Discovery of common subsequences in cognitive evoked potentials

Arthur Flexer[1] and Herbert Bauer[2]

[1] The Austrian Research Institute for Artificial Intelligence
Schottengasse 3, A-1010 Vienna, Austria
arthur@ai.univie.ac.at
[2] Department of Psychology, University of Vienna
Liebiggasse 5, A-1010 Vienna, Austria

Abstract. This work is about developing a new method for the analysis of evoked potentials of cognitive activities that combines methods from statistics and sequence alignment to tackle the following two problems: the visualization of high dimensional sequential data and the unsupervised discovery of patterns within this multivariate set of real valued time series data. The sequences of the original high dimensional vectors are transformed to discrete sequences by vector quantization plus Sammon mapping of the codebook. Instead of having to conduct a time-consuming search for common subsequences in the set of multivariate sequential data a multiple sequence alignment procedure can be applied to the set of one-dimensional discrete symbolic time series. The methods are described in detail and the results are shown to be significantly better than those obtained for two sets of randomized artificial data.

1 Introduction

This work is part of the development of a new method for the analysis of evoked potentials (EP) of cognitive activities that combines methods from statistics and sequence alignment to tackle the following two problems: the visualization of high dimensional sequential data and the unsupervised discovery of patterns within this multivariate real valued set of time series data.

A cognitive EP is the electro cortical potential measurable in the electro encephalogram (EEG) during a cognitive task (spatial imagination). Since an EP is measured via a number of electrodes it is a multi dimensional time series. Our approach to visualize this high dimensional sequential set of data is to replace the sequence of the original vectors by a sequence of prototypical codebook vectors obtained from a clustering procedure. Additionally, a dimensionality reduction technique is applied to obtain an ordered one-dimensional representation of the high dimensional codebook vectors that allows for the depiction of the original sequence as a one-dimensional time series.

The fact that cognitive activities do not elicit one specific EP waveform time locked to the onset of the recording prohibits the conventional approach of simply computing the arithmetic average of all EPs at each sample point. Only

subsequences of the whole EPs that do not occur at fixed time after the onset of the recording can be expected to be due to the cognitive task. Searching for such subsequences in the set of real valued multivariate sequential data is computationally prohibitive. Instead we can use the set of univariate discrete time series, the trajectories across codebook vectors, and apply a multiple sequence alignment procedure [1] that has originally been designed for molecular biology.

Other already existing data mining approaches to processing of sequential patterns are not applicable to our problem for the following reasons: Template based approaches require a query pattern [6] or frequent episode [8] to be defined before the search is started. No such sequential patterns can be formulated for cognitive EPs since only very vague knowledge about the subsequences to be discovered exists. Additionally, these template based approaches are designed for univariate and often symbolic sequences. [7] describes a method to cluster whole sequences of complex composite objects which is not suited for finding subsequences of multivariate real valued data.

Our work is structured in the following way: First we describe the EP data plus two sets of artificial data, which we need to ensure statistical significance of our results. Then all applied methods (clustering, dimensionality reduction, sequence comparison) are presented and optimal parameters are found via computer experiments. Finally, the results of the completed algorithm are described. The results are confirmed via an analysis of variance and discussed.

2 The data

The data stems from 21 EP trials from one person recorded during a spatial imagination task. After appropriate preprocessing (essentially limiting the signals to frequencies below $8Hz$ and eliminating the DC-like trend by subtracting a linear fit), each EP trial consists of 2125 samples, each being a 22 dimensional real valued vector. One complete 22-channel EP trial (duration is 8.5 seconds) is depicted in Fig. 1(a).

