Segmentation and dimensionality reduction

Ella Bingham Aristides Gionis Niina Haiminen Heli Hiisilä Heikki Mannila
Evimaria Terzi

HIIT Basic Research Unit
University of Helsinki and Helsinki University of Technology
Finland

Abstract

Sequence segmentation and dimensionality reduction have been used as methods for studying high-dimensional sequences — they both reduce the complexity of the representation of the original data. In this paper we study the interplay of these two techniques. We formulate the problem of segmenting a sequence while modeling it with a basis of small size, thus essentially reducing the dimension of the input sequence. We give three different algorithms for this problem: all combine existing methods for sequence segmentation and dimensionality reduction. For two of the proposed algorithms we prove guarantees for the quality of the solutions obtained. We describe experimental results on synthetic and real datasets, including data on exchange rates and genomic sequences. Our experiments show that the algorithms indeed discover underlying structure in the data, including both segmental structure and interdependencies between the dimensions.

Keywords: segmentation, multidimensional data, PCA, time series

1 Introduction

The goal in segmentation is to decompose the sequence into a small number of homogeneous pieces, *segments*, such that the data in each segment can be described accurately by a simple model, for example a constant plus noise. Segmentation algorithms are widely used for extracting structure from sequences; there exist a variety of applications where this approach has been taken [12, 13, 16, 17, 19, 20]. Sequence segmentation is suitable in the numerous cases where the underlying process producing the sequence has several relatively stable states, and in each state the sequence can be assumed to be described by a simple model.

Naturally, dividing a sequence into homogeneous segments does not yield a perfect description of the sequence. For a multidimensional time series, a segmentation of the series into k pieces leaves open the question of the relationships between segments: are the representative values for different segments somehow connected to each other?

Example. As a simple example consider the case of analyzing a dataset of financial time series, such as stock or currency prices. The dataset can be viewed as a multidimensional time series; each stock or currency time series forms a separate dimension having one observation per time unit (day, hour, etc.). The set of time series is naturally synchronized on the time axis. Segmenting this financial dataset corresponds to splitting the time into different phases for the economy, such as recession, recovery, expansion, market behavior after a terrorist attack, etc. On the other hand, it is clear that there are a lot of interdependencies among the dimensions. Many series can be explained in part by using a small number of underlying variables, e.g., oil price, general state of different sectors of the economy, monetary policy, etc. Furthermore, the dependency of a time series from the underlying variables might be different at different periods of time. For example, the stock of a biotech company might in general follow the trend dictated by government support on R&D, but not so much so during a period following the announcement of a new exciting technology in the field. □

In this paper we study the following problem. Given a multidimensional time series, find a small set of latent variables and a segmentation of the series such that the data in each segment can be explained well by some (linear) combination of the latent variables. We call this problem the *basis segmentation problem*. In the previous example, we want to discover the time periods of the market and the underlying variables that explain the behavior of the financial time series well, but we are also interested in how each series is affected by the underlying variables during each time period.

The generative model behind our problem definition is as follows. First we assume that a small number of latent variables are present and responsible for the generation of the whole d-dimensional sequence. We call such latent variables the *basis* of the sequence. The values of the latent variables are d-dimensional vectors, and the number m of the latent variables satisfies $m < d$; typically, we want m to be considerably smaller than d. The sequence consists of segments generated by different states of the underlying process. In each state, a d-dimensional vector is created by a linear combination of the basis vectors using arbitrary coefficients. Data points are generated from that vector by a noisy process.

Our problem formulation allows decomposing the sequences into segments in which the data points are explained by a model unique to the segment, yet the whole sequence can be explained adequately by the vectors of the basis.

To solve the basis segmentation problem, we combine existing methods for sequence segmentation and for dimensionality reduction. Both problems are very well studied in the area of data analysis and they can be used to reduce the complexity of the representation of the original data. Our primary algorithmic tools are (i) k-segmentation, an optimal algorithm based on dynamic programming for segmenting a sequence into k pieces, and (ii) Principal Component Analysis (PCA), one of the most commonly used methods for dimensionality reduction.

We give three different algorithms for solving the basis segmentation problem, each of which combines k-segmentation and PCA in different ways. For two of our algorithms we are able to show that the cost of the solutions they produce is at most 5 times the cost of the optimal solution.

We have performed extensive experimentation with synthetic and real datasets, and demonstrate the ability of the approach to discover underlying structure in multidimensional sequences.

The rest of the paper is organized as follows. First we discuss related work in Section 2. Section 3 presents the notation and defines the problem. In Section 4 we describe the three basic algorithms, and give the theoretical bounds on the approximation ratio for two of them. Empirical results on simulated and real data are given in Section 5, and Section 6 is a short conclusion.

2 Related work

Segmentation of time series has been considered in many areas, starting from the classic paper of Bellman [4]. Many other formulations exist and the problem has been studied extensively in various settings. To name a few approaches, Himberg et al. [12] compare a large number of different algorithms on real-world mobile-device data. Keogh et al. [13] show how one can use segmentation in order to obtain efficient indexing of time series. Guha et al. [10] provide a graceful trade-off between the running time and the quality of the obtained solution. Azad et al. [2], Li [16], and Ramensky et al. [20] apply segmentation on genomic sequences, while Koivisto et al. [15] use segmentation to find blocks on haplotypes. One should note that in statistics the question of segmentation of a sequence or time series is often called the change-point problem [6].

Dimensionality reduction through Principal Component Analysis (PCA) and its variants is one of the most widely used methods in data analysis [3, 7, 18]. Closer to our paper is the work by Attias [1] who applied Independent Factor Analysis in order to identify a small number of statistically independent sources in multidimensional time series. Also, Bingham et al. [5] analysed dynamically evolving chat room discussions, finding distinct topics of discussion. However, to the best of our knowledge, this is the first paper that studies the connections between segmentation and dimensionality reduction in multidimensional sequences.

In the experimental section, we compare our algorithms with the (k, h)-segmentation method [8]. This is a particular formulation of segmentation seeking to find a segmentation into k segments and an assignment of each segment to one of h labels ($h < k$). Therefore, the (k, h)-segmentation model captures the idea that segments of the same type might re-appear in the sequence.

