KL Transform

G.A. GIRALDI[1]

[1]LNCC–National Laboratory for Scientific Computing -
Av. Getulio Vargas, 333, 25651-070, Petrópolis, RJ, Brazil
{gilson}@lncc.br

Abstract.

1 Introduction

In image databases it is straightforward to consider each data point (image) as a point in a n-dimensional space, where n is the number of pixels of each image. Therefore, dimensionality reduction may be necessary in order to discard redundancy and simplify further operations. The most know technique in this subject is the Principal Components Analysis (PCA) [1, 2].

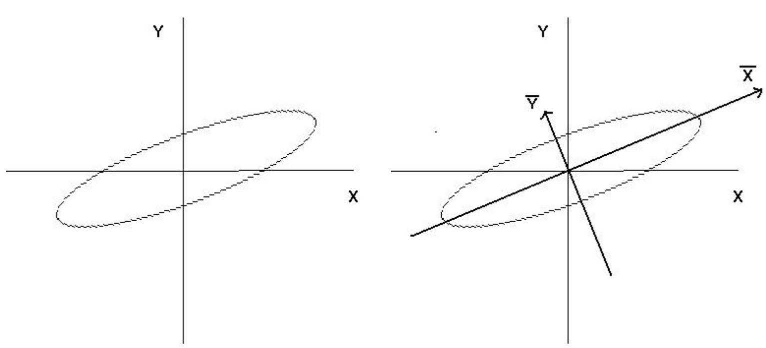
In what follows we review the basic theory behind PCA, the **KL Transform**, which is the best unitary transform from the viewpoint of compression.

2 KL Transform

The Karhunen-Loeve (or KL) transform, can be seen as a method for data compression or dimensionality reduction (see [3], section 5.11 also). Thus, let us suppose that the data to be compressed consist of N tuples or data vectors, from a n-dimensional space. Then, PCA searches for k n-dimensional orthonormal vectors that can best be used to represent the data, where $k \leq n$. Figure 1.a,b picture this idea using a bidimensional representation. If we suppose the the data points are distributed over the elipse, it follows that the coordinate system $(\left(\overline{X}, \overline{Y}\right)$, shown in Figure 1.b is more suitable for representing the data set in a sense that will be formally described next.

Thus, let $S = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N\}$ the data set represented on Figure 1. By now, let us suppose that the centroid of the data set is the center of the coordinate system, that means:

$$C_M = \frac{1}{N} \sum_{i=1}^{N} \mathbf{u}_i = \mathbf{0}. \tag{1}$$



(a) (b)

Figure 1: (a) Dataset and original coordinate system. (b) Principal directions $\overline{X}$ and $\overline{Y}$.

To address the issue of compression, we need a vector basis that satisfies a proper optimization criterium (rotated axes in Figure 1.b). Following [3], consider the operations in Figure 2. The vector $\mathbf{u}_j$ is first transformed to a vector $\mathbf{v}_j$ by the matrix (transformation) A. Thus, we truncate $\mathbf{v}_j$ by choosing the first m elements of $\mathbf{v}_j$. The obtained vector $\mathbf{w}_j$ is just the transformation of $\mathbf{v}_j$ by I_m, that is a matrix with 1s along the first m diagonal elements and zeros elsewhere. Finally, $\mathbf{w}_j$ is transformed to $\mathbf{z}_j$ by the matrix B. Let the square error defined as follows:

$$J_m = \frac{1}{N}\sum_{j=0}^{N}\|\mathbf{u}_j - \mathbf{z}_j\|^2 = \frac{1}{n}Tr\left[\sum_{j=0}^{N}(\mathbf{u}_j - \mathbf{z}_j)(\mathbf{u}_j - \mathbf{z}_j)^{*T}\right], \tag{2}$$

where Tr means the trace of the matrix between the square brackets and the notation $(*T)$ means the transpose of the complex conjugate of a matrix. Following Figure 2, we observe that $\mathbf{z}_j = BI_mA\mathbf{u}_j$. Thus we can rewrite (2) as:

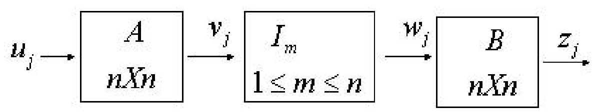


Figure 2: KL Transform formulation. Reprinted from [3].

