Dictionary Learning Based Dimensionality Reduction for Classification

Karin Schnass and Pierre Vandergheynst
Signal Processing Institute
Swiss Federal Institute of Technology
Lausanne, Switzerland
{karin.schnass, pierre.vandergheynst}@epfl.ch
EPFL-STI-ITS-LTS2
CH-1015 Lausanne
Tel: +41 21 693 2657
Fax: +41 21 693 7600
EDICS: SPC-CODC

***Abstract*— In this article we present a signal model for classification based on a low dimensional dictionary embedded into the high dimensional signal space. We develop an alternate projection algorithm to find the embedding and the dictionary and finally test the classification performance of our scheme in comparison to Fisher's LDA.**

Key words: dictionary learning, alternate projections, dimensionality reduction, classification, kernel-based learning

I. Introduction

When dealing with very high dimensional signals, like images or music, we are often interested in reducing their dimension in order to process or store them more efficiently, while at the same time keeping their key properties. A good example from every day life is image compression,i.e. jpeg [2]. The key property in this case are the images themselves. In order to store them more efficiently we are looking for a basis or dictionary which allows us to represent them as superposition of a small number of its elements so that in the end we can reduce each signal to a small number of coefficients. Another field where dimensionality reduction is important and which we are going to investigate is classification. The problem can be simply stated. Given a set of N training signals $x \in \mathbb{R}^d$ belonging to c classes and a new signal x_{new} find out which class the new signal belongs to. The probably simplest approach to solving this problem is to calculate the correlation between the new signal and the training signals and then give the new signal the same class label as the training signal with which the correlation is maximal. If we have collected the signals in class i as columns of the matrix X_i and then combined these to a big $d \times N$ data matrix $X = (X_1 \dots X_c) = (x_1^1 \dots x_1^{n_1} \dots x_c^1 \dots x_c^{n_c})$, the computational effort of this scheme amounts to calculating the matrix vector product $X^\star x_{new}$ and searching the resulting correlation vector for its absolute maximum, i.e. is of the order $\mathcal{O}(dN)$. There are however three disadvantages to this scheme. It is computationally intensive, it requires a lot of storage space - as both N and d can be very large - and most importantly it does not really work well. Let's ignore the first two problems for a moment and try to have a closer look at the third one. When correlating two complete signals we are ignoring the fact that the information that is related to the class labels may not be the whole signal but just parts of it. Take as easy example face recognition where our training data consist of 4 images of the same girl. In the first she is smiling, in the second she has a red clown nose, in the third she is wearing glasses and in the fourth has a red clown nose and glasses. If we now get a picture of a smiling girl with a red clown nose and glasses and want to know if she is smiling the correlation scheme will give us the wrong answer, because the similarity in the eye and nose region with picture 4 is larger than the similarity in the mouth region with picture 1. The obvious solution in this case is to concentrate on the mouth region and the simplest way to translate the extraction of the mouth region to a mathematical operation on our signals is a linear operator, ie a multiplication with a $p \times d$ matrix A. If we now use the correlation scheme on images of the signals under this operator we will not only find the right solution, but can also save storage space and computational effort. Instead of storing the $d \times N$ matrix of all training signals X we just need to store the $p \times N$ matrix of relevant features $F = AX$. For every new signal to classify we have to compute its image $f_{new} = Ax_{new}$, cost $\mathcal{O}(pd)$, and its correlation with the features $F^\star f_{new} = (AX)^\star(Ax_{new})$, cost $\mathcal{O}(pN)$, adding up to $\mathcal{O}(p(d+N))$ operations, which is smaller than the original cost of $\mathcal{O}(dN)$, as soon as

$$p < \frac{dN}{d+N}. \tag{1}$$

While in our toy example it was obvious that to identify a smile we should concentrate on the mouth region, this is not the case in general. The only information helping us choose the operator, i.e. find out which part of the signal we should look at, is the training signals and their class labels. The strategy is to choose the operator in a way that it increases the similarity of signals in the same class and decreases the similarity between classes. In our example, even if we would not know that smiles tend to manifest

