PCA in Autocorrelation Space

Vlad Popovici
Signal Processing Institute
Swiss Federal Institute of Technology
1015 Lausanne, Switzerland
vlad.popovici@epfl.ch

Jean-Philippe Thiran
Signal Processing Institute
Swiss Federal Institute of Technology
1015 Lausanne, Swizterland
jp.thiran@epfl.ch

Abstract

The use of higher order autocorrelations as features for pattern classification has been usually restricted to second or third orders due to high computational costs. Since the autocorrelation space is a high dimensional space we are interested in reducing the dimensionality of feature vectors for the benefit of the pattern classification task.

An established technique is Principal Component Analysis (PCA) which, however, cannot be applied directly in the autocorrelation space. In this paper we develop a new method for performing PCA in autocorrelation space, without explicitly computing the autocorrelations. The connections with the nonlinear PCA and possible extensions are also discussed.

1. Introduction

In most pattern recognition problems, each pattern can be viewed – at a raw level – as a scalar function of time or spatial coordinates. In many cases, a translation or a scale change has no effect on class membership. Regarding each pattern as a point in a vector space, we wish to map all points corresponding to translated (or scaled) versions of one pattern in a single point. In addition, patterns which differ in other ways should map into distinct points, and in some sense, patterns which are similar should map into points that are close together.

The higher-order measures possess the uniqueness property for even orders [6] and they are shift–invariant. Higher-order autocorrelations have been previously used as features describing patterns ([4], [5], [6], [8]), but their applicability has been usually limited to second or third orders and a small local neighborhood, due to high computational costs. Exceptions are [6] and [8] where the use of higher order autocorrelations has been shown to increase the discrimination power.

On the other hand, the dimensionality of the autocorrelation space grows combinatorially as the order of autocorrelations increases. This justifies the search of a lower dimensional space in which data could be represented accurately enough, while preserving its intrisic dimensionality. A standard technique for dimensionality reduction is Principal Component Analisys (PCA). However, performing PCA requires the covariance matrix to be computed, an operation which is much too expensive in the case of higher order autocorrelation features as it relies on the explicit computation of the autocorrelation vectors. By consequence, other ways of performing PCA in autocorrelation space must be sought. In this paper we develop such a technique which avoids direct computation of autocorrelations. Also, we will show its connections with non-linear versions of PCA, such as Kernel PCA.

The paper is organized as follows: Section 2 discusses general properties of autocorrelation functions, Section 3 develops the PCA in autocorrelation space, Section 4 presents some experimental results and, finally, Sections 5 and 6 discuss some extensions to nonlinear PCA and different neighborhood topologies and draw some conclusions.

2. Higher Order Autocorrelations

Let $\psi : D \subseteq \mathbb{R}^m \to \mathbb{R}$ be a real-valued function. Then the associated n-th order autocorrelation function is defined as

$$r_\psi^{(n)}(\tau_1, \ldots, \tau_n) = \int_D \psi(t) \prod_{k=1}^n \psi(t + \tau_k) dt \qquad (1)$$

It is easy to see that $r_\psi^{(n)}$ is shift-invariant, in the sense that $\psi(t)$ and $\psi(t+\tau)$ have the same n-th order autocorrelation.

On the other hand, for two functions ψ_1 and ψ_2 it can be proven ([6]) that the second order (and higher even order) autocorrelation functions are equal only if $\psi_1(t) = \psi_2(t+\tau)$, meaning that the two patterns have the same representation in autocorrelation space if ψ_2 is a shifted version of ψ_1.

This also means that ψ_1 can be recovered from $r_{\psi_1}^{(2k)}$ except for an unknown translation τ. Generally, in the case of pattern recognition, this is a valuable property.

Considering the set of admissible values for τ_k being discrete and having m_k distinct values, it follows that the space of n-th order autocorrelations has $\prod_{k=1}^{n} m_k$ dimensions, making the explicit computation of autocorrelations prohibitively expensive.