3 Problem definition

Consider a sequence X of n observations of d-dimensional vectors. I.e., $X = \langle x_1 \dots x_n \rangle$ where $x_i \in \mathbb{R}^d$ for all $i = 1, \dots, n$. In matrix form X contains the observations as its rows, so X is an $n \times d$ matrix. A k-segmentation $\mathcal{S}$ of X is defined by $k + 1$ boundary points $1 = b_1 < b_2 < \dots < b_k < b_{k+1} = n + 1$, yielding segments $\mathcal{S} = \langle S_1 \dots S_k \rangle$, where $S_j = \langle x_{b_j} \ \dots \ x_{b_{j+1}}{-}1 \rangle$. Thus, $\mathcal{S}$ partitions the sequence X into continuous intervals so that each point $x_i \in X$ belongs to exactly one interval. We denote by $j(i) \in \{1, \dots, k\}$ the segment to which point x_i belongs to, i.e., $j(i)$ is the index such that $b_{j(i)} \leq i < b_{j(i)+1}$.

We will now consider basis-vector representations of the data. We denote by $V = \{v_1, \dots, v_m\}$ a set of m basis vectors $v_t \in \mathbb{R}^d$, $t = 1, \dots, m$. The number of basis vectors m is typically significantly smaller than the number n of data points. In matrix form, V is an $m \times d$ matrix containing the basis vectors as its rows.

Along with the basis vectors V, we need for each segment the coefficients of a linear combination of the basis vectors. For each segment S_j we have a set of coefficients $a_{jt} \in \mathbb{R}$, for $t = 1, \dots, m$; in matrix notation, $A = (a_{jt})$ is an $k \times m$ matrix of coefficients. We often indicate the size of a matrix as a subscript: for example, a matrix X of size $n \times d$ is written as $X_{n \times d}$.

A *basis segmentation* consists of a k-segmentation $\mathcal{S} = \langle S_1 \dots S_k \rangle$ of the input sequence X, a set $V = \{v_1, \dots, v_m\}$ of $m < d$ basis vectors, and coefficients $A = (a_{jt})$, for $j = 1, \dots, k$ and $t = 1, \dots, m$ for each segment and basis vector pair. We approximate the sequence with piecewise constant linear combinations of the basis vectors, i.e., all observations in segment S_j are represented by the single vector

$$u_j = \sum_{t=1}^{m} a_{jt} v_t. \tag{3.1}$$

The problem we consider in this paper is the following.

Problem 1. *Given a sequence $X = \langle x_1 \dots x_n \rangle$, and integers k and m, find a basis segmentation $(\mathcal{S}, V, A)$ that uses k segments and a basis of size m, so that the reconstruction error*

$$E(X; \mathcal{S}, V, A) = \sum_{i=1}^{n} ||x_i - u_{j(i)}||^2$$

is minimized. The constant vector $u_{j(i)}$ for approximating segment S_j is given by Equation (3.1).

In other words, the goal is to find a small basis (m vectors) such that the input sequence can be segmented in k segments, and each segment can be described well as a linear combination of the basis vectors.

The difficulty of the basis segmentation problem stems from the interplay between segmentation and dimensionality reduction. While we can segment a sequence optimally in polynomial time, and we can reduce the dimensionality of a sequence optimally in polynomial time, it is not at all clear that the optimal solution can be obtained by combining these two steps that are in themselves optimal.

Whether the basis segmentation problem can be solved in polynomial time, or if it is NP-hard, remains an open problem. However, we can show that a straightforward algorithm that first segments and then performs dimensionality reduction can be no more than a factor of 5 worse than the optimum.

4 The algorithms

We discuss three algorithms for solving the basis segmentation problem, each of which combines segmentation and dimensionality reduction in a different way. Matlab implementations of the methods are available in http://www.cs.helsinki.fi/hiit_bru/software/

4.1 Building blocks. All the algorithms combine dynamic programming for k-segmentation with the PCA technique for dimensionality reduction. We describe briefly these two basic ingredients.

Given a sequence X, the classic algorithm of Bellman [4] finds the optimal k-segmentation of X, in the sense that the error of the original sequence to a piecewise-constant representation with k segments is minimized (in fact, the algorithm can be used to compute optimal k-segmentations with piecewise polynomial models). Bellman's algorithm is based on dynamic programming and the running time is $O(n^2 k)$.

The method is based on computing an $(n \times k)$-size table E_S, where the entry $E_S[i,p]$ denotes the error of segmenting the prefix sequence $\langle x_1, \dots, x_i \rangle$ using p segments. The computation is based on the equation

$$E_S[i,p] = \min_{1 \le j \le i} (E_S[j-1, p-1]) + E[j,i]),$$

where $E[j,i]$ is the error of representing the subsequence $\langle x_j, \dots, x_i \rangle$ with one segment.

The second fundamental data-analysis tool employed by our algorithms is Principal Component Analysis (PCA). Given a matrix $Z_{n \times d}$ with points as rows, the goal is to find a subspace $\mathcal{L}$ of dimension $m < d$ so that the residual error of the points of Z projected onto $\mathcal{L}$ is minimized (or equivalently, the variance of the points of Z projected onto $\mathcal{L}$ is maximized). The PCA algorithm computes a matrix Y of rank m, and the decomposition $Y = AV$ of Y into the orthogonal basis V of size m, such that

$$Y = \underset{\text{rank}(Y') \le m}{\arg\min} \; ||Z - Y'||_F. \tag{4.2}$$

PCA is accomplished by eigenvalue decomposition on the correlation matrix, or by Singular Value Decomposition on the data matrix. The basis vectors $v_1, \dots, v_m$ are the eigenvectors of the correlation matrix or the right singular vectors of the data matrix, respectively.[1]

4.2 Overview of the algorithms. Next we describe our three algorithms. Briefly, they are as follows:

- Seg-PCA: First segment into k segments, then find a basis of size m for the segment means.
- Seg-PCA-DP: First segment into k segments, then find a basis of size m for the segment means, and then refine the segmentation by using the discovered basis.
- PCA-Seg: First do PCA to dimension m on the whole data set. Then obtain the optimal segmentation of the resulting m-dimensional sequence.

[1] The matrix norm $|| \cdot ||_F$ in Equation (4.2) is the *Frobenius norm*, and this is the property that we will need for the rest of our analysis, however, it is worth noting that Equation (4.2) holds for all matrix norms induced by L_p vector norms.

4.3 Algorithm 1: Seg-PCA.

The first algorithm we propose for the basis segmentation problem is called Seg-PCA, and as we will show, it produces a solution that is a factor 5 approximation to the optimal.