$$J_m = \frac{1}{N}Tr\left[\sum_{i=0}^{N}(\mathbf{u}_j - BI_mA\mathbf{u}_j)(\mathbf{u}_j - BI_mA\mathbf{u}_j)^{*T}\right], \tag{3}$$

which yields:

$$J_m = \frac{1}{N}Tr\left[(I - BI_mA)\,R\,(I - BI_mA)^{*T}\right], \tag{4}$$

where:

$$R = \sum_{i=0}^{N}\mathbf{u}_j\mathbf{u}_j^{*T}. \tag{5}$$

Following the literature, we call R the covariance matrix. We can now stating the optimization problem by saying that we want to find out the matrices A, B that minimizes J_m. The next theorem gives the solution for this problem.

Theorem 1: The error J_m in expression (4) is minimum when

$$A = \Phi^{*T}, \quad B = \Phi, \quad AB = BA = I, \tag{6}$$

where Φ is the matrix obtained by the orthonormalized eigenvectors of R arranged according to the decreasing order of its eigenvalues.

Proof. To minimize J_m we first observe that J_m must be zero if $m = n$. Thus, the only possibility would be

$$I = BA \Rightarrow A = B^{-1}. \tag{7}$$

Besides, by remembering that

$$Tr\,(CD) = Tr\,(DC)\,, \tag{8}$$

we can also write:

$$J_m = \frac{1}{n}Tr\left[(I - BI_mA)^{*T}(I - BI_mA)\,R\right]. \tag{9}$$

Again, this expression must be null if $m = n$. Thus:

$$J_n = \frac{1}{n} Tr \left[\left(I - BA - A^{*T} B^{*T} + A^{*T} B^{*T} BA \right) R \right].$$

This error is minimum if:

$$B^{*T} B = I, \quad A^{*T} A = I, \tag{10}$$

that is, if A and B are unitary matrix. The next condition comes from the differentiation of J_m respect to the elements of A. We should set the result to zero in order to obtain the necessary condition to minimize J_m. This yields:

$$I_m A^{*T} \left(I - A^{*T} I_m A \right) R = 0, \tag{11}$$

which renders:

$$J_m = \frac{1}{n} Tr \left[\left(I - A^{*T} I_m A \right) R \right]. \tag{12}$$

By using the property (8), the last expression can be rewritten as

$$J_m = \frac{1}{n} Tr \left[R - I_m A R A^{*T} \right].$$

Since R is fixed, J_m will be minimized if

$$\tilde{J}_m = Tr \left[I_m A R A^{*T} \right] = \sum_{i=0}^{m-1} a_i^T R a_i^*, \tag{13}$$

is maximized where a_i^T is the *ith* row of A. Once A is unitary, we must impose the constrain:

$$a_i^T a_i^* = 1. \tag{14}$$

Thus, we shall maximize $\tilde{J}_m$ subjected to the last condition. The Lagrangian has the form:

$$\tilde{J}_m = \sum_{i=0}^{m-1} a_i^T R a_i^* + \sum_{i=0}^{m-1} \lambda_i \left(1 - a_i^T a_i^* \right),$$

where the λ_i are the Lagrangian multipliers. By differentiating this expression respect to a_i we get:

$$R a_i^* = \lambda_i a_i^*, \tag{15}$$

Thus, a_i^* are orthonormalized eigenvectors of R. Substituting this result in expression (13) produces:

$$\tilde{J}_m = \sum_{i=0}^{m-1} \lambda_i, \tag{16}$$

which is maximized if $\{a_i^*, \quad i = 0, 1, ..., m-1\}$ correspond to the largest eigenvalues of R. (**End of prof**)

A straightforward variation of the above statement is obtained if we have a random vector $\mathbf{u}$ with zero mean. In this case, the pipeline of Figure 2 yields a random vector $\mathbf{z}$ and the square error can be expressed as:

$$J_m = \frac{1}{n} Tr \left[E \left\{ \left(\mathbf{u} - B I_m A \mathbf{u} \right) \left(\mathbf{u} - B I_m A \mathbf{u} \right)^{*T} \right\} \right],$$

which can be written as:

$$J_m = \frac{1}{n} Tr \left[\left(I - B I_m A \right) R \left(I - B I_m A \right)^{*T} \right], \tag{17}$$

