SEARCHING AND INDEXING DISTORTED DATA COLLECTIONS

Tomáš Kocyan, Jan Martinovič, Michal Podhorányi

VŠB - Technical University of Ostrava
IT4Innovations
17. listopadu 15/2172, 708 33 Ostrava, Czech Republic

tomas.kocyan@vsb.cz, jan.marinovic@vsb.cz, michal.podhoranyi@vsb.cz

ABSTRACT
Success of many models and artificial intelligence methods strongly depends on ability to quickly and precisely search input data collection. Despite the existence of many algorithms for faster searching, the most of them fail while processing distorted input. Unfortunately, the distortion is natural for many types of data collections, especially for measurements of natural phenomena such as precipitations, river discharge volume etc. In this type of collections, there are no exact levels for generated values. This paper discusses possibilities of indexing and searching such distorted inputs and also proposes an alternative approach for their indexing. The proposed approach utilizes the Voting Experts algorithm for splitting the input regarding statistical indicators, the Dynamic Time Warping for dealing with distorted inaccuracies and hierarchical clustering for grouping similar sequences. Finally, the sample result of proposed algorithm applied on data collections consisting of measured river discharge volumes is shown.

Keywords: time series, indexing, dynamic time warping, voting experts, symbolic approximation

1. INTRODUCTION

Searching time series data collections is a usual task in applications of many domains and in some of these applications, speed and accuracy of searching can be very crucial. This is mainly true for machine learning methods such as Case-Based Reasoning (Watson, 2007), where ability of precise searching similar situations strongly reflects overall success and usability of whole system.

There are many information technologies infrastructures which support hydrologic science. Common possibility is to storage hydrologic observations in a relational database. The purpose for such database is to store hydrologic observations data in an optimizing system. Main task for such system is to provide information in standard formats, to afford sufficient auxiliary metadata about data values and to design whole system for fast searches. A relational database format is used to provide querying capability to allow data retrieval supporting diverse analysis. Very important part of such system or database is organizing data into clear departments such as where they have been measured, at which locations and for what period of time (Horsburgh, 2008). Logical structure of system facilitates searching of observed data or different kinds of patterns (e.g. rain or discharge patterns). Searching in such type of database is usually done by means of commonly used algorithms in hydrology. Spatial patterns in catchment hydrology are often extreme events and therefore developing of programs which might reveal them is very significant point of view in hydrologic community.

Many papers are focused on indexing and searching relatively short time series (called episodes) of the same length (e.g. Chen, Huang, Wang, and Wang 2009, Keogh and Pazzani 2000) stored separately, which were either collected in the form of partial outputs (e.g. of modeling software) or created by splitting a long-standing time series in equidistant intervals. In such cases, the indexing is pretty simplified. Despite the possible distortion, the only thing to be done is to choose a suitable metric or method defining similarity between two episodes. Then, common indexing methods can be used. For dealing with eventual distortions or inaccuracies, the Euclidian distance is usually replaced by Dynamic Time Warping (Keogh and Pazzani 2000), which will be described in Section 2.3.

However, a serious problem with indexing arises when the data collection is not split, e.g. it has a form of a single long-terming time series. In such cases, at first, it is necessary to meaningfully split the time series into the particular episodes. It can be done either by cutting the time series into equidistant intervals, or by analyzing its behavior and trying to identify the meaningful parts. The first approach is much easier and faster, but it has an obvious disadvantage – there is absolutely no control over the cutting process. By this way, a meaningful episode may be divided into the two parts, which causes loss of the important information. For this reason, we decided to follow the second approach using the Voting Experts (VE) algorithm introduced in Section 2.2. Since this algorithm is originally designed for processing categorical time series, the input has to be first converted into suitable format. It is done using the Symbolic Aggregate Approximation (SAX) described in

Section 2.1. Tests showed (Kocyan, 2012), that the combination of SAX and VE splits the input correctly, but very roughly (high precision, low recall). It means that the found splits were correct, but not complete. To solve this, we extended the splitting process by a refining part. Whole process of the fine splitting including creation of an index file can be seen in Figure 1: First of all, categorical time series is roughly split into the episodes. The episodes are then clustered by hierarchical clustering into the reasonable number of clusters of similar episodes. Using the DTW, the clustered episodes are split into shorter episodes. If the fineness is not sufficient, the process of clustering and splitting can be repeated. On the other hand, if the required fineness is reached, clusters' representatives can be chosen and the index file can be built.