The algorithm works as follows: First, the optimal k-segmentation is found for the sequence X in the full d-dimensional space. Thus we obtain segments $\mathcal{S} = (S_1, \ldots, S_k)$ and d-dimensional vectors $u_1, \ldots, u_k$ representing the points in each segment. Then the algorithm considers the set of the k vectors $\{u_1, \ldots, u_k\}$, where each vector u_j is weighted by $|S_j|$, the length of segment S_j. We denote this set of weighted vectors by $U_S = \{(u_1, |S_1|), \ldots, (u_k, |S_k|)\}$. Intuitively, the set U_S is an approximation of the n d-dimensional points of the sequence X with a set of k d-dimensional points. In order to reduce the dimensionality from d to m we perform PCA on the set of weighted points U_S.[2] The PCA computation gives for each segment vector u_j an approximate representation u'_j such that

$$u'_j = \sum_{t=1}^{m} a_{jt} v_t, \quad j = 1, \ldots, k, \tag{4.3}$$

where $\{v_1, \ldots, v_m\}$ constitute a basis, and a_{jt} are real-valued coefficients. The vectors $\{u'_1, \ldots, u'_k\}$ given by Equation (4.3) lie on an m-dimensional space, and the optimality of the weighted PCA guarantees that they minimize the error $\sum_j |S_j| ||u_j - u'_j||^2$ among all possible k vectors lying on an m-dimensional space. The last step of the Seg-PCA algorithm is to assign to each segment S_j the vector u'_j computed by PCA.

As an insight into the Seg-PCA algorithm, note that any data point x_i is approximated by the mean value of the segment j into which it belongs; denoted by $u_{j(i)}$. The mean value $u_{j(i)}$ itself is approximated as a linear combination of basis vectors: $u'_{j(i)} = \sum_{t=1}^{m} a_{j(i)t} v_t$. Thus the error in representing the whole data set is

$$\sum_{i=1}^{n} ||x_i - \sum_{t=1}^{m} a_{j(i)t} v_t||^2$$

Next we show that the Seg-PCA algorithm yields a solution to the basis segmentation problem that is at most factor 5 from the optimal solution.

Theorem 4.1. *Let X be a sequence of n points in $\mathbb{R}^d$ and let k and m be numbers such that $m < k < n$ and $m < d$. If $E(X, k, m)$ is the error achieved by the* Seg-PCA *algorithm and $E^*(X, k, m)$ is the error of the optimal solution to the basis segmentation problem, then $E(X, k, m) \leq 5\, E^*(X, k, m)$.*

Proof. Let $u_1, \ldots, u_k$ be the vectors of the segments $S_1, \ldots, S_k$ found by the optimal segmentation. Then, if $E(X, k) = \sum_{i=1}^{n} ||x_i - u_{j(i)}||^2$, we have $E(X, k) \leq E^*(X, k, m)$. The reason is that for $d > m$ the error of an optimal basis segmentation of size d is at most as large as the error of an optimal basis segmentation of size m. The algorithm performs PCA on the weighted points $U_S = \{u_1 |S_1|, \ldots, u_k |S_k|\}$, and it finds an m-basis decomposition $U' = AV$, such that each segment S_j is represented by the point $u'_j = \sum_{t=1}^{m} a_{jt} v_t$, $j = 1, \ldots, k$. The error of the Seg-PCA algorithm is

$$\begin{aligned}
E(X,k,m) &= \sum_{i=1}^{n} ||x_i - u'_{j(i)}||^2 \\
&= \sum_{j=1}^{k} \sum_{i \in S_j} ||x_i - u'_j||^2 \\
&\overset{(*)}{=} \sum_{j=1}^{k} |S_j|\, ||u_j - u'_j||^2 \\
&\quad + \sum_{j=1}^{k} \sum_{i \in S_j} ||x_i - u_j||^2 \\
&= \sum_{j=1}^{k} |S_j|\, ||u_j - u'_j||^2 + E(X,k) \\
&\leq \sum_{j=1}^{k} |S_j|\, ||u_j - u'_j||^2 + E^*(X,k,m). \qquad (4.4)
\end{aligned}$$

Equality $(*)$ follows from Lemma 4.1 below. We now proceed with bounding the term $\sum_{j=1}^{k} |S_j|\, ||u_j - u'_j||^2$. Notice that the values u'_j are the optimal values found by PCA for the weighted points u_j. Therefore, if u^*_j are the optimal values for the basis segmentation problem, such that u^*_j can be written with an optimal m-basis V^*, then

$$\begin{aligned}
\sum_{j=1}^{k} |S_j|\, ||u_j - u'_j||^2 &\leq \sum_{j=1}^{k} |S_j|\, ||u_j - u^*_j||^2 \\
&\leq 2 \sum_{i=1}^{n} ||x_i - u^*_{j(i)}||^2 \\
&\quad + 2 \sum_{j=1}^{k} \sum_{i \in S_j} ||x_i - u_j||^2 \\
&\leq 2E^*(X,k,m) + 2E(X,k) \\
&\leq 4E^*(X,k,m). \qquad (4.5)
\end{aligned}$$

The first inequality above comes from the optimality of PCA in the Frobenius norm, and the second inequality

[2] Considering PCA with weighted points is by definition equivalent to replicating each point as many times as its corresponding weight. Algorithmically, the weighted version of PCA can be performed equally efficiently without replicating the points.

is the "double triangle inequality" for the square of distances. Finally, by $u^*_{j(i)}$ we denote the point in the set $\{u^*_1, \ldots, u^*_k\}$ that it is the closest to $u_{j(i)}$.

The claim $E(X, k, m) \leq 5E^*(X, k, m)$ follows now from Equations (4.4) and (4.5). □

Lemma 4.1. *Let $\{x_1, \ldots, x_n\}$ be a set of n points in $\mathbb{R}^d$ and let $\bar{x}$ be their coordinate-wise average. Then, for any point $y \in \mathbb{R}^d$, we have*

$$\sum_{i=1}^{n} ||x_i - y||^2 = n \cdot ||\bar{x} - y||^2 + \sum_{i=1}^{n} ||x_i - \bar{x}||^2.$$

Proof. A straightforward computation. □

Note that Lemma 4.1 can be used in the proof of Theorem 4.1 since each vector u_j is the average of all points x_i in the segment S_j.

4.4 Algorithm 2: Seg-PCA-DP**.** Our second algorithm is an extension of the previous one. As previously, we start by obtaining the optimal k-segmentation on the original sequence X, and finding the representative vectors u_j for each segment (the means of the segments). We then perform PCA on the weighted set $\{(u_1, |S_1|) \ldots, (u_k, |S_k|)\}$ and we obtain vectors u'_j that can be expressed with a basis V of size m.