For two given functions ψ_1 and ψ_2, the inner product of the corresponding n-th order autocorrelations is given by ([8]):

$$\langle \mathbf{r}_1, \mathbf{r}_2 \rangle = \int_T \left\{ \int_D \psi_1(t)\psi_2(t+\tau)dt \right\}^{n+1} d\tau \quad (2)$$

For any ψ, the values of $r_\psi^{(n)}(\tau_1, \ldots, \tau_n)$ can be ordered sequentially, obtaining a (column) vector $\mathbf{r}_\psi^{(n)}$. To simplify notation, we will denote by $\mathbf{r}_k$ the vector [1] corresponding to $\mathbf{r}_{\psi_k}^{(n)}$. In the following, we will investigate the properties of those vectors, using the discrete version of (2):

$$\langle \mathbf{r}_1, \mathbf{r}_2 \rangle = \sum_{\tau \in T} \left\{ \sum_{t \in D} \psi_1(t)\psi_2(t+\tau) \right\}^{n+1} \quad (3)$$

where τ and t take values in the discrete corresponding sets T and D, respectively.

Let now $\{\mathbf{r}_1, \ldots, \mathbf{r}_m\}$ be a set of linearly independent autocorrelation vectors (not necessarily orthogonal) and let $\mathbf{r}$ be a new vector to be projected on the space spanned by $\{\mathbf{r}_k\}_{k=1}^m$. The vector $\mathbf{r}$ can be decomposed into two components:

$$\mathbf{r} = \mathbf{r}_W + \mathbf{r}_{\perp W} \quad (4)$$

where $\mathbf{r}_W \in W = \mathrm{Span}(\{\mathbf{r}_k\})$ and $\mathbf{r}_{\perp W}$ is orthogonal on W. Since $\mathbf{r}_W$ is a linear combination of vectors $\mathbf{r}_1, \ldots, \mathbf{r}_m$, there exists a vector $\mathbf{c} \in \mathbb{R}^m$ such as

$$\mathbf{r}_W = R\mathbf{c} \quad (5)$$

where $R = [\mathbf{r}_1 | \ldots | \mathbf{r}_m]$ is the matrix built with the basis vectors. From (4) and using the fact that $\mathbf{r}_{\perp W}$ is orthogonal on W we have

$$R'\mathbf{r} = R'\mathbf{r}_W = R'R\mathbf{c} \quad (6)$$

It follows that

$$\begin{aligned} \mathbf{c} &= (R'R)^{-1}R'\mathbf{r} \\ \mathbf{r}_W &= R(R'R)^{-1}R'\mathbf{r} \end{aligned} \quad (7)$$

---

[1] The transpose of a vector $\mathbf{r}$ will be denoted by $\mathbf{r}'$. The similar notation will be used for the transposed matrices.

Thus we have the orthogonal projection of $\mathbf{r}$ on W. Further, we can obtain the modulus of the projection and the distance from the space W by

$$\|\mathbf{r}_W\|^2 = \langle \mathbf{r}_W, \mathbf{r}_W \rangle = (R'\mathbf{r})'(R'R)^{-1}(R'\mathbf{r}) \quad (8)$$

$$\begin{aligned} \|\mathbf{r}_{\perp W}\|^2 &= \|\mathbf{r}\|^2 - \|\mathbf{r}_W\|^2 = \\ \langle \mathbf{r}, \mathbf{r} \rangle &- (R'\mathbf{r})'(R'R)^{-1}(R'\mathbf{r}) \end{aligned} \quad (9)$$

Note that all products $R'\mathbf{r}$ and $R'R$ imply only computations of inner products between autocorrelation vectors which can be computed by means of (2)-(3), avoiding the explicit computation of autocorrelations.