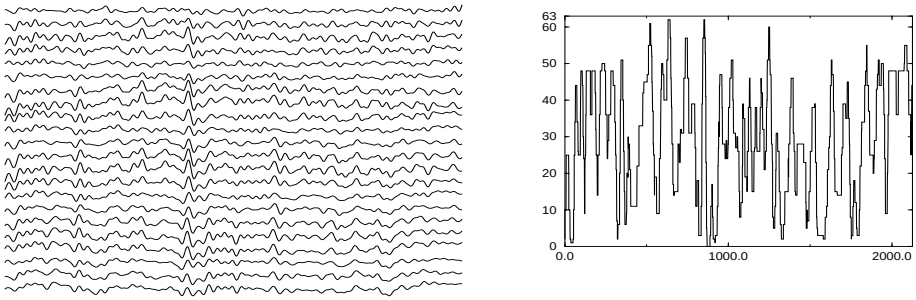
To verify that our procedure yields different results for real human EEG and for unstructured random input and thereby ensuring statistical significance, we produced two kinds of artificial data sets. To produce *time-shuffled EEG* we took a concatenation of the 21 unfiltered EP trials of length 2125 described above and submitted it to a random permutation of the samples in time. 21×2125 times the positions of 2 of the 21×2125 samples were exchanged. The procedure produced 21 EP sequences with destroyed temporal but intact spatial structure. After preprocessing identical to that applied to the real EPs, the time-shuffled EEG data was divided into 21 sequences of length 2125. It should be noted that this procedure also changes the frequency properties of the EPs, i.e. it intensifies higher frequencies and diminishes lower ones, which makes time-shuffled EEG look more random.

To have another set of data that resembles the frequency spectrum of the real EP data more closely, *random Gaussian EEG* was produced by computing 21 sequences of dimension 22 and length 2125 of random Gaussian data and

subjecting it to an FFT (Fast Fourier Transform). The power spectrum of the Gaussian random data was changed so that it resembled the characteristics of the real EEG instead of those of white noise Gaussian random data by directly changing the real and imaginary parts of the FFT-ed signal appropriately and then retransforming it back to the time domain. In doing so we were able to produce artificial EPs which showed the same limitation to frequencies below $8Hz$ with an emphasis of very slow parts around $1Hz$ as real EPs and which are therefore hard to distingiush from real EPs even by a human expert.

To compensate for different amplitude ranges across channels and between real and artificial EEG, all signals in all the dimensions are being standardized separately to zero mean and unit variance.

3 Dimensionality reduction and visualization



(a) (b)

Fig. 1. (a) Example of a complete 8.5 second 22-channel EP recording. (b) The corresponding trajectory across codebook vectors depicted as ordered codebook numbers (y-axis) as a function of time (x-axis).

The EP time series are vector quantized together by using all the EP vectors at all the sample points as input vectors to a clustering algorithm disregarding their ordering in time. Then the sequence of the original vectors x is replaced by the sequence of the prototypical codebook vectors $\hat{x}$. There is a double benefit of this step: it is part of the visualization scheme and the sequences of $\hat{x}$ serve as input to the sequence alignment procedure.

K-means clustering (see e.g. [3, p.201]) is used for vector quantization using the sum of squared differences $d(x,\hat{x}) = \sum_{i=0}^{k-1} \mid x_i - \hat{x}_i \mid^2$ as measure of distance, where both x and $\hat{x}$ are of dimension k. Since observation of the sum of distances $d(x,\hat{x})$ with growing size of codebooks did not indicate an optimal codebook size, we pragmatically decided to use 64 codebook vectors based on the following consideration: We vector quantized the EP data with increasing numbers of codebook vectors. We then took the original EPs and substituted

at each sample point the original real valued vector x_l with the appropriate codebook vector $\hat{x}_i$ (i.e. where $d(x_l, \hat{x}_i) < d(x_l, \hat{x}_s)$ for all s). We then visually inspected both the original EP time series and the coarser codebook time series as series of topographical patterns (surface plots of the 22 values at a single point in time) and checked, whether the important features of the topographies still were existent in the coarser codebook approximation. "Important" features are positive and negative peaks and their development in time. Instead of a set of 22-dimensional time series, we now have sequences of discrete symbols, where each symbol is drawn from the alphabet of the 64 codebook vectors $\hat{x}$.