The novel step of the algorithm Seg-PCA-DP is to adjust the segmentation boundaries by taking into account the basis V found by PCA. Such an adjustment can be done by a second application of dynamic programming. The crucial observation is that now we assume that the vector basis V is known. Given V we can evaluate the goodness of representing each subsegment $S_{ab} = \langle x_a \ldots x_b \rangle$ of the sequence X by a single vector u_{ab} on the subspace spanned by V. In particular, the representation error of a segment S is $\sum_{i \in S} ||x_i - u||^2$, where $u = \sum_{t=1}^{m} a_t v_t$ is chosen so that it minimizes $||\mu - u||^2$ where μ is the mean of the segment S. This is equivalent[3] to finding the vector $u = \sum_{t=1}^{m} a_t v_t$ that minimizes directly $\sum_{i \in S} ||x_i - u||^2$. In other words, the representative vector u for a segment S is the projection of the mean of the points in S onto the subspace spanned by V.

Since the representation error of every potential segment can be computed efficiently given the basis V, and since the total error of a segmentation can be decomposed into the errors of its segments, dynamic programming can be applied to compute the optimal k-segmentation given the basis V. Notice that the first two steps of the algorithm are identical to the Seg-PCA algorithm, while the last step can only improve the cost of the solution. Therefore, the same approximation factor of 5 holds also Seg-PCA-DP. One can also iterate the process in a EM fashion: Given a segmentation $\mathcal{S}^{(i)}$, compute a new basis $V^{(i+1)}$, and given the basis $V^{(i+1)}$ compute a new segmentation $\mathcal{S}^{(i+1)}$. The process is repeated until the representation error does not decrease anymore.

4.5 Algorithm 3: PCA-Seg**.** Our last algorithm, PCA-Seg, has the advantage that it does not perform segmentation on a high-dimensional space, which can be a time-consuming process. The algorithm performs an optimal rank-m approximation to the original data, and then it finds the optimal k-segmentation on the rank-reduced data.

First PCA is used to compute $U_{n \times d} = B_{n \times m} V_{m \times d}$ as an optimal rank-m approximation to the data $X_{n \times d}$. Instead of performing segmentation in the original (d-dimensional) space, PCA-Seg algorithm projects the data X onto the subspace spanned by V and then finds the optimal k-segmentation on the projected data.

Since PCA approximates X by U, the projected data can be written as $UV^T = BVV^T = B$, where the last simplification uses the orthogonality of the basis vectors of V. Therefore, the k-segmentation step is performed on the low-dimensional data $B = (b_{it})$, where $i = 1, \ldots, n$ and $t = 1, \ldots, m$. The representation error of a segment S is simply $\sum_{i \in S} ||b_i - \mu_{b_i}||^2$ where μ_{b_i} is the mean of the segment S. The total reconstruction error of algorithm PCA-Seg is

$$\sum_{j=1}^{k} \sum_{i: b_i \in S_j} ||x_i - \sum_{t=1}^{m} a_{S_j t} v_t||^2 = \sum_{i=1}^{n} ||x_i - \sum_{t=1}^{m} a_{j(i)t} v_t||^2$$

where the $a_{S_j t}$'s are chosen so that they minimize $||\mu_{S_j} - \sum_{t=1}^{m} a_{S_j t} v_t||^2$, with μ_{S_j} being the mean of segment S_j.

4.6 Complexity of the algorithms. The components of the algorithms have polynomial complexity. For PCA/SVD, the complexity for n rows with d dimensions is $O(nd^2 + d^3)$. For dynamic programming, the complexity is $O(n^2 k)$. This assumes that the initial costs of describing each of the $O(n^2)$ potential segments by one level can be computed in unit time per segment. Thus the dynamic programming dominates the complexity. For the last algorithm, PCA-Seg, notice that the segmentation on the data $B_{n \times m}$ is faster than the segmentation of the original data $X_{n \times d}$ by a factor of d/m. Therefore, PCA-Seg is a faster algorithm than the other two, however, the empirical results show that in many cases it produces solutions of the poorest quality.

[3] We omit the proof due to space limitations.

d	k	m	s	Seg-PCA	Seg-PCA-DP	PCA-Seg	Seg	(k,h)
10	10	4	0.05	1710	1710	1769	1707	1785
10	10	4	0.1	3741	3741	3780	3727	3829
10	10	4	0.15	6485	6485	6543	6466	6568
10	10	4	0.25	7790	7790	7887	7772	7811
10	5	4	0.05	2603	2603	2648	2603	2603
10	5	4	0.1	4764	4764	4851	4764	4764
10	5	4	0.15	6230	6230	6304	6230	6230
10	5	4	0.25	8286	8286	8330	8286	8287
20	10	5	0.05	7065	7065	7141	7028	7134
20	10	5	0.1	10510	10510	10610	10475	10781
20	10	5	0.15	14520	14520	14646	14486	14640
20	10	5	0.25	17187	17187	17370	17152	17291

Table 1: Reconstruction errors for generated data at true k and m ($n = 1000$). s is the standard deviation of the data. Seg is the error of the k-segmentation and (k,h) is the error of the (k,h) segmentation; no dimensionality reduction takes place in these two methods.

k	m	Seg-PCA	Seg-PCA-DP	PCA-Seg	Seg	(k,h)
10	3	4382	4381	4393	3310	5026
15	3	3091	3074	3078	1750	4153
20	3	2694	2676	2679	1244	3838
10	4	3608	3608	3618	3310	4140
15	4	2255	2245	2248	1750	3231
20	4	1839	1820	1821	1244	2577
10	5	3436	3436	3454	3310	3388

Table 2: Reconstruction errors for the exchange rate data ($n = 2567$ and $d = 12$).

5 Experimental results

In this section we empirically evaluate the three algorithms on generated and real-life data sets. All three algorithms output both segment boundaries and basis vectors, and thus comparing their results is quite straightforward.

In all our figures, the grayscale in each subplot is spread linearly from highest negative (white) to highest positive (black) value, and so blocks with similar shades correspond to their coefficients having similar values.

5.1 Generated data. To study the solutions produced by the different algorithms, we generated artificial data as follows. We first created m random, orthogonal, unit-norm, d-dimensional basis vectors v_t. We then chose the boundaries of k segments at random for a sequence of length $n = 1000$. For each segment $j = 1, \ldots, k$, the basis vectors v_t were mixed with random coefficients c_{jt} to produce the mean $\mu_j = \sum_{t=1}^{m} c_{jt} v_t$ of the segment; data points belonging to the segment were then drawn from a Gaussian distribution $\mathcal{N}(\mu_j, s^2)$. The data was made zero mean and unit variance in each dimension. We used three sets of the parameters (d, k, m) to generate the data: $(10, 10, 4)$, $(10, 5, 4)$, and $(20, 10, 5)$. Furthermore, we varied the standard deviation as $s = 0.05, 0.1, 0.15, 0.25$.