3. PCA in Autocorrelation Space

Principal Component Analysis (PCA) is a technique for extracting the structure from a high-dimensional data set. PCA can be viewed as an orthogonal transformation of the coordinate system in which the data is described. This transformation is performed in the hope that a small number of principal directions will suffice to well-approximate the data. There are different methods to perform PCA, the most common requiring the diagonalization of the covariance matrix, or, equavalently, to solve the eigenproblem

$$C\lambda_i = \lambda_i \mathbf{v}_i \quad (10)$$

where C is the covariance matrix and λ_i are the eigenvalues corresponding to the eigenvectors $\mathbf{v}_i$. Naturally, we are interested only in the non-trivial solution of (10).

Let $\{\mathbf{r}_k\}$ be a set of mean-centered autocorrelation vectors. The equivalent problem of (10) is

$$RR'\mathbf{v}_i = \lambda_i \mathbf{v}_i \quad (11)$$

where the elements of the matrix RR' are formed by outer products $\mathbf{r}_i\mathbf{r}_j'$. The rank of RR' cannot exceed m, the number of data/autocorrelation vectors, even if its dimensionality is usually much bigger than $m \times m$. We can solve the problem (11) indirectly, by first solving the smaller problem

$$R'R\mathbf{w}_i = \lambda_i \mathbf{w}_i. \quad (12)$$

By left-multiplying with R we obtain

$$(RR')(R\mathbf{w}_i) = \lambda_i(R\mathbf{w}_i) \quad (13)$$

If $\lambda_i \neq 0$ (the only case we are interested in), then

$$\mathbf{v}_i = \frac{R\mathbf{w}_i}{\sqrt{\lambda_i}} \quad (14)$$

is the solution of (11) when $\mathbf{w}_i$ is the solution of (12). Since the ranks of RR' and $R'R$ are equal, there are no eigenvectors missed or added by this indirect method.

Then, the projections of the vectors $\{\mathbf{r}_k\}_k$ on the principal directions will be given by

$$\mathbf{a}_i = R'\mathbf{v}_i = \frac{R'R\mathbf{w}_i}{\sqrt{\lambda_i}} \qquad (15)$$

Generally, $\mathbf{v}_i$ are not valid n-th order autocorrelations so the projection of a vector $\mathbf{r}$ on $\mathbf{v}_i$ cannot be computed directly as a simple inner product. Instead we have to use

$$\mathbf{r}'\mathbf{v}_i = \frac{\mathbf{r}'R\mathbf{w}_i}{\sqrt{\lambda_i}} \qquad (16)$$

All of the above development has been done supposing that the vectors $\mathbf{r}_k$ are centered around their mean. We will remove now this restriction and we will prove that the centering in the autocorrelation space can be carried out indirectly, without computing the autocorrelations. In (11-16) we have to replace the matrix R with R_* where $R_* = [\mathbf{r}_1 - \bar{\mathbf{r}}| \ldots |\mathbf{r}_m - \bar{\mathbf{r}}]$, with $\bar{\mathbf{r}}$ being the mean autocorrelation vector. Computing the product $R_*'R_*$ reduces to compute the inner products $\langle \mathbf{r}_i - \bar{\mathbf{r}}, \mathbf{r}_j - \bar{\mathbf{r}} \rangle$ for all $i, j = 1, \ldots, m$:

$$\langle \mathbf{r}_i - \bar{\mathbf{r}}, \mathbf{r}_j - \bar{\mathbf{r}} \rangle =$$

$$= \langle \mathbf{r}_i, \mathbf{r}_j \rangle - \frac{1}{m}\sum_{k=1}^{m}\langle \mathbf{r}_i, \mathbf{r}_k \rangle$$

$$- \frac{1}{m}\sum_{k=1}^{m}\langle \mathbf{r}_j, \mathbf{r}_k \rangle + \frac{1}{m^2}\sum_{k,l=1}^{m}\langle \mathbf{r}_k, \mathbf{r}_l \rangle \qquad (17)$$

which translates into

$$R_*'R_* = R'R - \frac{1}{m}1_{mm}(R'R) - \frac{1}{m}(R'R)1_{mm}$$

$$+ \frac{1}{m^2}1_{mm}(R'R)1_{mm} \qquad (18)$$

where 1_{mm} is an $m \times m$ matrix of ones.