themselves in the mouth region, by looking at all the smiling images we could see that they vary a lot in the eye and nose region, so in order to increase similarity we should ignore these regions which at the same time would decrease the similarity between the three smiles and the non smile. The objective increase/decrease similarity seems rather vague but it is exactly this - the definition of similarity - that leads to different methods. Techniques based on principal component analysis (PCA), cp. [5], choose an orthogonal projection that minimises the scatter of all projected samples. Fisher's Linear Discriminant Analysis, cp [1], [3], tries to maximise the ratio of between class scatter to within class scatter. In the approach we take here the similarity and dissimilarity is defined based on properties of the Gram matrix of the embedded images or features, $G = F^\star F = X^\star A^\star A X$. The Gram matrix is also called kernel matrix and the mapping from $x \mapsto Ax$ is called the kernel function. From the labels we decide what shape the ideal Gram matrix G^{best} should have, e.g. $G^{best}_{i,j}$ is one if x_i, x_j are in the same class and zero if they are not, and then we try to find the matrix A^{best} that results in a Gram matrix that is closest to the desired shape in some matrix norm,

$$A^{best} = \arg\min_A \|G^{best} - X^\star A^\star A X\|.$$

In the special case where the norm is the Frobenius norm and the matrix A^{best} is allowed to have the same rank as the signal matrix, we can reformulate the problem as minimisation over symmetric, semi-definite positive matrices K, which can later be factorised into a product $K = A^\star A$.

$$K^{best} = \arg\min_{K=K^\star, K\succeq 0} \|G^{best} - X^\star K X\|_F.$$

Since the class of feasible matrices K is convex this optimisation problem can be solved via semidefinite programming, see [6]. However to do dimensionality reduction we need to have $p = \mathrm{rk}(A) < \mathrm{rk}(X)$. Writing the problem again as minimisation over symmetric, semi-definite positive matrices, we get as additional constraint $\mathrm{rk}(K) = \mathrm{rk}(A) \leq p$, so the set of feasible matrices is no longer convex and semi-definite programming not applicable. To solve the problem we propose to use an alternative projection algorithm, extensively studied in [9], which has the additional advantage that we can easily replace the matrix G^{best} by a set of matrices. In the next section we introduce our class model and the resulting notion of similarity and dissimilarity. From that we infer the concrete properties of the embedding and subsequently discuss in how far they are achievable, depending on the number of classes and dimension of the embedding space. Section III is used to explain the concept of minimisation via alternate projections and customise the algorithm for our needs. In Section IV we show some promising results about the performance of our embeddings in comparison to existing schemes on the Yale face database before finally drawing conclusions and pointing out directions of ongoing further research.

II. Class Model

We want to characterise similarity and dissimilarity using the Gram matrix of the embedded signals or features, a notion based on the following class model. We assume that for every class we have a class specific unit norm feature vector f_i. These feature vectors live in a low-dimensional space $\mathbb{R}^p$ and have only small correlations i.e. $|\langle f_i, f_j\rangle| < \mu$, $i \neq j$, meaning they form a dictionary or frama of c elements with coherence μ. Signals in the same class, $x^k_i \in X_c$ are generated by taking the class specific feature vector f_i, scaling it with c^k_i, and mapping it with an invertible linear transform T to the higher dimensional space $\mathbb{R}^d$. Finally to model all the signal parts that contain no class specific information we add noise r^k_i, which is assumed to be orthogonal to the image of T, i.e. $|\langle r^k_i, Tv, |\rangle = 0$, $\forall v \in \mathbb{R}^p$.

$$x^k_i = T f_i c^k_i + r^k_i. \tag{2}$$

If we seek the analogy of the elements in the above model with our toy example the feature vectors correspond to the smiling or non smiling mouth and the noise to the eyes, glasses and (clown) noses. Applying T can be thought of as positioning the small picture of the mouth in the correct place in the picture of the whole face.