where $R = E\left(\mathbf{u}\mathbf{u}^{*T}\right)$ is the covariance matrix. Besides, if C_m in expression (1) is not zero, we must translate the coordinate system to C_m before computing the matrix R , that is:

$$\widetilde{\mathbf{u}_j} = \mathbf{u}_j - \mathbf{C}_m. \tag{18}$$

In this case, matrix R will be given by:

$$R = \sum_{i=0}^{N} \widetilde{\mathbf{u}_j}\widetilde{\mathbf{u}_j}^{*T}.$$

Also, sometimes may be useful to consider in expression (2) some other norm, not necessarily the 2-norm. In this case, there will be a real, symmetric and positive-defined matrix M, that defines the norm. Thus, the square error J_m will be rewritten in more general form:

$$J_m = \frac{1}{n}\sum_{j=0}^{N} \left\|\mathbf{u}_j - \mathbf{z}_j\right\|_M^2 = \frac{1}{n}\sum_{j=0}^{N} \left(\mathbf{u}_j - \mathbf{z}_j\right)^{*T} M \left(\mathbf{u}_j - \mathbf{z}_j\right). \tag{19}$$

Obviously, if $M = I$ we recover expression (2). The link between this case and the above one is easily obtained by observing that there is non-singular and real matrix W, such that:

$$W^T M W = I. \tag{20}$$

The matrix W defines the transformation:

$$W\widehat{\mathbf{u}_j} = \mathbf{u}_j, \quad W\widehat{\mathbf{z}_j} = \mathbf{z}_j. \tag{21}$$

Thus, by inserting these expressions in equation (19) we obtain:

$$J_m = \frac{1}{n}\sum_{j=0}^{N} \left(\widehat{\mathbf{u}_j} - \widehat{\mathbf{z}_j}\right)^{*T} \left(\widehat{\mathbf{u}_j} - \widehat{\mathbf{z}_j}\right). \tag{22}$$

Expression (22) can be written as:

$$J_m = \frac{1}{n}\sum_{j=0}^{N} \left\|\widehat{\mathbf{u}_j} - \widehat{\mathbf{z}_j}\right\|^2, \tag{23}$$

now using the 2-norm, like in expression (2). Therefore:

$$J_m = \frac{1}{n} Tr\left[\sum_{j=0}^{N} \left(\widehat{\mathbf{u}_j} - \widehat{\mathbf{z}_j}\right) \cdot \left(\widehat{\mathbf{u}_j} - \widehat{\mathbf{z}_j}\right)^{*T}\right]. \tag{24}$$

Following the same development performed above, we will find that we must solve the equation:

$$\widehat{R}\widehat{a_i^*} = \lambda_i \widehat{a_i^*}, \tag{25}$$

where:

$$\widehat{R} = \sum_{j=0}^{N} \widehat{\mathbf{u}_j}\widehat{\mathbf{u}_j}^{*T}. \tag{26}$$

Thus, from transformations (21) it follows that:

$$\widehat{R} = W R W^T. \tag{27}$$

and, therefore, we must solve the following eigenvalue/eigenvector problem:

$$\left(WRW^T\right)\widehat{a_i^*} = \lambda_i \widehat{a_i^*}. \tag{28}$$

The eigenvectors, in the original coordinate system, are finally given by:

$$W\widehat{a_i^*} = a_i^*. \tag{29}$$

Definition: Given an input vector $\mathbf{u}$, its KL transform is defined as:

$$\mathbf{v} = \Phi^{*T}\mathbf{u}, \tag{30}$$

where Φ is the matrix obtained by the orthonormalized eigenvectors of the associated covariance matrix R arranged according to the decreasing order of its eigenvalues.

3 Exercises

1. Apply the KL transform to a database. Compare the compression efficiency with DFT and DCT.

2. Show that the the components of vector $\mathbf{v}$ in expression (30) are uncorrelated and have zero mean, that is:

$$E\left[v\left(k\right)\right] = 0, \quad k = 0, 1, 2, ..., N-1, \tag{31}$$

$$E\left[v\left(k\right)v^*\left(l\right)\right] = 0, \quad k, l = 0, 1, 2, ..., N-1. \tag{32}$$

3. Extend the KL transform to an $N \times N$ image $u(m, n)$ (see [3], page 164-165).

References

[1] K. Fukunaga. Introduction to Statistical Patterns Recognition. *Academic Press, New York.*, 18(8):831–836, 1990.

[2] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[3] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., 1989.