Determining Subunits for Sign Language Recognition by Evolutionary Cluster-Based Segmentation of Time Series

Mariusz Oszust and Marian Wysocki

Rzeszow University of Technology
Department of Computer and Control Engineering
W. Pola 2, 35-959 Rzeszow, Poland
{moszust, mwysocki}@prz-rzeszow.pl

Abstract. The paper considers partitioning time series into subsequences which form homogeneous groups. To determine the cut points an evolutionary optimization procedure based on multicriteria quality assessment of the resulting clusters is applied. The problem is motivated by automatic recognition of signed expressions, based on modeling gestures with subunits, which is similar to modeling speech by means of phonemes. In the paper the problem is formulated, its solution method is proposed and experimentally verified.

Keywords: time series segmentation, multiobjective clustering, evolutionary optimization, sign language recognition

1 Introduction

Automatic sign language recognition is an important prospective application of gesture-based human-computer interfaces. The aim of the research is a system that properly interprets gestures, e.g. translates them into written or spoken language. Most of such systems described in the literature (see e.g. [1], [2]) are based on word models where one sign represents one model in the model database. They can achieve good performance only with small vocabularies or gesture data sets. The training corpus and the training complexity increase with vocabulary size. So, large-vocabulary systems require the modeling of signed expressions in smaller units than words i.e. the words are modeled with subunits, which is similar to modeling speech by means of phonemes. The main advantage

Please cite this paper as follows: Oszust M., Wysocki M.: Determining Subunits for Sign Language Recognition by Evolutionary Cluster–Based Segmentation of Time Series, Rutkowski L. et al. (Eds.): Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, Springer–Verlag Berlin / Heidelberg pp. 189–196, 2010. The final publication is available at `http://link.springer.com/chapter/10.1007%2F978-3-642-13232-2_23` of this approach is that an enlargement of the vocabulary can be achieved by composing new signs through concatenation of subunit models and by tuning the composite models with only small sets of examples. However, an additional knowledge of how to break down signs into subunits is needed.

Different vision-based subunit segmentation algorithms have been developed. Following Liddell and Johnson's movement-hold model the authors of [3] propose modeling each sign (word) as a series of movement and hold segments. Kraiss et al. in [1] present an iterative process of data-driven extraction of subunits using hidden Markov models (HMMs). In all following steps, two state HMMs for subunits determined in prior iteration step are concatenated to models of single signs. The boundaries of subunits for the next step result from the alignment of appropriate feature vector sequence to the states by the Vitterbi algorithm. Han et al. in [4] define the subunit boundary using hand motion discontinuity. Temporal clustering by dynamic time warping is adopted to merge similar segments.

In this paper we propose a new approach where the subunits' boundary points are considered as decision variables in a multiobjective optimization problem. The problem consists in finding subunits which can be grouped in clusters of good quality. The quality is measured by two cluster validity indices, one based on entropy [5] and another the Dunn's index [6], [7]. The indices are optimized simultaneously using lexicographic ordering [8] and an immune-based evolutionary algorithm [9], [10]. The approach refers to clustering of time series data [11], [12], multiobjective clustering [13], [14], and cluster-based time series segmentation [15]. The contribution of the paper lies in (1) formulation of the problem of determining subunits for sign language recognition as a multiobjective cluster optimization, (2) proposition of a solution method, and (3) verification of the approach by experiments on both synthetic and real data.

The rest of the paper is organized as follows. Section 2 contains formulation of the problem. Section 3 gives the details of the proposed solution method. The results of experiments using synthetic data, as well as data obtained for isolated words of the Polish Sign Language (PSL) are given in section 4. Section 5 concludes the paper.

2 Problem formulation

Let $S = \{X_1, X_2, \ldots, X_n\}$ denote a data set, where $X_i = \{x_i(1), x_i(2), \ldots, x_i(T_i)\}$ is a sequence of real valued vectors representing a signed word. All feature vectors $x_i(t), t\epsilon\{1, 2, \ldots, T_i\}, i\epsilon I = \{1, 2, \ldots, n\}$ have identical structures. The integers $t = 1, 2, \ldots$ represent equidistant time points. Two time sequences X_i and $X_{j \neq i}$ may represent different words or different realizations of the same word.