The sequences of codebook vectors can be visualized in a graph where the x-axis stands for time and the y-axis for the number of the codebook vector. Since in the course of time, the trajectory moves only between codebook vectors that are close to each other in the 22 dimensional vector space, this neighbourhood should also be reflected by an appropriate ordering of the numbers of the codebook vectors. Such an ordered numbering results in smooth curves of the time vs. codebook number graphs and enables visual inspection of similarities between trajectories. Algorithms for finding such ordered lower dimensional representations are, amongst others, various forms of multidimensional scaling (MDS).

We performed a Sammon [10] mapping of the 22-dimensional codebook vectors to one output dimension. Sammon mapping is doing MDS by minimizing the following via steepest descent:

$$\frac{1}{\sum_{i=0}^{c-1}\sum_{j<i} d(\hat{x}_i, \hat{x_j})} \sum_{i=0}^{c-1}\sum_{j<i} \frac{\left(d(\hat{x}_i, \hat{x}_j) - \dot{d}(y_i, y_j)\right)^2}{d(\hat{x}_i, \hat{x_j})} \qquad (1)$$

The $\hat{x}$ are the c=64 codebook vectors and the y are their lower dimensional representations. The distance $\dot{d}(y_i, y_j)$ is the distance in the one-dimensional output space that corresponds to the distance $d(\hat{x}_i, \hat{x}_j)$ in the 22-dimensional input space. This combined technique of K-means clustering plus Sammon mapping of the codebook is described in [4]. An example for a trajectory across an ordered set of codebook vectors is given in Fig. 1(b). Note that the ordering of the numbers of the codebook vectors is needed only for visualization and is not necessary for the subsequent sequence alignment.

4 Unsupervised discovery of patterns in sequences

4.1 Computation of distance matrix

We have 21 sequences made of 64 different items (corresponding to 64 codebook vectors) of length 2125, each for the real, the time-shuffled and the artificial EP data. When running the sequence alignment algorithm, we will need distances between single elements of our sequences. For the 64 codebook vectors, we can calculate a 64×64 distance matrix D which serves as a lookup table for these comparisons. This avoids the repeated computation of distances between the original multivariate vectors which would be computationally intractable.

When running the k-means clustering algorithm as well as when computing trajectories across clusters centers, we used the sum of squared differences $d(x, \hat{x})$ as a measure of distance between an input vector x and a codebook vector $\hat{x}$. By using a codebook vector $\hat{x}(S_i)$ as representative of its partition S_i ($S_i = \{x_l | d(x_l, \hat{x}_i) < d(x_l, \hat{x}_s)$ for all $s\}$), the mean of all $\mid x_l \in S_i \mid$ points of S_i is chosen as representative of S_i. We decided to use the more accurate average of all $n_i \times n_j$ possible distances, $d'(S_i, S_j) = \frac{1}{n_i n_j} \sum_{x_l \in S_i, x_m \in S_j} d(x_l, x_m)$, for computation of the distance matrix D. This measure of distance additionally takes into account the variances of the partitions. Contrary to the earlier tasks, the singular computation of the 64×64 distance matrix allows using the computationally more expensive distance function.

4.2 Fixed segment algorithm

We chose a so-called *fixed length segment* approach for sequence comparison. Given two sequences A and B of length m and n, all possible overlapping segments having a particular window length W from A are compared to all segments from B. This requires of the order $m \times n$ comparisons to be made. For each pair of elements the score taken from our 64×64 distance matrix D is recorded and summed up for the segmental comparison. The distance between two segments of length W from two sequences A and B is therefore:

$$D_{align}(s_a, s_b, W) = \sum_{i=0}^{W-1} d'(A_{s_a+i}, B_{s_b+i}) \tag{2}$$

The indices s_a and s_b are the starting points of the segments in the sequences A and B and A_{s_a+i} and B_{s_b+i} are the codebook vectors or the corresponding partitions respectively. Successive application of this pairwise methods allows for the alignment of more than two sequences. Such a *fixed segment* approach that is explicitly designed for *multiple sequence alignment* is given by [1]. It computes a multiple alignment by iteratively comparing sequences to the multiple alignment obtained so far, keeping always just the L best segments as an intermediate result. The succession of sequences is chosen at random. It starts with first evaluating all possible segment pairs of the first two sequences, keeping the best L of them. The intermediate L two-way alignment segments are compared against all segments of the third sequence, again keeping only the L best three-way alignments (each is a set of three segments, every segment from one of the sequences). This procedure continues until all sequences have been aligned. When a segment is compared to an intermediate "more"-way (let us say p-way) segment, the resulting score is computed as the sum of the p pairwise comparisons of the segments in the intermediate solution with the new segment that is to be aligned. The number of all such crosswise comparisons within the final overall alignment is given by $P = (1 + 2 + \ldots + (p-1)) = \sum_{i=1}^{p-1} i$. The number of all element-wise comparisons within the final overall alignment is given by WP, and its average per element, the average element-wise within alignment distance, by:

$$\bar{D}_{align} = \frac{1}{WP} \sum_{i=1}^{p-1} \sum_{j=i+1}^{p-1} D_{align}(s_i, s_j, W) \quad (3)$$

Desired is a set of starting points s_i for which $\bar{D}_{align}$ is minimal. This approach guarantees that the obtained multiple alignments contain segments that are subsequences of all the original sequences. The number of single element-wise comparisons is $LW(m+1-W)P$, where m=2125, the length of the p=21 sequences. For a given L and m, this function is proportional to p^2, in contrast with m^p comparisons in "brute force" searching where not just the L best but all possible alignments are considered.

4.3 Finding optimal parameters

The number of alignments, L, that are being kept as intermediate results during sequence comparison, has to be sufficiently large to avoid the omission of good alignments that are weak in the first few sequences but strong in the later ones. It should also be small enough to allow for tractable computation time.

The following experiments have been conducted with the real EP data: The window length W was set to 125 (half a second of EP), L was varied from 100 to 1000 and 10000. For each of these three settings, five runs of the fixed segment algorithm have been computed. To compare the results, the average element-wise within alignment distances $\bar{D}_{align}$ (see Equ. 3) of the best scoring alignments for each of fifteen runs were computed. The results are (mean $\pm$ std.dev.) $3.99 \pm .033$ for L=100, $3.97 \pm .017$ for L=1000 and $3.96 \pm .022$ for L=10000. The mean of the distribution of values of the distance matrix D is 6.67 ± 2.78. Comparing this to the results of our experiment, it can be seen that keeping 1000 or even 10000 alignments yields only an insignificant improvement above storing just 100.

The window length W should not be too short, since EP subsequences shorter than 0.25 seconds are of little significance in terms of their psychophysiological interpretation. If the window is made too big, only poorly matched segments can be found, since so-called gaps are not allowed (see Sec. 6).

The following experiments have been conducted with the real EP data: The number of best scoring elements that are being kept, L, was set to 1000, the window length W was varied from 31 to 62, 125 and 187 (corresponding to $0.125, 0.25, 0.5$ and 0.75 seconds). For each of the four settings, five runs of the fixed segment algorithm have been computed. To compare the different results, again the average element-wise within alignment distance $\bar{D}_{align}$ (see Equ. 3) of the best scoring alignments for each of the twenty runs were computed. The results are (mean $\pm$ std.dev.) $3.10 \pm .035$ for W=31, $3.46 \pm .012$ for W=62, $3.97 \pm .020$ for W=125 and $4.28 \pm .041$ for W=187.