This paper is organized as follows: First, used tools will be introduced in Section 2. Second, the proposed approach briefly introduced earlier will be described in detail. In the Experiments section, results of proposed approach will be demonstrated and compared with usual methods for indexing. At the end, both advantages and disadvantages of the approach will be summarized and the future work will be outlined.

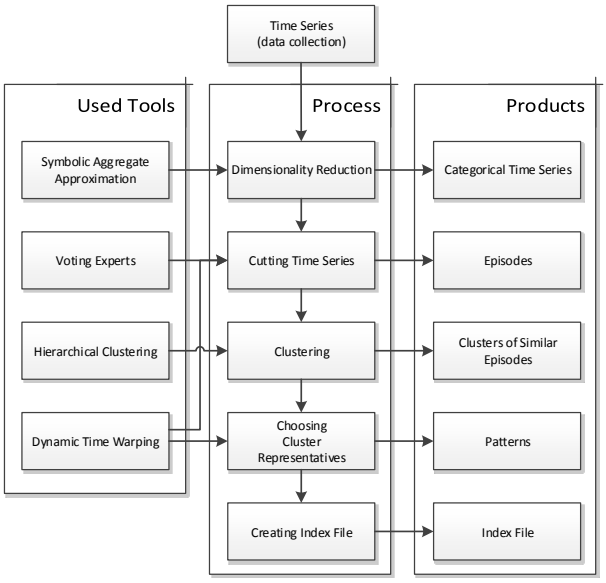

Figure 1: Proposed Approach Schema

2. USED TOOLS

In this section, used tools mentioned earlier will be described in necessary detail. For eventual deeper exploration of particular tools, please follow the corresponding references.

2.1. Symbolic Aggregate Approximation

Symbolic Aggregate approXimation (Lin, Keogh, Wei, and Lonardi 2007) is a simple dimensionality reduction method for transforming a time series T of length n into the string of length l, where $l \ll n$. The resultant string is composed from an alphabet A of size $a > 2$. The algorithm works in three steps: First, the time series is Z-Normalized (Goldin and Kanellakis 1995), i.e. transformed into the time series with approximately zero mean and standard deviation close to 1. Then, in the second step, the Piecewise Aggregate Approximation (Lin, Keogh, Wei, and Lonardi 2007) is applied in order to reduce time dimension. At the end, the reduced time series is converted into a string. An example of original, z-normalized, PAA and SAX transformed time series is shown in Figure 2.

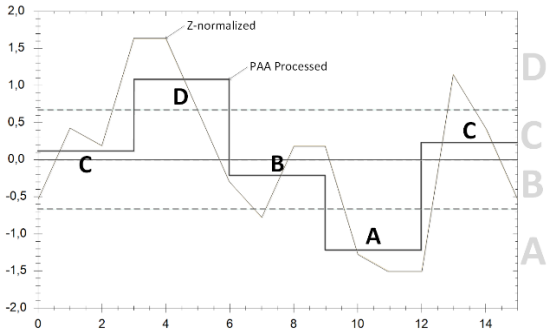

Figure 2: Example of PAA and SAX

2.2. Voting Experts Algorithm

The Voting Expert Algorithm is a domain-independent unsupervised algorithm for segmenting of categorical time series into meaningful episodes. It was first presented by Cohen and Adams (2001). Since this introduction, the algorithm has been extended and improved in many ways, but the main idea is always the same. The basic Voting Experts algorithm is based on the simple hypothesis that natural breaks in a sequence are usually accompanied by two statistical indicators (Cohen, Adams, and Heeringa 2007): low internal entropy of episode and high boundary entropy between episodes. The basic Voting Experts algorithm consists of the following three main steps:

- Build an nGram tree from the input, calculate statistics for each node of this tree (internal and boundary entropy) and standardize these values in nodes at the same depth.
- Pass a sliding window of length *n* over the input and let experts vote. Each of the experts has its own point of view on current context (current content of the sliding window) and votes for the best location for the split. The first expert votes for locations with the highest boundary entropy, the second expert votes for locations with a minimal sum of internal split entropy. By this way, the votes are counted for each location in the input.
- Look for local maximums which overcome selected threshold. These points are adepts for a split of sequence.