Let us consider a decomposition D, which, for each $i\epsilon I$, defines a number $k_i = k_i(D) \geq 1$ and k_{i-1} cut points $t_i^j = t_i^j(D)$, where $1 < t_i^1 < t_i^2 < \ldots < t_i^{k_i-1} < T_i$. The decomposition means that X_i is partitioned into k_i subsequences. The first subsequence $s_i^1(D)$ starts at $t = 1$ and ends at $t = t_i^1$, the next subsequence

$s_i^2(D)$ starts at $t = t_i^1$ and ends at $t = t_i^2$, and so on until the last subsequence $s_i^{ki}(D)$ which starts at $t = t_i^{k_i-1}$ and ends at T_i. The resulting data set $S'(D) = \{s_1^1(D), \ldots, s_1^{k_1(D)}(D), s_2^1(D), \ldots, s_2^{k_2(D)}(D), \ldots, s_n^1(D), \ldots, s_n^{k_n(D)}(D)\} = \{s'_1, s'_2, \ldots, s'_{n'}\}$ contains $n' = n'(D) = \sum_{i=1}^{n} k_i(D)$ sequences. The length of each subsequence is constrained by the minimal l_{min} and the maximal l_{max} number of points. We propose determining a good decomposition into subsequences by solving a multicriteria decision problem, based on the following main steps: (i) partition the set $S'(D)$ into m (a given number) clusters, i.e. $S'(D) = \{C_1(D), C_2(D), \ldots, C_m(D)\}$, (ii) evaluation of the decomposition D using a vector of $p > 1$ criteria (indices) $J(D) = [J_1(D), J_2(D), \ldots, J_p(D)]$ which characterizes the quality of the resulting clusters. In next sections we suggest a solution method and we show the results of experiments on both synthetic and real data sequences.

3 Basic elements of the solution method

3.1 Distance measure

To compare discrete sequences we use dynamic time warping (DTW) [6], [16]. Given two time series $Q = \{q(1), q(2), \ldots, q(T_q)\}$ and $R = \{r(1), r(2), \ldots, r(T_r)\}$ DTW aligns the two series so that their difference is minimized. To this end, a $T_q \times T_r$ matrix, where the (i, j) element of the matrix contains the distance $d(q(i), r(j))$ between two points $q(i)$, and $r(j)$. Usually the Euclidean distance is used. A warping path, $W = w_1, w_2, \ldots, w_K$ where $max(T_q, T_r) \leq K \leq T_q + T_r - 1$, is a set of matrix elements that satisfies three constraints: boundary condition, continuity and monotonicity. The boundary condition constraint requires the warping path to start and finish in diagonally opposite corner cells of the matrix. That is $w_1 = (1, 1), w_K = (T_q, T_r)$. The continuity constraint restricts the allowable steps to adjacent cells. The monotonicity constraint forces the points in the warping path to be monotonically arranged in time. The warping path that has the minimum distance $d_{DTW} = \sum_{k=1}^{K} \frac{w_k}{K}$ between the two series is of interest. Dynamic programming is used to effectively find this path. To prevent pathological warping, where a relatively small section of one sequence maps to a much larger section of another, warping window constraints are applied which, additionally, speed up the computation [16]. The warping window usually defines the search region as a narrow strip around the diagonal connecting points w_1, w_K.

3.2 Clustering procedure

As the clustering algorithm we propose minimum entropy clustering (MEC) described in [5]. Entropy is a measure of information and the uncertainty of a random variable. The method uses entropy measured on a posteriori probabilities as the criterion of clustering. In fact, it is the conditional entropy of clusters given the observations. The problem of clustering consists of two subproblems (1) estimating a posteriori probabilities and (2) minimizing the entropy. Experiments presented in [5] show that MEC performs significantly better than k-means clustering, hierarchical clustering, SOM and EM. Moreover, it can correctly reveal the structure of data and effectively identify outliers simultaneously.

In our problem we used the Java package prepared by the authors of [5] and accessible online [17]. As it performs clustering of vector defined data we considered two approaches based on $n'(D)$ similarity vectors representing the set $S'(D)$ of subsequences to be clustered. Each of the similarity vectors has $n'(D)$ elements where the $j-th$ element of the $i-th$ similarity vector is determined as the DTW distance between the subsequences s'_i and s'_j in the set $S'(D)$. In the first case MEC performs clustering of the similarity vectors. Alternatively, shorter vectors obtained from the similarity vectors by the PCA can be used.

