid confusion.

we have $d \geqslant c \cdot s$, it is theoretically possible to have c subspaces of dimension s which are mutually orthogonal to each other, and μ could be zero. As soon as the above inequality is reversed, because for instance the actual dimension of the span of all features, i.e. $rank(F)$, is smaller than d, not all subspaces corresponding to the different classes can be orthogonal but will have to overlap. This overlap or coherence is measured by $\|F_j^\star F_i\|_{2,2}$ and from theory about Grassmannian manifolds, see [3], we know that the maximal coherence between two of c subspaces of dimension s embedded in the space $\mathbb{R}^d$ can be lower bounded by

$$\max_{i\neq j} \|F_j^\star F_i\|_{2,2}^2 \geqslant \frac{s \cdot c - d}{d(c-1)}. \tag{7}$$

The problem with setting μ as above is that we are not controlling the interaction between the sets of features directly but only indirectly over the training data. There the worst case might not be assumed and so μ as above would be too large. Therefore we use the above bound as an indication of order of magnitude and, when testing our scheme on real data, vary the parameter μ. Lastly for the initialisation for each class we choose the orthogonal system that maximises the energy taken from this class opposed to the energy taken from the other classes, i.e.

$$F^0_{s,i} = \operatorname*{argmin}_{F_i^\star F_i = I_s} \|F_i^\star Y_i\|_{\mathbf{2}}^2 - \sum_{j\neq i} \|F_i^\star Y_j\|_{\mathbf{2}}^2. \tag{8}$$

This problem can be easily solved, by considering the rewritten version of the function to minimise,

$$\min_{F_i^\star F_i = I_s} \operatorname{trace}\big(F_i^\star (Y_i Y_i^\star - \sum_{j\neq i} Y_j Y_j^\star) F_i\big). \tag{9}$$

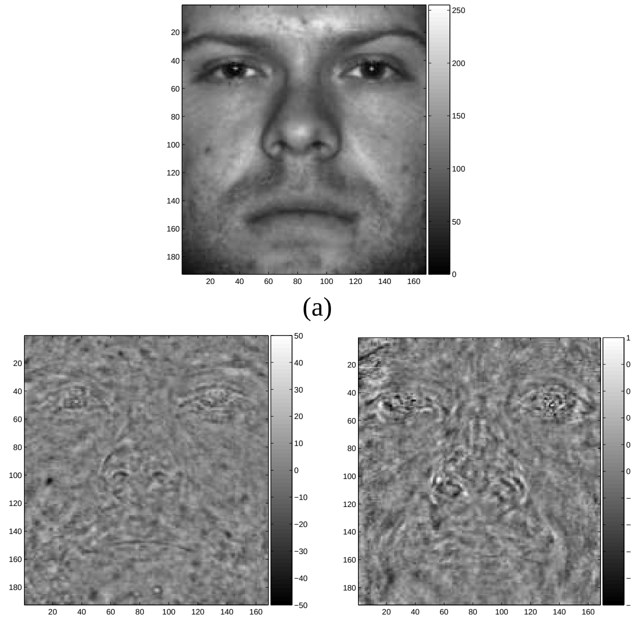
If $UDU^\star$ is an eigenvalue decomposition of the symmetric (Hermitian) matrix $Y_iY_i^\star - \sum_{j\neq i} Y_jY_j^\star$, then the minimum is attained for $F^0_{s,i}$ consisting of the s eigenvectors corresponding to the s largest eigenvalues.

4. TESTING

We tested our technique on the popular Yale B face image database [7] containing several instances of the same individual, with varying expressions, pose and lighting conditions. We used the 2414 frontal face images, about 64 images taken under varying illumination conditions for each of the 38 people. For the test we randomly split the set of images per person into an equal number of training and test images, using one more training than test image in case of an odd number of images per class. We then ran our classification scheme with the number of features per class varying from 2 to 5 and with the values of μ running only from 0 to 0.05. For comparison we ran Fisher's LDA with 37 and 30 discriminative axes in combination with the nearest neighbour classifier. This procedure was repeated 19 times and the mean of all 20 runs was computed.

The results of our method can be found in Table 1. While Fisher's LDA on average missclassified 23.30 ± 6.42 images (success rate of 98.07 ± 0.53%) using 37 discriminant axes and 231.55 ± 23.48 images (success rate 80.78 ± 1.95%) using 30 discriminant axes, our method in the best case only misclassified 13.60 ± 4.22 images (success rate 98.87 ± 0.35%). In general it outperformed Fisher's LDA for a wide range of values for μ and s.
Comparison to the ℓ_1-minimisation scheme in [4] is harder, as it seems that there only a single run was used. However, their best success rate of 98.26%, achieved at the same time as Fisher's LDA with 30 discriminant axes achieved 87.57% (the maximal rate for Fisher's LDA we encountered in 20 runs was 84.73%), is still below our best average rate of 98.87%.