For detailed explanation of each of mentioned steps see (Cohen, Adams, and Heeringa 2007). Tests showed that the algorithm is able to segment selected input into meaningful episodes successfully. It was tested in many domains of interest, such as looking for words in a text (Cohen and Adams 2001) or segmenting of speech record

(Miller, Wong, and Stoytchev 2009). There are several ways how to improve the basic Voting Experts algorithm. Simply we can divide these improvements into the two main groups. On the one hand, a custom "expert" can be added to voting process (for example Markov Expert by Cheng and Mitzenmacher (2005)) and receive additional point of view on your input. On the other hand, there are methods based on repeated or hierarchical segmenting of the input (Miller and Stoytchev 2008, Hewlett and Cohen 2009).

2.3. Dynamic Time Warping

Nowadays, searching the time series databases generated by computers, which consists of accurate time cycles and which achieves a determined finite number of value levels, is a trivial problem. A main attention is focused more likely on the optimization of searching speed. A non-trivial task occurs while comparing or searching the signals, which are not strictly defined and which have various distortions in time and amplitude. As a typical example, we can mention measurement of functionality of human body (ECG, EEG) or the elements (precipitation, flow rates in riverbeds), in which does not exist an accurate timing for signal generation. Therefore, comparison of such episodes is significantly difficult, and almost excluded while using standard functions for similarity (distance) computation. Examples of such signals are presented in Figure 3. A problem of standard functions for similarity (distance) computation consists in sequential comparison of opposite elements in both episodes (comparison of elements with the identical indexes).

DTW is a technique for finding the optimal matching of two warped episodes using pre-defined rules (Muller 2007). Essentially, it is a non-linear mapping of particular elements to match them in the most appropriate way. The output of such DTW mapping of episodes from Figure 3 can be seen in Figure 4. This approach was used for example for comparison of two voice patterns during an automatic recognition of voice commands (Rabiner 1993). The main goal of DTW method is a comparison of two time dependent episodes X and Y, where $X = (x_1, x_2, \dots, x_n)$ of the length $n \in \mathbb{N}$ and $Y = (y_1, y_2, \dots, y_m)$ of the length $m \in \mathbb{N}$, and to find an optimal mapping of their elements. A detailed description of DTW including particular steps of the algorithm is presented in (Muller 2007).

2.4. Searching the mutual subsequences

Despite the fact that the DTW has its own modification for searching subsequences, it works perfectly only in a case of searching exact pattern in some signal database. This case is demonstrated in Figure 5, where sequence s_1 = '3456' exactly matches the corresponding subsequence in sequence s_2 = '12**3456**789'. However, in real situations exact patterns are not available because they are surrounded by additional values (Figure 5a), or even repeated several times in the sequence (Figure 5b).

Unfortunately, the basic DTW is not able to handle these situations and it fails or returns only a single occurrence of the pattern. To deal with this type of situations, own DTW modification was created. This modification is able to find the longest common time warped subsequences under selected restrictions. Because this paper's topic is not focused on modifications of the DTW, see (Kocyan, 2012) for detail explanation.

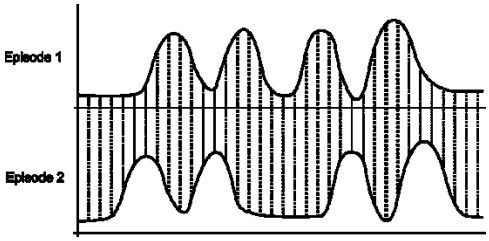

Figure 3: Standard Metrics Comparison

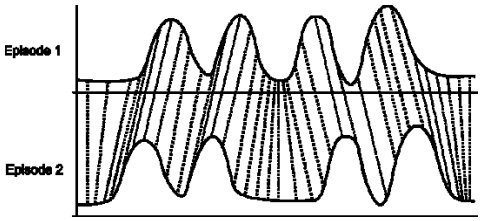

Figure 4: DTW Comparison

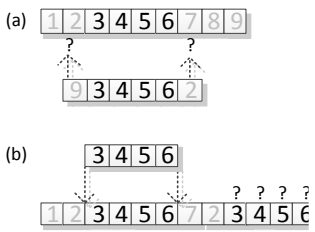

Figure 5: Basic DTW inaccuracies

2.5. Using DTW For Finding Cluster Representative

The DTW can be also utilized in cases, where it is necessary to gain the most suitable representative of the set of similar episodes. For this purpose, sometimes it is possible to use simple average of episodes X and Y, which means that for a representative episode R is valid, that:

$$R_i = \frac{X_i + Y_i}{2}, \forall i = 1, \dots, o, where\ o = |X| = |Y|. \qquad (1)$$