3.3 Clustering results evaluation

The vector index $J(D)$ introduced in section 2 actually contains two elements. The first, more important, is the conditional entropy minimized by MEC. The second in the hierarchy is the Dunn's index DI [6], [7]. It is defined by two parameters: the diameter $diam(C_i)$ of the cluster C_i and the set distance $\delta(C_i, C_j)$ between C_i and C_j, where

$$diamC_i = max_{x,y\epsilon C_i}\{d(x,y)\}, \delta(C_i, C_j) = min_{x\epsilon C_i, y\epsilon C_j}\{d(x,y)\} \tag{1}$$

and $d(x, y)$ indicates the distance between points x, y.

$$DI = min_{1\leq j\leq m}\{min_{1\leq i\leq m, i\neq j}\{\frac{\delta(C_i, C_j)}{max_{1\leq k\leq m} diamC_k}\}\} \tag{2}$$

Larger values of DI correspond to good clusters.

Note that the distances needed in DI can be considered as distances between the similarity vectors or, alternatively, as distances between respective sequences. Obviously, in the second approach necessary information is extracted from the similarity vectors.

3.4 Optimization algorithm

As follows from section 3.3 our problem is a multiobjective optimization problem (MOP) with two criteria. To solve MOPs evolutionary algorithms are often used. Evolutionary algorithms deal simultaneously with a set of possible solutions (the so called population) which allow us to find several members of the Pareto optimal set in single run of the algorithm [18].

Our approach to solve the MOP adopts the immune-based algorithm CLONALG originally used for single-objective optimization [9], [10]. We use lexicographic ordering [8]. Here the single objective J_1 (considered the most important) is optimized without considering J_2. Then the J_2 is optimized but without decreasing the quality of the solution obtained for J_1. In the sequel we shortly describe the algorithm, the encoding method, and the mutation operator.

CLONALG. The main loop (repeated *gen* times, where *gen* is the number of generations) consists of four main steps: one initial step where all the elements of the population are evaluated and three transformation steps: clonal selection, mutation, apoptosis.

1. Evaluation. For each element D in the population P compute $J_i(D), i = 1, 2$ and perform lexicographic ordering of the elements.
2. Clonal selection. Choose a reference set $P_a \subset P$ consisting of h elements at the top of the ranking obtained in step 1.
3. Mutation.
   - 3.1. For each $D \epsilon P_a$ make c mutated clones $Dc_j, j = 1, 2, \ldots, c$, compute their values $J_1(Dc_j), J_2(Dc_j)$, and place the clones in the clonal pool CP.
   - 3.2. Lexicographically order the elements of $P \cup CP$, choose a subset $P_c \subset P \cup CP$ containing N best elements, where N denotes the size of P.
4. Apoptosis. Replace b worst elements in P_c by randomly generated elements.
5. Set $P \subset PC$.

In the algorithm the current population P is mixed with the clonal pool CP and the predefined number of best elements (i.e. at the top of the ranking) is picked up to form new population. The last step of the main loop replaces b worst solutions by randomly generated elements.

Encoding and mutation. Each element of the population P represents a decomposition D of the set S into a set S'(see section 2). It has the form of the integer valued vector $D = [t_1^1, t_1^2, \ldots, t_1^{k_1-1}, t_2^1, t_2^2, \ldots, t_2^{k_2-1}, \ldots, t_n^1, t_n^2, \ldots, t_n^{k_n-1}]$ composed of the cut points of the original sequences. The mutation process consists of a given number M of mutations conducted on a population element. The mutation means an operation randomly chosen from the following variants: (a) add cut point (probability 1/4), (b) remove cut point (probability 1/4), (c) move cut point (probability 1/2). In all cases a subsequence is randomly selected and, depending on a drown variant, it is: (a) divided into two shorter subsequences, (b) joined together with its preceding subsequence, (c) made shorter or longer by shifting its initial point. The new cut point in (a) and (c) is placed in a position randomly chosen from the corresponding set of feasible points, i.e. the points for which the resulting subsequences satisfy the length constraints. Similarly, the union in (b) is accepted if the resulting subsequence is not too long.

4 Experiments

In this section we present results of two experiments. In the first case synthetic data are considered, the other experiment is based on real sequences obtained for signed Polish words.