From the model we can directly see that the low dimensional embedding we are looking for is just the orthogonal projection onto the image of T concatenated with the inverse of T, since like this all signals in the same class are mapped back to scaled versions of the same feature vector. Assuming for the start that the scaling factor is constant over all signals and classes, i.e. $c^k_i = c$, this leads to a Gram matrix $G = X^\star A^\star A X$ of rank p with the following shape. Blocks $G_{ii} = X^\star_i A^\star A X_i$ storing inner products between embedded signals in the same class and therefore the same feature vectors are constant to c^2,

$$G_{ii}(k,l) = \langle A x^k_i,\ A x^l_i\rangle = \langle c f_i, c f_i\rangle = c^2,$$

while blocks $G_{ij} = X^\star_i A^\star A X_j$, $i \neq j$ storing inner products between embedded signals in different classes and therefore different feature vectors have entries of absolut value smaller μ,

$$|G_{ij}(k,l)| = |\langle A x^k_i, A x^l_j\rangle| = |\langle c f_i, c f_j\rangle| \leq c^2 \cdot \mu.$$

If we rescale A by $1/c$ we can formulate the problem of finding the right embedding as find a Gram matrix of the form $G = X^\star A^\star A X$ with rank $\mathrm{rk}(G) \leq p$, diagonal block entries equal to one and off-diagonal block entries smaller than μ.

Taking the desired dimension of the features p and their maximal correlation μ as input parameters we could go directly to the development of an algorithm constructing a corresponding Gram matrix. However, it will be instructive to first get an idea which magnitudes μ we can expect depending on the feature dimension and the number of different classes c. The ideal case in terms of minimising the correlation would be to have $\mu = 0$, meaning that the feature vectors form an orthonormal system. The drawback in this case is that we cannot have the dimensionality of the feature vectors, which determines the computational cost, smaller than the number of classes simply because we cannot fit more than p orthonormal vectors into a space of dimension p. Thus if we want to further reduce the cost we have to relax our requirement from having the inter-class correlations zero to having them small. The question is how small. From frame theory, see [8], we know

that for c unit norm vectors φ_i in $\mathbb{R}^p$ the maximal inner product can be lower bounded as

$$\max_{i\neq j} |\langle \varphi_i, \varphi_j \rangle| \geq \sqrt{\frac{c-p}{p(c-1)}} =: \mu_{p,c} \tag{3}$$

This lower bound is met with equality if and only if the frame is tight and equiangular, meaning that not only the maximal inner product but all of them have to be equal to $\mu_{p,c}$. Frames attaining the bound are called Grassmannian Frames. The problem with Grassmannian frames is that they are quite elusive and do not exist for all combinations of c and p, which makes it unlikely that our features can be modelled as one. On the other hand we know that it is not hard to construct dictionaries $\mathbb{R}^p$ with a lot of elements and keep the maximal correlation, i.e. the coherence, smaller than $1/\sqrt{p}$, cp [4].
As we want to find a collection of feature vectors forming an incoherent dictionary or frame we can also view the problem in the context of dictionary learning. Finding the embedding is equivalent to finding a subspace of our signal space and a dictionary of feature vectors, such that restricted to this subspace every signal has the ultimate sparse approximation $Ax_i^k = f_i c_i^k$. What sets the problem apart from regular dictionary learning and thus makes it much easier is that we know which dictionary element has to approximate which signal through the labels.
Keeping these theoretical considerations in mind we now turn to the development of an algorithm for constructing our desired Gram matrix.

III. Learning a Low-rank Embedding via Alternate Projections

We want to learn an embedding A such that we get a Gram matrix $G = X^\star A^\star A X$ with rank $\mathrm{rk}(G) \leq p$, diagonal block entries equal to one and off-diagonal block entries smaller than μ. So if we define the two sets of matrices

$$\mathcal{H}_\mu := \{H : \quad H_{ii}(k,l) = 1, \quad |H_{ij}(k,l)| \leq \mu,\ i \neq j\} \tag{4}$$

$$\mathcal{G}_p := \{G : \ G = X^\star A^\star A X,\ \mathrm{rk}(A) \leq p\} \tag{5}$$

and equip the space of all $N \times N$ matrices with the Frobenius norm we can write the problem as

$$\min \|G - H\|_F \ \text{ s.t } \ G \in \mathcal{G}_p, H \in \mathcal{H}_\mu. \tag{6}$$