For comparison, we also show results on plain k-segmentation, and on (k,h)-segmentation [8] (recall the description of the latter problem from Section 2). We chose h so that the number of parameters in our model and the data model of (k,h)-segmentation are as close as possible: $h = \lceil m(d+k)/d \rceil$.

Table 1 shows the reconstruction errors of the three algorithms on the generated data, rounded to the nearest integer. We also show the error of plain k-segmentation and (k,h)-segmentation, in which the reconstruction error is measured as the difference between a data point and the segment mean. The plain k-segmentation can be seen as a baseline method since it does not reduce the dimensionality of the data; (k,h)-segmentation poses further constraints on the levels of the segments, giving rise to an increase in the reconstruction error. Algorithm PCA-Seg has a slightly larger error than the other two algorithms, as it operates on a rank-reduced space.

We then compared the estimated segment bound-

k	m	Seg-PCA	Seg-PCA-DP	PCA-Seg	Seg
10	2	6634	6516	6713	5747
10	3	6214	6190	6287	5747
10	4	6001	5998	5985	5747
10	5	5835	5835	5854	5747
15	2	5895	5744	5910	4608
15	3	5374	5330	5368	4608
15	4	4981	4972	4984	4608
15	5	4765	4764	4781	4608
20	2	5489	5309	5349	3828
20	3	4659	4655	4672	3828
20	4	4251	4249	4267	3828
20	5	4020	4018	4031	3828
25	2	5035	4918	4936	3254
25	3	4144	4136	4155	3254
25	4	3714	3711	3728	3254
25	5	3477	3476	3489	3254

Table 3: Reconstruction errors for human chromosome 22 ($n = 697$ and $d = 16$).

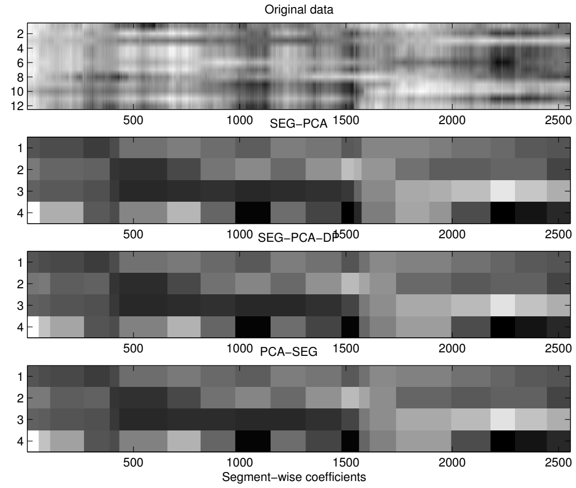

Figure 1: Basis segmentation of data on exchange rates ($n = 2567$, $d = 12$). The segment-wise coefficients a_{jt} for $k = 20$ and $m = 4$, algorithms Seg-PCA, Seg-PCA-DP and PCA-Seg.

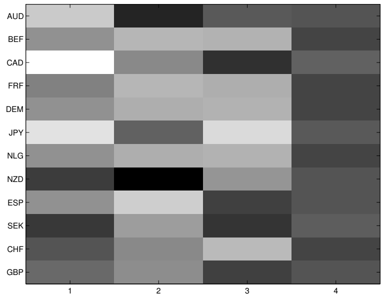

Figure 2: Basis vectors of the exchange rate data at Algorithms Seg-PCA and Seg-PCA-DP at $k = 20, m = 4$.

aries to the true boundaries, by measuring the sum of distances between the boundaries. The boundaries are recovered almost perfectly. The sum of distances between true and estimated boundaries was 10 elements (out of 1000) in the worst case ($d = 20, k = 10, m = 5, s = 0.25$), and 0 to 2 for all datasets with $s < 0.25$.

5.2 Exchange rates. As a first experiment on real data, we applied our algorithms on exchange-rate data on 12 currencies in dollars; the 2567 observations of daily rates are from 1986 to 1996. The 12 currencies are AUD, BEF, CAD, FRF, DEM, JPY, NLG, NZD, ESP, SEK, CHF and GBP. The data were made zero mean and unit variance in each dimension. The data is available at UCI KDD archive [11].

The segmentations found by the three algorithms are shown in Figure 1, where parameter values $k = 20$ and $m = 4$ are chosen as an example. The results are quite similar, and the changes in the original data are captured nicely (remember that white is smallest, and black largest value). The reconstruction errors are shown in Table 2 for $k = 10, 15, 20$ and $m = 3, 4, 5$. Also the reconstruction errors have only small differences.

Figure 2 shows the basis vectors for the case $k = 20$ and $m = 4$ for algorithms Seg-PCA and Seg-PCA-DP, which use the same set of basis vectors. The basis vectors are ordered according to their importance.

k	m	Seg-PCA	Seg-PCA-DP	PCA-Seg	Seg
10	2	3633	3618	3675	3181
10	3	3355	3355	3375	3181
10	4	3221	3221	3246	3181
10	5	3195	3195	3218	3181
15	2	3375	3237	3314	2593
15	3	2937	2867	2882	2593
15	4	2691	2691	2699	2593
15	5	2649	2649	2661	2593
20	2	3196	3023	3092	2205
20	3	2588	2534	2538	2205
20	4	2318	2318	2325	2205
20	5	2271	2271	2284	2205
25	2	3000	2923	2984	1914
25	3	2339	2288	2292	1914
25	4	2054	2051	2056	1914
25	5	1994	1994	1998	1914

Table 4: Reconstruction errors for human chromosome 22 + zebrafish chromosome 25 ($n = 1031$ and $d = 16$).

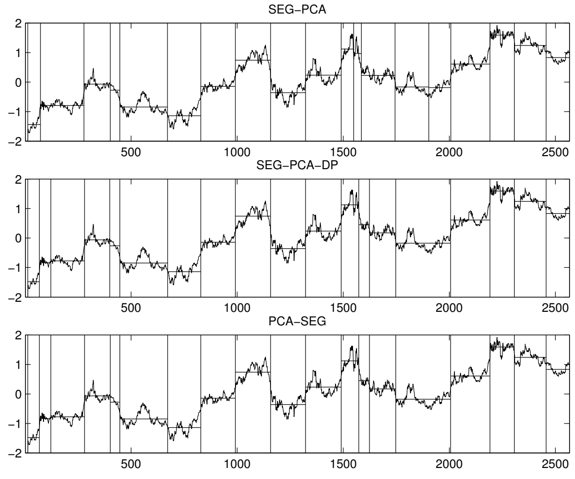

Figure 3: Segmentation on DEM of the exchange rate data ($n = 2567$, $d = 12$) for $k = 20$ and $m = 4$, algorithms Seg-PCA, Seg-PCA-DP and PCA-Seg.