Finally, we have to compute the projection of $\mathbf{r}_* = \mathbf{r}_0 - \bar{\mathbf{r}}$ on the principal axis, where $\mathbf{r}_0$ is a new autocorrelation vector which has to be projected on the principal directions. Similar to (18), we have:

$$\mathbf{r}_*'R_* = \mathbf{r}_0'R - \frac{1}{m}1_{1m}(R'R) - \frac{1}{m}(\mathbf{r}_0'R)1_{mm}$$

$$+ \frac{1}{m^2}1_{1m}(R'R)1_{mm} \qquad (19)$$

and from (16) we have the projection on the i-th principal direction:

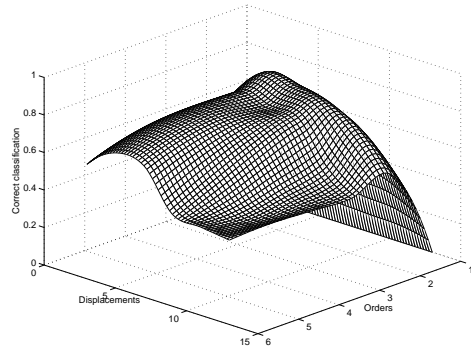$$\mathbf{r}_*'\mathbf{v}_{*i} = \frac{\mathbf{r}_*'R_*\mathbf{w}_{*i}}{\sqrt{\lambda_{*i}}} \qquad (20)$$

Figure 1. Classification rate as a function of autocorrelation parameters (orders and displacements)

where $\mathbf{v}_{*i}, \mathbf{w}_{*i}$ and λ_{*i} are obtained by considering equations (12) and (14) with R_* replaced for R.

4. Experiments

In order to assess the validity of the method presented here, we carried out a number of tests on the dataset *Waveform* from the UCI database ([1]). The set consists of 5000 samples of $1D$ signals, distributed equally in 3 classes. The goal of the experiment was to study the influence of different parameters in a binary classification task: discriminate between the first class (called A) and the other two (B and C) (see Figure 1).

The discrimination function was based on the *distance from feature space* (Dffs):

$$\mathbf{r} \in A \iff \text{Dffs}(\mathbf{r}, A) \leq \theta \qquad (21)$$

In all experiments, 500 vectors from class A (randomly chosen) have been used to perform PCA in autocorrelation space, and to determine the threshold. Other 500 vectors from class A and 500 from classes B and C have been used to test the classification. Table 1 presents some results obtained by autocorrelations features in comparison with some other methods (notation ACorr(n,d) is used to designate an autocorrelation function of order n applied on a neighborhood of size d).

5. Extensions

5.1. Kernel PCA in Autocorrelation Space

While developing PCA in autocorrelation space we were interested in expressing all the computations in terms of

Table 1. Classification rates (%)

Method	Rate
Optimal Bayes classifier ([1])	**86.0%**
Nearest Neighbor ([1])	78.0%
CART decision tree ([1])	72.0%
ACorr(2,5)	**80.5%**
ACorr(2,7)	79.0%
ACorr(3,5)	**81.6%**
ACorr(3,7)	78.0%
ACorr(4,5)	**81.5%**
ACorr(4,7)	**80.0%**

inner products of autocorrelation vectors. As a result, the eigenproblem to be solved was

$$R'_* R_* \mathbf{w}_{*i} = \lambda_{*i} \mathbf{w}_{*i} \qquad (22)$$

Now, if $\Phi : A \to \mathbb{R}^p$ is a mapping from the autocorrelation space to some feature space, then the matrix $K_* = R'_* R_*$ can be viewed as a kernel matrix ([11]) for $\Phi(\mathbf{r}) = \mathbf{r}$. In general, the kernel matrix has the following form:

$$K = [\langle \Phi(\mathbf{r}_i), \Phi(\mathbf{r}_j) \rangle]_{ij} \qquad (23)$$

but, for the majority of the kernel functions used in practice, it can be expressed as

$$K = [\Psi(\langle \mathbf{r}_i, \mathbf{r}_j \rangle)]_{ij} \qquad (24)$$

The latter form can be directly plugged into (18) resulting in a kernel–based PCA in autocorrelation space.