One line of attack is to use an alternate projection method, i.e. we fix a maximal number of iterations and maybe some additional stopping criterion, initialise $G^0 = X^\star X$ and then in each iterative step do:

- find a matrix $H^k \in \arg\min_{H\in\mathcal{H}_\mu} \|G^{k-1} - H\|_F$
- find a matrix $G^k \in \arg\min_{G\in\mathcal{G}_p} \|H^k - G\|_F$
- check if G^k is better than what we have so far and if yes store it

After the last iteration we can extract our embedding and the feature dictionary from the best matrix G^{k_0}, which by definition is of the form $G^{k_0} = X^\star A^\star A X$. 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, however, results in much more complex behaviour, i.e. instead of converging the algorithm just creates a sequence (H^i, G^i) with at least one accumulation point. We will not discuss all the possible difficulties here but refer to the inspiring paper [9], where all details, proofs and background information can be found and wherein the authors conclude that alternate projection is a valid strategy for solving the posed problem.
So let's start investigating the two minimisation problems. The first problem, given a matrix G find

$$\arg\min_{H\in\mathcal{H}_\mu} \|G - H\|_F \tag{7}$$

is easy to solve since we can choose every component $H_{\cdot,\cdot}(k,l)$ in every block H_{ij} independently, i.e.

$$H_{ii}(k,l) = 1,$$
$$H_{ij}(k,l) = \min\{\mu, |G_{ij}(k,l)|\} \cdot \mathrm{sign}(G_{ij}(k,l)),\ i \neq j.$$

Bear in mind that if G is Hermitian also H will be Hermitian. The second problem, given a matrix H find

$$\arg\min_{\mathrm{rk}(A)\leq p} \|X^\star A^\star A X - H\|_F \tag{8}$$

is more intricate and so in order to keep the flow of the paper we will postpone its solution to the appendix. Note that in case the number of training signals per class is unbalanced the above problem should be replaced by its reweighted version, i.e. using matlab notation multiply the expression inside the norm from the left and the right by $\Omega = diag(ones(1,n_1)/n_1, \ldots, ones(1,n_c)/n_c)$. The analysis remains the same when replacing X by $X\Omega$ and H by $\Omega H \Omega$. Let us now turn to investigate how the proposed scheme performs in practice.

IV. Numerical Simulations& Comparison

We tested the dictionary based class model and the arising algorithm on the Yale Face Database[1], which contains 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. To test the performance and compare it to Fisher's LDA, cp. [1], we centred and normalised all images and employed the leave one out strategy. Taking every image in turn we used all the others to calculate the embedding, gave the image the label of the training image it was most correlated with under the embedding and counted how often this gave us the correct label. In case of the Fisher embedding we used the relative correlation $\frac{|\langle u,v\rangle|}{\|u\|_2\|v\|_2}$. This was done for the number of projections p varying from 5 to 14. To calculate the dictionary based embedding we fixed the number of iterations to 500 and kept the one giving the minimal distance to $\mathcal{H}_\mu$, where we once chose $\mu = \sqrt{\frac{c-p}{p(c-1)}}$ and once $\mu = 1/\sqrt{p}$. The results are displayed in Figure 1.

Note that for $\mu = \sqrt{\frac{c-p}{p(c-1)}}$ our scheme always outperforms the Fisher faces, and for $\mu = \mu = 1/\sqrt{p}$ all but once. For feature dimensions p close to the number of classes c the

[1] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

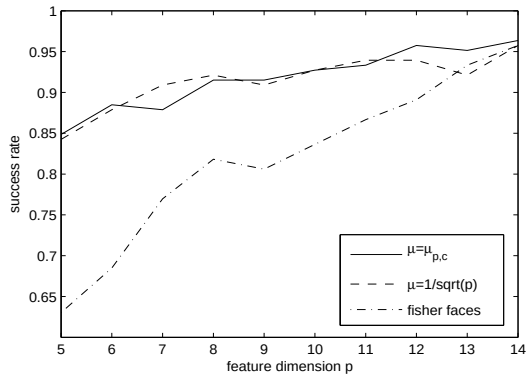

Fig. 1. Comparison of Fisher's to the dictionary based embedding for two choices for μ