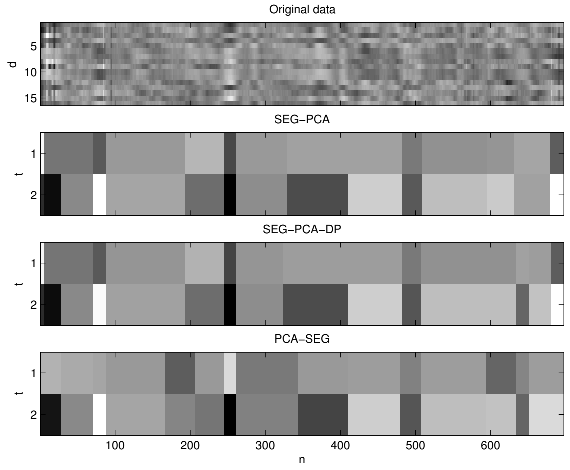

Figure 4: Basis segmentation of data on human chr 22 ($n = 697$, $d = 16$). The segment-wise coefficients a_{jt} for $k = 15$ and $m = 2$, algorithms Seg-PCA, Seg-PCA-DP and PCA-Seg.

The importance of a vector is measured as the amount of variability of the data it accounts for. We see that the first basis vector has a large negative value (light) for AUD, CAD and JPY which thus behave similarly, whereas SEK and NZD (dark) behave in the opposite manner. Similarly, for the second and third basis vectors we can find a few extreme currencies, and the fourth shows only minor distinctions between the currencies. (The basis vectors of algorithm PCA-Seg are quite similar to those in Figure 2.) Combining Figures 1 and 2 we see which currencies have the largest contribution (large a_{jt}) in each segment.

In Figure 3 we show the segmentation for a single currency, DEM, for the different algorithms at $k = 20$ and $m = 4$. Also the constant levels approximating the data at each segment are shown. We see that the piecewise structure of the currency is captured well in the segmentations found by different algorithms. Also the segmentations found by different algorithms are very similar.

chr	k	m	Seg-PCA	Seg-PCA-DP	PCA-Seg	Seg
1	25	3	69251	69232	70201	68240
1	50	2	61929	61429	62960	58467
1	50	3	60736	60367	61092	58467
2	40	3	84640	84610	84667	82676
2	45	3	82443	82390	82429	80180
3	50	3	58024	57378	57389	54815
3	60	3	55361	54880	54963	51990
3	70	3	53235	52817	52915	49705
13	50	2	24229	24059	24123	20799
13	50	3	22580	22450	22766	20799
13	50	4	21788	21774	22651	20799
14	50	2	20483	20154	20174	17846
14	50	3	19549	19362	19441	17846
14	50	4	18915	18795	19263	17846

Table 5: Reconstruction errors for human chromosomes 1 ($n = 2259$), 2 ($n = 2377$), 3 ($n = 1952$), 13 ($n = 956$) and 14 ($n = 882$); the data consist of the frequencies of all 3-letter words, so $d = 64$.

k	Seg-PCA	Seg-PCA-DP	PCA-Seg	Seg	(k, h)
7	9429	9429	9644	9409	9409
10	8545	8542	8764	8454	8454
15	7961	7935	8071	7684	7785

Table 6: Reconstruction errors for the El Niño data with $m = 5$ ($n = 1480$ and $d = 10$).

5.3 Genome sequences. Another application of our method is on mining the structure of genome sequences. The existence of segments in DNA sequences is well documented [16, 19], and discovering a vector basis could shed more light to the composition of the different genomic segments.

To demonstrate the validity of our approach we experimented on small chromosomes of both human (chromosome 22) and zebrafish (chromosome 25). We converted the sequenced regions of these chromosomes into multidimensional time series by counting frequencies of 2-letter words in fixed-sized overlapping windows. Thus the dimension of the resulting time series is $d = 4^2 = 16$. The data were normalized to have zero mean and unit variance in each dimension separately. We used window size of 500 kilobase pairs (Kbp), overlapping by 50 Kbp, which resulted in 697 and 334 windows for the human and zebrafish chromosomes, respectively.

The resulting 16-dimensional time series, as well as the results of the different algorithms on the human chromosome are shown in Figure 4 for $k = 15$ and $m = 2$. The corresponding basis vectors are shown in Figure 6. All the algorithms find segmentations with similar boundaries and similar basis vector coefficients. For instance, by observing the basis vectors one notes that the words 'AA', 'AT', 'TA', 'TT' are described by a similar high positive value in the second basis vector. The algorithms Seg-PCA and Seg-PCA-DP produce a basis vector with high values for 'AC' and 'CG', while PCA-Seg produces a vector with high values for 'GT' and 'TG'.

We also experimented on a concatenation of the human and zebrafish chromosomes. The idea is to check if the algorithms can discover the point where the two chromosomes were concatenated. The results are shown in Figures 5 and 7, for the same parameters as before: $k = 15$ and $m = 2$. The boundary between the species is at position 697, and all algorithms produce a boundary point close to this position. The coefficients are almost identical for all algorithms. The first vector is dominated by the value of 'CG'. The second vector is predominately used in the zebrafish section of the sequence. It is also notable that the zebrafish section of the data is described by the algorithms to be very different from the rest of the sequence. In this experiment, the algorithms discover the "true" segment boundaries, and give some insight into which dimensions behave similarly in a segment.

Reconstruction errors on the sequences are displayed in Tables 3 and 4 for various values of the parameters k and m. In one case algorithm PCA-Seg is the best (human chromosome 22, $k = 10$, $m = 4$),

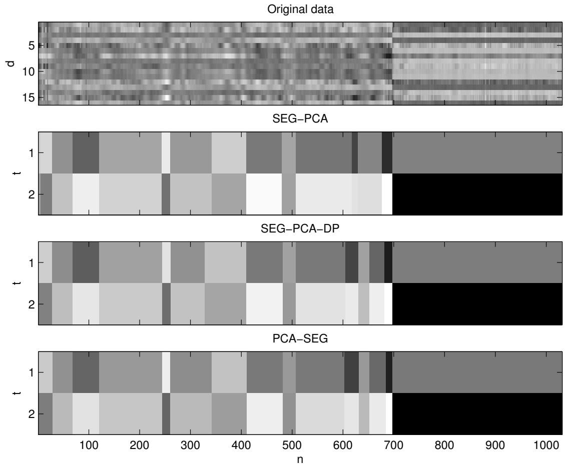

Figure 5: Basis segmentation of data on human chr 22 + zebrafish chr 25 ($n = 1031$, $d = 16$). The segment-wise coefficients a_{jt} for $k = 15$ and $m = 2$, algorithms SEG-PCA, SEG-PCA-DP and PCA-SEG. The species boundary is located at position 697.

but in most cases it has the largest error. Algorithm SEG-PCA is in general very close to SEG-PCA-DP.