However, this approach is not sufficient in cases, where we have data with distortion. Examples of such episodes are presented in Figure 6a and 6b. If only we used simple average presented in Equation 1, we would achieve an episode showed in Figure 6c. As we can see, this episode absolutely is not a representative and all the information about the episode course is lost. For this purpose, we designed our own DTW modification resulting in representative in Figure 6d. For detailed explanation how to derive representative using the DTW, see (Kocyan, 2012).

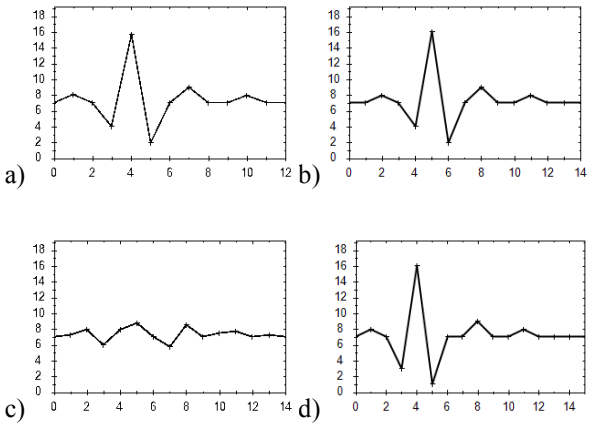

Figure 6: Sample Figure Caption

2.6. Voting Experts DTW Post-Process

Proposed solution for Voting Experts improvement takes the task of using Dynamic Time Warping algorithm (introduced above) and high precision cuts from Voting Experts as a starting point for looking for typical patterns located in the input. The basic idea is to refine the sparse set of high precision cuts into regular sequences as correctly as possible. The principle is very simple (as shown in Figure 7). If there are high precision cuts in the input (such as cuts A, B, C and D in Figure 7) and if the shorter sequence (bounded by cuts C and D) is a subsequence of the longer one (bounded by cuts A and B), we can deduce new boundaries E and F by projecting the boundaries of common subsequence to the longer sequence. In this very simplified example the sequences were composed by definite number of values and limited length, so the evaluation is quite straightforward.

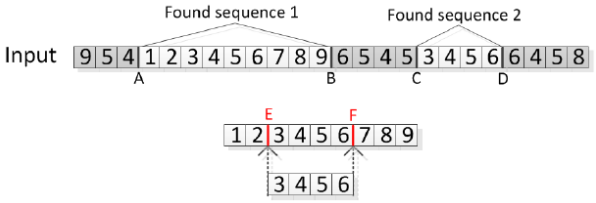

Figure 7: Refinement of sequences

In the case of application of previously mentioned process on distorted data, it is necessary to slightly modify it. Typical episodes of measurement of natural phenomena (such as precipitations, measured discharge volume etc.) are, unfortunately, subject to distortion in both time and value axes. For this reason, the Dynamic Time Warping (DTW) algorithm is used for this purpose. For our purposes, the DTW algorithm will be used as a tool for finding the longest common subsequence of two sequences (described in (Kocyan, 2012) in detail).

The main idea of the Voting Experts DTW post-process is summarized into the following steps:

1. First of all, the high precision (but not complete) cuts are created by splitting the input with high level of threshold by the Two-Way Voting Experts method.
2. Let's suppose that there are m unique sequences which have been created according to the cuts from step 1.
3. A $m \times m$ distance matrix is build.
4. For each pair in this matrix, where the length of sequence s_1 is bigger than length of sequence s_2:
   (a) The optimal mapping of shorter sequence s_2 to the longer sequence s_1 is found by using DTW modified.
   (b) If the mapping cost does not overcome selected threshold, the longest sequence s_1 stores the shorter sequence s_2 into its own list of similar sequences. By this way, every sequence gets its own list of the most similar shorter sequences.
   (c) Each of the shorter sequences points to positions in the longer sequence, where it should be split. Because there is usually more than one similar shorter sequence, it is pointed to several locations whereas many of these locations are duplicated. For this reason, the votes are collected into internal vote storage.
   (d) After these votes are collected, the local maximums are detected. These places are suggested as new cuts in original input.
5. The granted votes from step 4(d) are summed with votes of frequency and entropy experts in the input. Subsequently, the local maximums of votes are searched again. The cuts are made in locations where the number of granted votes is higher than the specified threshold.
6. Algorithm ends or it can continue with step 2 for further refinement.
7. After each post-process iteration, all found patterns can be received from the 4(b) step. Actually, the found patterns are groups of similar chunks. For building an index file, it is necessary to choose a representative in the same way as described in (Kocyan, 2013).

3. EXPERIMENTS

This part is focused on applying proposed algorithm on data collection containing measured river discharge volume. For this purpose, an artificial collection using analytic rainfall-runoff model and covering all possible situations was created.

Our process of creating data collection is done in three steps: First of all, the precipitations input containing real and artificial episodes is constructed. The input was sampled with one hour step for the total length of 10 years. Moreover, the partial episodes were distorted in both axes using predefined rules with respect to the normal distribution. After that, the input was split into short episodes and each episode was used as an input to the analytic rainfall-runoff model. At the end, the model's outputs were concatenated into the long-standing time series. As an environment for simulation run, the Floreon+ system was selected (Martinovič and Kuchař 2010). The input was composed from rainfall

episodes generated for both the left and the right tributary separately. Inserted episodes are chosen from one of the following categories:

1. Artificial Flash Flood precipitations.
2. Artificial Regional precipitations.
3. Real precipitations received from database.
4. Empty episode with no raining for decreasing the discharge volume.

The type of each episode was chosen randomly, but with a specific probability. By adjusting these probabilities, the final character can be strongly influenced. Once the episodes are selected, they have to be distorted to get closer the reality. The distortion is done in two stages: First, the selected rainfall episode is significantly distorted and used as a precipitation pattern for a single tributary. Then, this significantly distorted episode is finely distorted separately for each station belonging to the current tributary. Thanks to the distortion process described below, we get a random unique precipitation episode for both left and right tributary, but slightly varied for particular station. For detailed information see (Kocyan, 2014).

3.1. Post-Process adjustment

Adjusting the post-process part consists of several parameters. Despite the fact that the algorithm is built as domain independent, some domain knowledge is needed. First of all, it is necessary to set up the Voting Experts algorithm providing the rough segmentation. This algorithm needs categorical data on its input, so the measured data have to be converted first. It is done using the SAX described in paragraph 2.1. During the SAX conversion, a dimension reduction using PAA (described in Section 2.1) can be performed. Tests showed the PAA size of 6 sufficiently reduces the dimension with no loss of important information, regarding the nature of river discharge data. Data was converted in raw format, using its first differences and coefficients of growth. The number of output levels was set in all cases to 7. A sample of SAXed data can be seen in figures 8, 9 and 10. From the histogram in Figure 11 it is evident that using the first differences spreads the categorical data most symmetrically. For this purpose, this type of data was used.

Once the data is converted into the suitable format for Voting Experts, the sliding window size and threshold have to be determined. In order to take the necessary context into account, the size of sliding window have to be bigger than expected patterns. In our case of searching river discharge patterns, at least a “week” window has to be used. Since the discharge volume is measured with an hour step, a sliding window with a length of at least 28 (168 hours of measurement, PAA of 6 values) have to be considered. At the end, the window size was set to 30 and the threshold to 10.

Using this settings, cuts described as “VE CUT” in Figure 12 were made. By applying the post-process described above, additional cuts were progressively added (marked with PP and cycle number). Since it is impossible to visualize whole data collection with increasing number of cuts, only increasing number of cuts and decreasing average length of sequences during the partial post-process cycles are shown in Figure 13.

Once enough short episodes are received, they are clustered into the clusters of similar episodes. For each of this similar episodes, a representative is derived as shown in Figure 14.