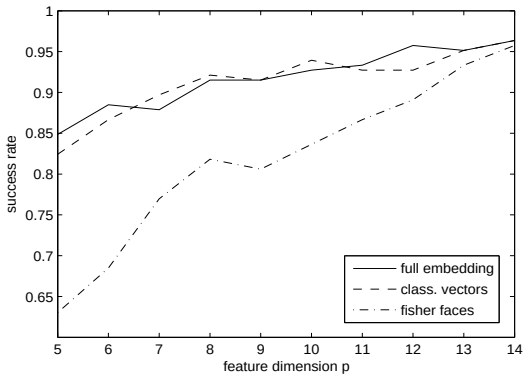

Fig. 2. Comparison of Fisher's embedding, the dictionary based embedding and the classification vectors

improvement is not drastic but becomes more significant as the feature dimension decreases. While the performance of the Fisher faces derails the dictionary based scheme turns out to be quite stable. Also we can see that it is stable in the choice of the maximally allowed inter class correlation or coherence μ.

V. Discussion

In the numerical section we demonstrated the promising performance of the embedding developed from the dictionary based signal model in combination with simple maximal correlation classification using all training signals. The question though is why did we use all embedded training signals for testing? According to the model A should map all training signals in the same class to the same feature vector, so correlating with embedded signals from the same class should give the same result,

$$\forall k: \ \langle Ax_i^k, Ax_{new}\rangle = \langle f_i, Ax_{new}\rangle.$$

Manipulating the expression a bit more,

$$\langle f_i, Ax_{new}\rangle = \langle A^\star f_i, x_{new}\rangle =: \langle s_i, x_{new}\rangle,$$

we see that we could actually classify x_{new} directly from its correlation with the classification vectors $s_i := A^\star f_i = A^\star Ax_i^k$. If we collect these vectors in the matrix $S = (s_1 \dots s_c)$, then the basic computational effort to classify is the multiplication $S^\star x_{new}$. Since S is of size $d \times c$ but as the image of the feature vector matrix has only rank p it can be decomposed into a $d \times p$ and a $p \times c$ matrix, e.g. by a reduced QR-decomposition $S = QR$, giving a computational cost for $S^\star x_{new}$ of $\mathcal{O}(p(d+c))$ as opposed to $\mathcal{O}(p(d+N))$, the cost of direct correlation.

The logical next step is to construct these classification vectors not through the embedding A but directly by looking for a $d \times c$ matrix S of rank p, such that $G = S^\star X$ consists of blocks $G_i = S^\star X_i$ with entries

$$\begin{aligned} \forall k: \qquad & G_i(i,k) = 1, \\ & |G_i(j,k)| \le \mu, \ i \neq j, \end{aligned}$$

which can again be calculated via alternate projection. Since the procedure is similar to the one described above, redefine $\mathcal{G}$ and $\mathcal{H}$, or the one used to calculate sensing dictionaries, cp. [7] and add a rank constraint, we will not detail it here but just show the test results for the leave one out strategy on the Yale Face database, $\mu = \sqrt{\frac{c-p}{p(c-1)}}$ cp Figure 2.

As can be seen the classification vectors perform as well as the embedding A, thus further confirming the usefulness of our signal model.

However remember that so far we have worked with the assumption that for each signal the contribution of its feature is constant, i.e.

$$x_i^k = Tf_i c_i^k + r_i^k \text{ and } c_i^k = c. \tag{9}$$

The first step in generalising the model is to allow this contribution to vary. Thinking back to our toy example this would for instance help to identify the smile under varying lighting conditions. While the energy in the mouth region is strong or weak compared to the rest of the image, the shape of the mouth remains the same. As a consequence of giving up the requirement that the contribution of the features is constant the correlation of two embedded signals will depend on their size. Thus if we want to see the underlying structure of the feature vectors we have to consider the relative instead of the absolute correlation of the embedded signals, i.e. denoting again with A the orthogonal projection onto the image of T concatenated with the inverse of T, we have