Finally, we experimented with human chromosomes 1, 2, 3, 13 and 14. We divided the sequenced regions of these chromosomes into nonoverlapping 100 Kbp windows, making the number of data points much larger than in the previous cases. In this experiment we count the frequencies of 3-letter words in each window, producing a time series of dimension $d = 4^3 = 64$; recall that in the previous experiments we used 2-letter words resulting in 16-dimensional data. Reconstruction errors for the long sequences of 3-letter words are displayed in Table 5. Algorithm SEG-PCA-DP is always the best of the three algorithms, but the differences are usually not very large. One noteworthy feature is that $m = 3$ or $m = 4$ is sufficient to get reconstruction errors very close to the segmentation error in the full-dimensional space.

5.4 El Niño data. We also experimented on data on El Niño [11] that contains oceanographic and surface meteorological readings taken from a series of buoys positioned throughout the equatorial Pacific. We selected 2 buoys having 1480 common dates of measurements, and constructed 10-dimensional data of zonal winds, meridional winds, relative humidity, air temperature, and sea surface temperature of the 2 buoys. We selected the number of basis vectors as $m = 5$, corresponding to the number of significant eigenvalues in the covariance matrix of the data. The data was made zero

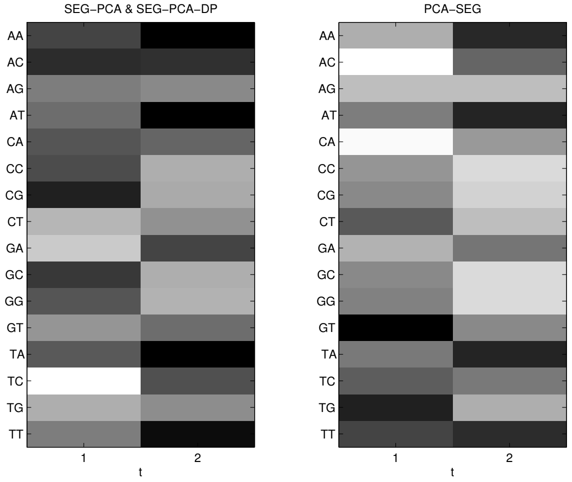

Figure 6: Basis vectors, human chr 22 ($k = 15$, $m = 2$).

mean and unit variance in each dimension separately. Table 6 shows the reconstruction errors for $k = 7$, 10 and 15 segments.

Naturally, the error decreases as k increases. It is evident that the reconstruction error of plain k-segmentation is quite close to that of algorithms SEG-PCA and SEG-PCA-DP, meaning that the dimensionality reduction from 10 to 5 does not hurt the accuracy much.

Figure 8 shows the original data together with the segment-wise coefficients a_{jt} as grayscale images, for each of the three algorithms SEG-PCA, SEG-PCA-DPand PCA-SEG for $k = 7, 10, 15$. Looking at the original data suggests that the block structure found makes sense. The segment boundaries are seen as the change points of the coefficient values; there are slight differences between the boundaries of the three algorithms.

To help interpreting the results of Figure 8, the basis vectors are shown in Figure 9 for $k = 15$; the cases at other k are quite similar. One notes that the zonal and meridional winds often behave similarly in PCA (dimensions 1 to 2, and 6 to 7), as do the air and sea surface temperatures (dimensions 4 to 5, and 9 to 10); this is not very surprising, considering the nature of the data. Again, we can identify a few significant features in each basis vector as the ones having a very light or very dark value in Figure 9.

6 Conclusions

We have introduced the basis segmentation problem, given three algorithms for it, and presented empirical results on their performance. The results show that the

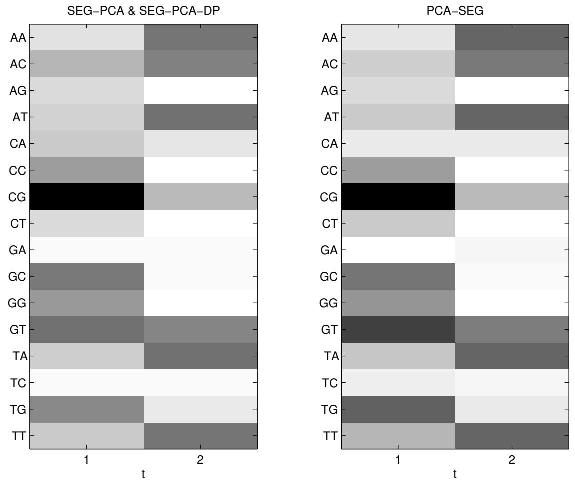

Figure 7: Basis vectors, human chr 22 + zebrafish chr 25 ($k = 15$, $m = 2$).

Seg-PCA and Seg-PCA-DP algorithms perform very well in practice, finding the true generating model in simulated data, and almost always yielding the smallest errors on real data. These observations are supported by the factor 5 approximability result. On the other hand, algorithm PCA-Seg in general produces results not too far from the other two algorithms, and is faster by a factor of d/m.

We have also discussed experimental results on real data that demonstrate the possible applications of our approach in sequence analysis; the general methodology and problem setting can be applied in analyzing multidimensional time series, such as in our examples with exchange-rate data, or in bioinformatics in discovering hidden structure in genomic sequences. A fascinating future theme would be to look for possibilities of online segmentation and dimensionality reduction, in the spirit of [14, 9].

Several open problems still remain with respect to the basis segmentation problem. First, what is the computational complexity of the problem? The components of the problem, segmentation and dimensionality reduction, are polynomial in at least some forms, but it is not clear whether this translates to a polynomial-time solution. No very simple reduction for proving NP-hardness seems apparent.

The practical applications of the basis segmentation algorithms are also interesting: how strong is the latent structure in real data, i.e., do the different segments in some cases really have an underlying common basis? Our experiments point to this direction.

References

[1] H. Attias. Independent factor analysis with temporally structured sources. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000.

[2] R. K. Azad, J. S. Rao, W. Li, and R. Ramaswamy. Simplifying the mosaic description of DNA sequences. *Physical Review E, 66, article 031913*, 2002.

[3] Y. Azar, A. Fiat, A. R. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *STOC*, 2001.