4.1 Synthetic data

The set S consists of six sequences presented in fig.1. In each sequence one can distinguish subsequences which are identical or mutually related by a nonlinear time scale transform. We considered partitioning of the sequences into subsequences which can be grouped into (i) two clusters ($m = 2$), (ii) four clusters ($m = 4$). In both cases the minimum (l_{min}) and the maximum (l_{max}) subsequence lengths (see section 2) are defined.

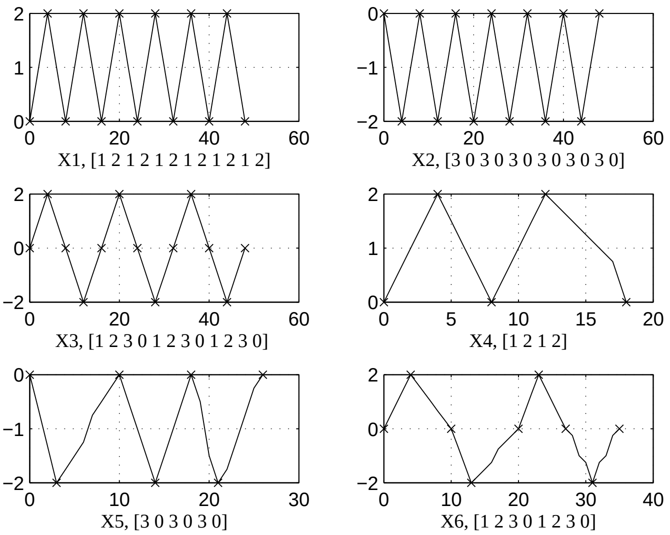
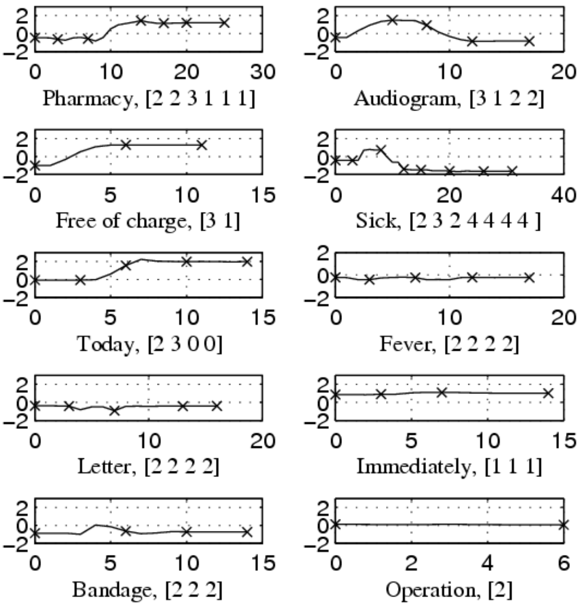


Fig. 1. Sequences $X_1 - X_6$ used in the experiment; automatically determined subsequences' boundaries for $m = 4$ are marked, resulting transcriptions based on four subunits $0, \ldots, 3$ are given in brackets.

The following parameters of the optimization procedure were used in the experiment: $N = h = 100, c = 15, b = 10, M = 2, gen = 60, l_{min} = 6, l_{max} = 12$ (2 clusters), $l_{min} = 4, l_{max} = 8$ (4 clusters). The best result obtained for $m = 4$ is characterized in fig. 1. Automatically obtained partitioning for $m = 2$ was also consistent with the result expected by human.

4.2 Real data

Sequences used in this experiment represent 10 signed words of PSL. Each sequence was chosen as a medoid of 40 realizations of appropriate word performed by two signers. Fig. 2. shows normalized ($mean = 0, stdev = 1$) values of the horizontal placement of the right hand center vs. frame number, obtained from pictures registered by the camera with the rate of 25 f/s. Parameters used in this experiment are the same as in the experiment with four clusters in subsection 4.1. We solved the optimization task for 2, . . . , 6 clusters. The best result (with the greatest value of the Dunn's index) has been obtained for five clusters ($m = 5$). Fig. 2. shows that the subunits with the same labels are quite similar, although of different lengths, whereas the subunits with different labels differ.



Pharmacy, [2 2 3 1 1 1]

Audiogram, [3 1 2 2]

Free of charge, [3 1]

Sick, [2 3 2 4 4 4 4]

Today, [2 3 0 0]

Fever, [2 2 2 2]

Letter, [2 2 2 2]

Immediately, [1 1 1]

Bandage, [2 2 2]

Operation, [2]

Fig. 2. Sequences representing signed words; automatically determined subsequences' boundaries for $m = 5$ are marked, resulting transcriptions based on five subunits $0, \ldots, 4$ are given in brackets.