$$\frac{\langle Ax_i^k, Ax_i^l\rangle}{\|Ax_i^k\|_2 \|Ax_i^l\|_2} = 1$$

for two signals in the same class and

$$\frac{\langle Ax_i^k, Ax_j^l\rangle}{\|Ax_i^k\|_2 \|Ax_j^l\|_2} = \mu, \qquad i \neq j$$

for two signals from different classes. The problem of finding the embedding can now be formulated as find A such that the weighted Gram matrix, in matlab notation $G_\omega = diag(1./\|AX\|_2)X^\star A^\star AX diag(1./\|AX\|_2)$ has rank p and is close to the ideal shape. Again we can attack this problem via alternate projections. However preliminary results show that in order to avoid overfitting some more care has to be taken. We need to assume a balanced common contribution of all signals per class and that T and in consequence A are

not too badly conditioned, making the problem much more intricate and necessitating further study.

The second step in generalising the model is to allow more than one feature vector per class. If we collect all feature vectors corresponding to the same class as columns in the matrix F_i, we can model each signal as

$$x_i^k = TF_i c_i^k + r_i^k \tag{10}$$

where c_i^k is a vector instead of a scalar and r_i^k is assumed to be orthogonal to the image of all features in all classes. To see how this multiple feature model could be useful think again of face recognition. Assume we have 2 people and of each one picture with glasses, one with a clown nose and one smiling. To separate them we do not know on which region to concentrate, as mouth, eye and nose region are equally important or disturbed. On the other hand if we learn one image per person without glasses, clown nose and smile, which should be reasonably (un)correlated with the training images we may have a problem to identify a picture of the first person with glasses and a clown nose. The disturbance in the eye and nose region will mask any information we can get from the mouth region. If we however learn three features per person, i.e. eyes, nose and mouth and sum their absolute contribution we will be less affected by disturbances and able to identify a person from his image even if just one feature, in this example the mouth, is active.

While this class model seems very promising, it is obviously also more complex and will require a lot of further study. So we need to find out how exactly to model the c_i^k - balanced? or sparse?, how to model the inter and intra class correlations between features - should they form a higher order Grassmannian frames in analogy to the single feature model? Depending on our choice we need so seek the appropriate way to sum the contribution of the features, e.g. as q-norm for which q. Finally we need to find a way to learn the features that is not based on the Gram matrix, which will not contain relevant information anymore, but that is rather similar to the direct learning of the classification vectors.

APPENDIX

We want to find the solution to

$$\arg\min_{\mathrm{rk}(A)\leq p} \|X^\star A^\star A X - H\|_F. \tag{11}$$

First, we can square the objective function. Then we know that the Frobenius norm is invariant under multiplication with a unitary matrix. Therefore we can simplify the above expression using the singular value decomposition (SVD) of X and the reduced SVD, which we get by splitting the diagonal matrix S into its part containing the $s \leq \min(d, N)$ non zero singular values, and the unitary matrices U, V into the parts corresponding to the non zero singular values and the remainder.

$$\begin{aligned} X &= \underset{d\times d}{U} \cdot \underset{d\times N}{S} \cdot \underset{N\times N}{V^\star} \\ &= (U_1, U_2) \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^\star \\ V_2^\star \end{pmatrix} = \underset{d\times s}{U_1} \cdot \underset{s\times s}{S_1} \cdot \underset{s\times N}{V_1^\star} \end{aligned}$$

We now replace X in the squared version of (11) by its SVD,

$$\arg\min_{\mathrm{rk}(A)\leq p} \|VS^\star U^\star A^\star A U S V^\star - H\|_F^2 \qquad |\cdot V, V^\star\cdot,$$

and multiply the expression, whose norm we want to minimise, from the right with $V^\star$ and from the left with V. Doing some matrix-juggling, which we skip for conciseness, we finally arrive at

$$\begin{aligned} \arg\min_{\mathrm{rk}(A)\leq p} &\|S_1^\star U_1^\star A^\star A U_1 S_1 - V_1^\star H V_1\|_F^2 \\ &+ 2\|V_1^\star H V_2\|_F^2 + \|V_2^\star H V_2\|_F^2. \end{aligned}$$