To illustrate the results, we show in Figure 1 what happens when a training image is projected on the features of its own class and any other subject's class. As expected the projections on features of their own class nicely filter out common traits like eyes, mouths and noses, but on top of that the features of the first subject capture the very distinctive birth mark on his right cheek. The projections on the wrong class on the other hand are not only much weaker (note the difference in scale) but also less clear. Two overlapping sets of features seems to appear at the same time, the ones that belong to the subject in the image and the ones that the projection is trying to filter out.



(a)

(b) (c)

Fig. 1. Original image (a) projected onto the span of features from its own class (b), projected onto the span of features of the wrong class (c).

Summarising the results, we can say that our method outperforms a classic scheme like Fisher's LDA. In comparison

$s\backslash\mu$	0	0.01	0.02	0.03	0.04	0.05
2	19.80 ± 5.74	20.30 ± 5.80	22.25 ± 7.20	23.85 ± 6.81	25.25 ± 6.66	26.25 ± 6.61
3	14.15 ± 4.37	13.60 ± 4.22	13.85 ± 3.73	15.85 ± 5.25	16.40 ± 4.78	17.55 ± 6.00
4	15.75 ± 3.82	14.05 ± 3.49	13.95 ± 3.55	15.35 ± 3.95	16.45 ± 4.10	16.95 ± 4.30
5	15.70 ± 4.78	15.00 ± 4.91	14.45 ± 4.30	15.30 ± 3.34	17.60 ± 4.65	17.65 ± 4.55

Table 1. Mean ± standard deviation of misclassified images on the Extended Yale B database for varying values s and μ.

to the state-of-the-art ℓ_1-minimisation scheme in [4] it performs quite similarly. However it has one big advantage over the ℓ_1-minimisation scheme, which is its low computational complexity. Not taking the calculation of the feature matrices into account, as this is part of the pre-processing, basically all that has to be done to classify a new data vector is to multiply it with the feature matrix and calculate some statistics on the resulting vector. The ℓ_1 minimisation method on the other hand requires on top of extracting the features the solution of a convex optimisation problem

$$\min \|z\|_1 \text{ s.t. } \|f_{new} - Fz\|_2 \leqslant \varepsilon, \qquad (10)$$

where F in this case is the $d_f \times N$ matrix containing the features of all the training data. For comparison in [4] the authors state that the classification of one image takes a few seconds on a typical 3 GHz Pc. At the same time for classifying 1205 images of size 192×168, using our method with 4 feature dimensions per class, MATLAB takes less than half a minute on a Dual 1.8Ghz PowerPC G5, which is less than 25ms per image.

5. CONCLUSION

We have presented a classification scheme based on a model of incoherent subspaces, each one associated to one class, and a model on how the elements in a class are represented in this subspace. From a more practical viewpoint we have developed an algorithm to calculate these subspaces, i.e. the feature matrices, and shown that the scheme gives promising results on standard database, as compared with a state of the art method like the ℓ_1-minimisation scheme in [4]. An interesting direction for future research would be to try to reduce the computational cost in the training phase if d and N are very large. A possibility would be to first take random samples of the training data, which reduce their dimension but very likely preserve the geometrical structure, as explained in [4]. Alternatively to reduce the dimension of F one can apply our scheme on classical features, like Eigen or Laplace features, instead of directly on the raw training data.

6. REFERENCES

[1] R. Brunelli and T. Poggio, “Face recognition: Features vs. templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042–1053, 1993.

[2] K. Lee, J. Ho, and D.J. Kriegman, “Acquiring linear subspaces for face recognition under variable lighting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 27, no 5, pp. 684-698, 2005.

[3] J. Tropp, I. Dhillon, R Heath Jr, and T. Strohmer, “Designing structured tight frames via an alternating projection method,” *IEEE Transactions on Information Theory*, vol 51, no 1, pp. 188–209, 2005.

[4] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 31, no 2, 2009.

[5] M. Turk and A. Pentland, “Eigenfaces for recognition,” *In Proc. IEEE CVPR*, 1991.

[6] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, “Face recognition using laplacianfaces,” *In Proc. IEEE CVPR*, 2005.

[7] A. Georghiades, P.N. Belhumeur, and D.J. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 23, no 6, pp. 643–660, 2001.

[8] K. Skretting and J.H. Husoy, “Texture classification using sparse frame-based representations,” *EURASIP Journal on Applied Signal Processing*, vol 2006, no 11, 2006.

[9] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Discriminative learned dictionaries for local image analysis,” *IMA Preprint Series*, no 2212, 2008.

[10] F. Rodriguez and G. Sapiro, “Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries,” *IMA Preprint Series* , no 2213, 2008.