If all of our sequences really did contain very similar subsegments of a certain length w, the following behaviour could be expected for values of $\bar{D}_{align}$: For $W \leq w$, $\bar{D}_{align}$ would always be approximately at the same low level (indicating good alignments), since subsegments shorter than w should be detected with the same high performance as subsegments of the full length w. $\bar{D}_{align}$ should

increase steadily for $W > w$ because ever longer parts of subsequences which are not similar across all sequences become part of the alignment. Computations with sufficiently high values W should asymptotically reach the average of the distance array (6.67 in the case of our real EP data).

The $\bar{D}_{align}$ values of our experiments do not show the behaviour described above. Instead of staying at a low level beginning at the smallest window length W=31, they rather increase at a steady pace from W=31 to W=187. We decided to use a window length of W=125 which is sufficiently long enough to allow for psychophysiological interpretation of the results but short enough to yield alignments of still satisfactory quality.

5 Results and comparison with artificial EPs

After having found optimal parameters, the following experiments have been conducted with our real EP data as well as with the time-shuffled and random Gaussian data. The number of best scoring elements that are being kept, L, was set to 100, the window length W to 125 (half a second of EP). For each of the three settings, five runs of the fixed segment algorithm have been computed. To compare the results for real and artificial EPs, the average element-wise within alignment distance $\bar{D}_{align}$ (see Equ. 3) of the best scoring alignments for each of the fifteen runs has been computed. The means and standard deviations for each of the three settings are given in Tab. 1.

type of EP	real EP	time-shuffled EP	random Gaussian EP
mean $\pm$ std.dev.	$3.97 \pm .020$	$4.38 \pm .010$	$6.33 \pm .017$

Table 1. Results of the 3×5 experiments to compare real and artificial EEG.

A one-way analysis of variance (ANOVA) for the variable "type of EP" (real EP, time-shuffled EP, random Gaussian EP) yielded an value F=29070 > 18=F_{99}(df=2 and 4). This indicates that the null hypothesis that the means of the three EP groups are equal is extremely unlikely, in fact its probability is very close to zero. Additional Duncan t-Tests allow us to rank the result for real EP as being significantly better than the result for time-shuffled EP, which is again significantly better than the result for random Gaussian EP, both with a probability of 99%. In Fig. 2, an alignment plus its average is depicted as a trajectory across ordered codebook centers. Since an average alignment corresponds to a time series of 22-dimensional codebook vectors, it can be depicted as a series of topographical patterns (surface plots of the 22 values at a single point in time) and thereby transformed back to its multivariate real valued representation. This series of topographical patterns is now accessible to interpretation for the psychophysiologist.

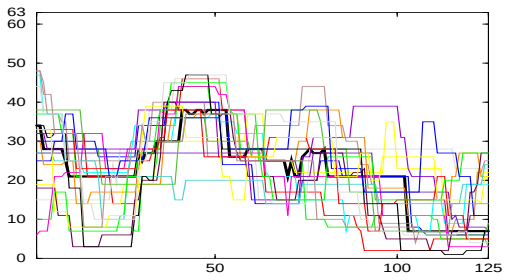


Fig. 2. An example of a best scoring alignment plus its average (the thick line).

6 Discussion

The analysis of cognitive evoked potentials is a largely unsolved problem in psychophysiological research. Classical methods are designed for univariate time series of simpler motoric or sensoric EPs only and can therefore not really cope with the harder problem of analysing cognitive EPs. Nevertheless they are still state of the art.

The transformation of the multivariate real valued EPs to sequences of discrete symbols achieved by using vector quantization plus a dimensionality reduction technique makes a wealth of algorithms for sequence alignment applicable to our problem. This enables psychophysiologists to look at their cognitive EP data by discovering subsequences of fixed length that are, with a certain variance, similar across all EP trials in their multivariate timely appearance.