Since the two rightmost terms in the above expression are independent of A it is equivalent to

$$\arg\min_{\mathrm{rk}(A)\leq p} \|S_1^\star U_1^\star A^\star A U_1 S_1 - V_1^\star H V_1\|_F^2.$$

Using the eigenvalue decomposition of $V_1^\star H V_1 = W\Sigma W^\star$, which exists and has only real eigenvalues because H and therefore $V_1^\star H V_1$ are Hermitian, we can further simplify to

$$\arg\min_{\mathrm{rk}(A)\leq p} \|W^\star S_1^\star U_1^\star A^\star A U_1 S_1 W - \Sigma\|_F^2.$$

For the last simplification observe that any feasible matrix A can be written as

$$A = \underbrace{AU}_{C} U^\star = (C_1, C_2) \begin{pmatrix} U_1^\star \\ U_2^\star \end{pmatrix} = C_1 U_1^\star + C_2 U_2^\star,$$

where $\mathrm{rk}(A) = \mathrm{rk}(C_1) + \mathrm{rk}(C_2)$ because U is unitary. However since the second $C_2 U_2^\star$ does not change the objective function we know that the minimal argument in is not unique and that for the minimum itself we have

$$\begin{aligned} &\min_{\mathrm{rk}(A)\leq p} \|W^\star S_1^\star U_1^\star A^\star A U_1 S_1 W - \Sigma\|_F^2 = \\ &\min_{\mathrm{rk}(C_1)\leq p} \|W^\star S_1^\star C_1^\star C_1 S_1 W - \Sigma\|_F^2 = \\ &\min_{\mathrm{rk}(B)\leq p} \|B^\star B - \Sigma\|_F^2. \end{aligned}$$

Thus it suffices to find a matrix B minimising the last expression and then reconstruct a projection matrix by setting $A = BW^\star S^{-1} U_1^\star$.

As Σ is a diagonal matrix also $B^\star B$ should be diagonal, which together with the considerations that $B^\star B$ is positive semidefinite and of rank maximally p leads to the problem of approximating the vector $\sigma = (\sigma_1 \dots \sigma_s) = (\Sigma_{11} \dots \Sigma_{ss})$ by a vector with maximally p non zero, positive entries.

$$\min_{\|b\|_0\leq b, b_i\geq 0} \|b - \sigma\|_2^2.$$

The solution to this problem finally is easy to find, ie choose $b_i = \sigma_i$ if σ_i is among the p largest positive components of σ and zero otherwise. Backtracing our steps, denoting by $I = (i_1 \dots i_p)$ the index set of the p largest positive components (or all if there are less than p) and writing $W = (w_1 \dots w_s)$, we get our final projection matrix as:

$$A = \begin{pmatrix} \sqrt{\sigma_{i_1}} w_{i_1}^\star \\ \vdots \\ \sqrt{\sigma_{i_p}} w_{i_p}^\star \end{pmatrix} S^{-1} U_1^\star.$$

REFERENCES

[1] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), July 1997.

[2] D.L. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies. Data compression and harmonic analysis. *IEEE Transactions on Information Theory*, 44:391–432, August 1998.

[3] R.A. Fisher. The use of multiple measures in taxonomic problems. *Ann. Eugenics*, 7:179–188, 1936.

[4] R Heath Jr, T. Strohmer, and A.J. Paulraj. On quasi-orthogonal signatures for CMDA systems. In *Allerton Conference on Communication, Control and Computers*, 2002.

[5] Ian T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 2002.

[6] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Gahoui, and M. Jordan. Learning the kernel matrix with semi-definite programming. In *ICML*, 2002.

[7] K. Schnass and P. Vandergheynst. Dictionary preconditioning for greedy algorithms. *accepted to IEEE Trans. Signal Processing*, 2007.

[8] T. Strohmer and R.W. Heath. Grassmannian frames with applications to coding and communication. *Appl. Comput. Harmon. Anal.*, 14(3):257–275, May 2003.

[9] J. Tropp, I. Dhillon, R Heath Jr, and T. Strohmer. Designing structured tight frames via an alternating projection method. *IEEE Transactions on Information Theory*, 51(1):188–209, January 2005.