5.2. Multi–order Autocorrelation Vectors

Combining autocorrelations of different orders in the same feature vector proved to be a useful technique ([8]). This can be achieved by simply considering the extended feature vectors $\mathbf{z}_\psi^{\{n_1,\ldots,n_k\}} = \left[(\mathbf{r}_\psi^{(n_1)})', \ldots, (\mathbf{r}_\psi^{(n_k)})' \right]'$, obtained by concatenating the corresponding autocorrelation vectors. Again, the inner product can be computed indirectly, without computing the autocorrelation vectors themselves:

$$\left\langle \mathbf{z}_{\psi_1}^{\{n_1,\ldots,n_k\}}, \mathbf{z}_{\psi_2}^{\{n_1,\ldots,n_k\}} \right\rangle = \sum_{n \in \{n_1,\ldots,n_2\}} \langle \mathbf{r}_{\psi_1}^{(n)}, \mathbf{r}_{\psi_2}^{(n)} \rangle \qquad (25)$$

On the other hand, one may consider the local neighborhood over which the autocorrelations are computed (set T in (2)) as being a union of partitions, obtaining a similar formula, but with summation over the set of partitions.

6. Conclusions

In this paper we have shown that even if the autocorrelation feature vectors lie in a very high dimensional space, using them is still a feasible task as long as the explicit computing of autocorrelations can be avoided. The autocorrelation functions possess a useful property alowing the easy computation of their inner product, without computing the autocorrelations themselves. This property is exploited and a new method for performing PCA in autocorrelation space is developped. The experimental results are very promising and justify the presented approach.

References

[1] C. Blake and C. Merz. UCI repository of machine learning databases. Technical report, University of California, Irvine, Dept. of Information and Computer Sciences, 1998.

[2] M. Á. Carreira-Perpiñán. A review of dimension reduction techniques. Technical Report CS-96-09, Dept. of Computer Science, University of Sheffield, January 1997.

[3] K. I. Diamantaras and S. Y. Kung. *Principal Component Neural Networks*. Adaptive and Learning Systems for Signal Processing, Communications, and Control. Wiley & Sons, 1st edition, 1996.

[4] T. Kurita, K. Hotta, and T. Mishima. Scale and rotation invariant recognition method using higher-order local autocorrelation features of log-polar image. In *Third Asian Conference on Computer Vision*, pages 89–96, 1998.

[5] T. Kurita, N. Otsu, and T. Sato. A face recognition method using higher order local autocorrelation and multivariate analisys. In *11th IAPR International Conference on Pattern Recognition*, pages 213 – 216, The Hague, 1992.

[6] J. A. McLaughlin and J. Raviv. Nth-order autocorrelations in pattern recognition. *Information and Control*, 12:121 – 142, 1968.

[7] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de–noising in feature spaces. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, 1999.

[8] V. Popovici and J. P. Thiran. Higher order autocorrelations for pattern classification. In *Proceedings of the International Conference on Image Processing (ICIP)*. IEEE, 2001.

[9] V. Roth and V. Steinhage. Nonlinear discriminant analysis using kernel functions. In S. Solla, T. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 568–574. MIT Press, 1999.

[10] B. Schölkopf, S. Mika, A. Smola, G. Rätsch, and K.-R. Müller. Kernel PCA pattern reconstruction *via* approximate pre-images. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proceedings of the 8th International Conference on Artificial Neural Networks*, pages 147–152. Springer Verlag, 1998.

[11] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.