One of the issues that need some further investigation is the self-consistency of the algorithm. Since our method of sequence comparison is a stochastic algorithm, repeated runs on the same set of data do not necessarily produce the same output alignment. Although the very similar values for $\bar{D}_{align}$ in Tab. 1 for each of the three sets of five runs seem to indicate similar solutions, this still needs some thorough consideration.

If the variance of the obtained alignments would be recognized as being too large for a successful psychophysiological interpretation, this could be due to the fact that so-called gaps are not allowed. When discrete symbolic sequences are compared element-wise three basic things can happen (see e.g. [11]): a *match* if two elements are identical, a *substitution* if they are different and so-called *deletions* or *insertions*, where an element in one subsequence is too different too match or substitute it and it is therefore deleted (inserted if you see it the other way round) which results in a gap in one of the subsequences. Future work could aim for subsequences showing considerable variation on the time axis, therefore needing algorithms that are able to deal with gaps. Since only subsequences and not the whole sequences are to be aligned, a so-called *local multiple sequence alignment* method is needed. Existing global approaches (see e.g. [2] or [12])

would have to be extended to the local case. Hidden Markov Models (see e.g. [9]) are another promising candidate to solve this problem.

Our general approach to the visualization of high dimensional sequential data and the unsupervised discovery of patterns within multivariate sets of time series data is of course not restricted to the problem presented in this work. The methods described can either be applied to multivariate real valued data by using the full approach including the transformation to sequences of discrete symbols through vector quantization plus Sammon mapping or, if already symbolic sequences are available, the fixed segment algorithm alone can be applied.

Acknowledgements: The EEG recordings have been made by R. Gstättner, Dept. of Psychology, University of Vienna. Parts of this work were done within the BIOMED-2 BMH4-CT97-2040 project SIESTA, funded by the EC DG XII. The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Science and Transport. The author was supported by a doctoral grant of the Austrian Academy of Sciences.

References

1. Bacon D.J., Anderson W.F.: Multiple Sequence Alignment, Journal of Molecular Biology, 191, 153-161, 1986.
2. Barton G.J.: Protein Multiple Sequence Alignment and Flexible Pattern Matching, Methods in Enzymology, Vol. 183, pp.403-428, 1990.
3. Duda R.O., Hart P.E.: Pattern Classification and Scene Analysis, John Wiley & Sons, N.Y., 1973.
4. Flexer A.: Limitations of Self-Organizing Maps for Vector Quantization and Multidimensional Scaling, in Mozer M.C., et al.(eds.), Advances in Neural Information Processing Systems 9, MIT Press/Bradford Books, pp.445-451, 1997.
5. Heckerman D., Mannila H., Pregibon D., Uthurusamy R.(eds.): KDD-97: Proceedings Third International Conference on Knowledge Discovery & Data Mining, AAAI Press, Menlo Park, 1997.
6. Keogh E., Smyth P.: A Probabilistic Approach to Fast Pattern Matching in Time Series Databases, in [5], pp.24-30.
7. Ketterlin A.: Clustering Sequences of Complex Objects, in [5], pp.215-218, 1997.
8. Mannila H., Toivonen H., Verkamo A.I.: Discovery of Frequent Episodes in Event Sequences, Data Mining and Knowledge Discovery, Volume 1, Issue 3, 1997.
9. Rabiner L.R., Juang B.H.: An Introduction To Hidden Markov Models, IEEE ASSP Magazine, 3(1):4-16, 1986.
10. Sammon J.W.: A Nonlinear Mapping for Data Structure Analysis, IEEE Transactions on Comp., Vol. C-18, No. 5, p.401-409, 1969.
11. Sankoff D., Kruskal J.B.(eds.): Time warps, string edits, and macromolecules: the theory and practice of sequence comparison, Addison-Wesley, Reading, MA, 1983.
12. Taylor W.R.: Multiple Protein Sequence Alignment: Algorithms and Gap Insertion, Methods in Enzymology, Vol. 266, pp.343-367, 1996.