[4] R. Bellman. On the approximation of curves by line segments using dynamic programming. *Communications of the ACM*, 4(6), 1961.

[5] E. Bingham, A. Kabán, and M. Girolami. Topic identification in dynamical text by complexity pursuit. *Neural Processing Letters*, 17(1):69–83, 2003.

[6] E. G. Carlstein, D. Siegmund, and H.-G. Muller. *Change-Point Problems*. IMS, 1994.

[7] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[8] A. Gionis and H. Mannila. Finding recurrent sources in sequences. In *ReCombB*, 2003.

[9] S. Guha, D. Gunopulos, and N. Koudas. Correlating synchronous and asynchronous data streams. In *SIGKDD*, 2003.

[10] S. Guha, N. Koudas, and K. Shim. Data-streams and histograms. In *STOC*, pages 471–475, 2001.

[11] S. Hettich and S. D. Bay. The UCI KDD Archive [http://kdd.ics.uci.edu], 1999. UC, Irvine.

[12] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmäki, and H. T. Toivonen. Time series segmentation for context recognition in mobile devices. In *ICDM*, 2001.

[13] E. Keogh, K. Chakrabarti, S. Mehrotra, and M. Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. In *SIGMOD*, 2001.

[14] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *IEEE International Conference on Data Mining*, 2001.

[15] M. Koivisto et al. An MDL method for finding haplotype blocks and for estimating the strength of haplotype block boundaries. In *Pacific Symposium on Biocomputing*, 2003.

[16] W. Li. DNA segmentation as a model selection process. In *RECOMB 2001*, pages 204–210, 2001.

[17] J. S. Liu and C. E. Lawrence. Bayesian inference on biopolymer models. *Bioinformatics*, 15(1):38–52, 1999.

[18] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent Semantic Indexing: A Probabilistic Analysis. In *PODS*, 1998.

[19] A. Pavlicek, J. Paces, O. Clay, and G. Bernardi. A compact view of isochores in the draft human genome sequence. *FEBS Letters*, 511:165–169, 2002.

[20] V. Ramensky, V. Makeev, M. Roytberg, and V. Tumanyan. DNA segmentation through the Bayesian approach. *Journal of Computational Biology*, 2000.

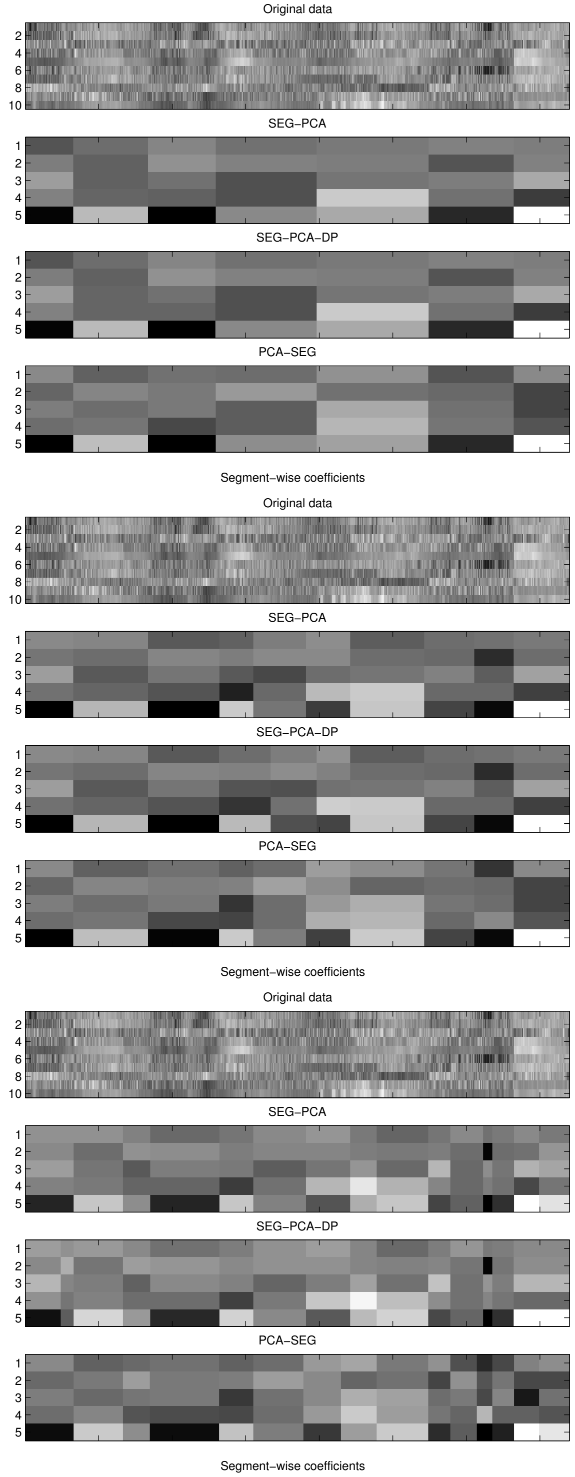

Figure 8: Basis segmentation of data on El Niño ($n = 1480$, $d = 10$). The segment-wise coefficients a_{jt} for $m = 5$, algorithms Seg-PCA, Seg-PCA-DP and PCA-Seg. Top: $k = 7$, middle: $k = 10$, bottom: $k = 15$.

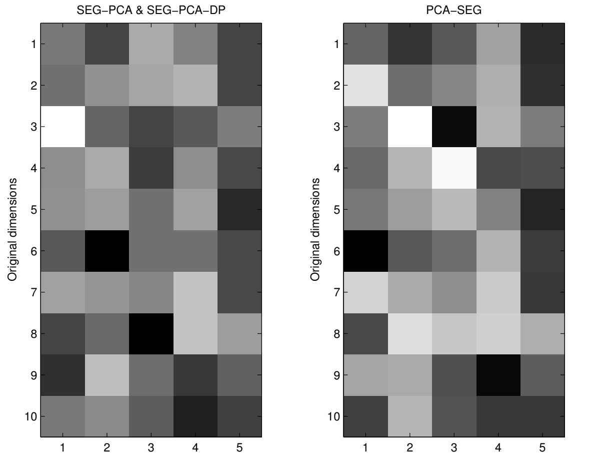

Figure 9: Basis vectors of Algorithms Seg-PCA and Seg-PCA-DP (left) and PCA-Seg (right) on the El Niño data for $k = 15$, $m = 5$. Vertical axis: Dimensions 1 to 5 give the zonal wind, meridional wind, relative humidity, air temperature, and sea surface temperature of one buoy, and dimensions 6 to 10 of another buoy respectively.