5 Conclusions

Large-vocabulary systems of sign language recognition require the modeling of signed expressions in smaller units than words. However, an additional knowledge of how to break down signs into subunits is needed. In vision-based systems the subunits are related to visual information. As linguistic knowledge about the useful partition of signs in regard of sign recognition is not available, the construction of an accordant partition is based on a data-driven process when signs are divided into segments that have no semantic meaning, then similar segments are grouped and labeled as a subunit. In this paper we propose a new approach to determining the subunits. Subunits' boundaries are considered as decision variables in a multiobjective optimization problem. We use two objective functions, entropy and the Dunn's index, as measures of cluster quality. These functions are optimized simultaneously. The method has been successfully verified, but there remain some open questions. The number of clusters is determined experimentally. In the future it will be included as the additional decision variable in the optimization task. Second question concerns including other cluster validity indices and using other optimization approaches. We use lexicographic ordering and an immune-based evolutionary algorithm, but other evolutionary optimization methods may be considered, see e.g. [18]. We will consider these issues in future research. A next step will be related to more advanced experimentation including recognition words and sentences of the PSL.

Acknowledgement

This research was supported by the Polish Ministry of Higher Education under grant N N516 369736.

References

1. Kraiss, K.F.: Advanced Man-Machine Interaction. Springer, Berlin (2006)
2. Ong, S.C.W., Ranganath, S.: Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning. IEEE Trans. PAMI 27, 873–891 (2005)
3. Vogler, C., Metaxas, D.: A Framework for Recognizing the Simultaneous Aspects of American Sign Language. Computer Vision and Image Understanding 81, 358–384 (2001)
4. Han, J., Awad, G., Sutherland, A.: Modelling and Segmenting Subunits for Sign Language Recognition Based on Hand Motion Analysis. Pattern Recognition Letters 30, 623–633 (2009)
5. Li, H., Zhang, K., Jiang, T.: Minimum Entropy Clustering and Applications to Gene Expression Analysis. In: 3rd IEEE Computational Systems Bioinformatics Conference, pp. 142–151 (2004)
6. Xu, R., Wunsch, D.C.: Clustering. J. Wiley and Sons, Inc., Hoboken, New Jersey (2009)
7. Maulik, U., Bandyopadhyay, S.: Performance Evaluation of Some Clustering Algorithms and Validity Indices. IEEE Trans. PAMI 24, 1650–1654 (2002)
8. Miettinen, K.M.: Nonlinear Multiobjective Optimization. Kluwer Acad. Publ. (1998)
9. De Castro, L.N., Von Zuben, F.J.: Learning and Optimization Using the Clonal Selection Principle. IEEE Trans. on Evolutionary Computation 6, 239–251 (2002)
10. Trojanowski, K., Wierzchon, S.: Immune-Based Algorithms for Dynamic Optimization. Information Sciences 179, 1495–1515 (2009)
11. Bicego, M., Murino, V., Figueiredo M.A.T.: Similarity-based Classification of Sequences Using Hidden Markov Models. Pattern Recognition 37, 2281–2291 (2004)
12. Liao, T.W.: Clustering of time series data - a survey. Pattern Recognition 38, 1857–1874 (2005)
13. Handl, J., Knowles, J.: An Evolutionary Approach to Multiobjective Clustering. IEEE Trans. on Evolutionary Computation 11, 56–76 (2007)
14. Saha, S., Bandyopadhyay, S.: A symmetry-based multiobjective clustering technique for automatic evolution of clusters. Pattern Recognition 43, 738–751 (2010)

15. Tseng, V.S., Chen, C.H., Huang, P.C., Hong, T.P.: Cluster-based genetic segmentation of time series with DWT. Pattern Recognition Letters 30, 1190–1197 (2009)
16. Ratanamahatana, C.A., Keogh, E.: Three Myths about Dynamic Time Warping Data Mining. In: SIAM Int. Conf. on Data Mining, pp. 506–510 (2005)
17. Minimum Entropy Clustering Java package, `http://www.cs.ucr.edu/~hli/mec/`
18. Coello Coello, C.A.: Evolutionary Multi-Objective Optimization: A Historical View of the Field. IEEE Computational Intelligence Magazine 1, 28–36 (2006)