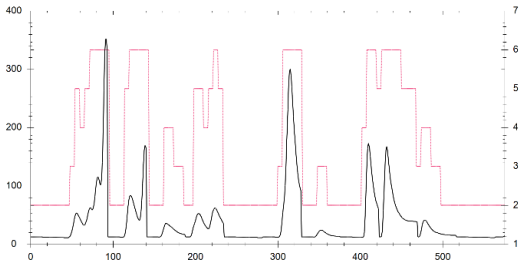

Figure 8: SAXed Raw Data

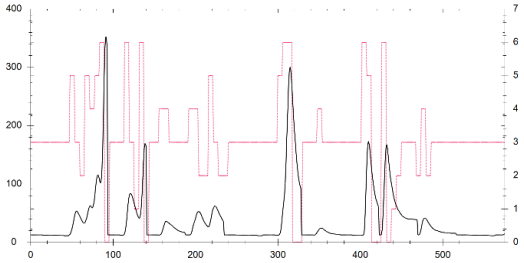

Figure 9: SAXed First Differences of Data

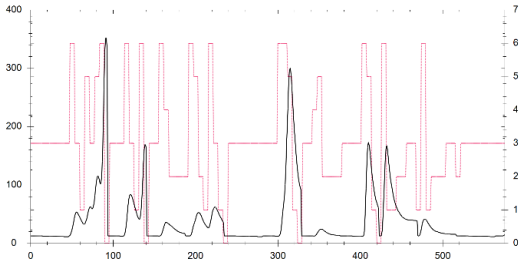

Figure 10: SAXed Growth Coefficients of Data

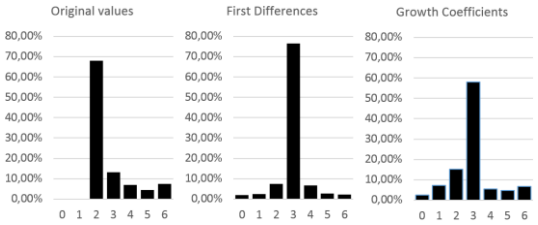

Figure 11: Histograms of SAXed Data

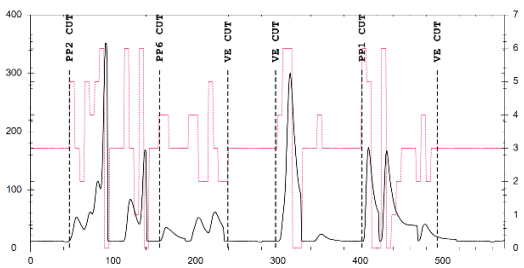

Figure 12: Voting Experts and Post-Process Cuts

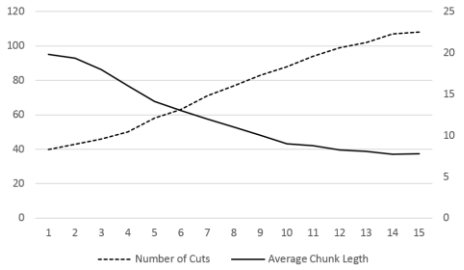

Figure 13: Voting Experts and Post-Process Cuts

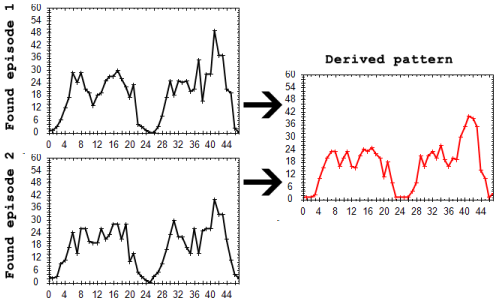

Figure 14: Found Episodes and Derived Pattern

CONCLUSION

Tests showed that the proposed algorithm is able to find characteristics patterns in distorted data collection and build an index file for faster and more precise searching. Since the algorithm is mostly based on DTW and mutually independent parts, it is very easy to parallelize the computations and rapidly speed up the pattern search. Future work will be focused on such optimization and automatic adjustment of algorithm's parameters.

ACKNOWLEDGEMENT

This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and the national budget of the Czech Republic via the Research and Development for Innovations Operational Programme, by the project New creative teams in priorities of scientific research, reg. no. CZ.1.07/2.3.00/30.0055, supported by Operational Programme Education for Competitiveness and co-financed by the European Social Fund.

REFERENCES

Cheng, J., Mitzenmacher, M., 2005. Markov Experts. *Proceedings of the Data Compression Conference (DCC)*.

Cohen, P. R., Adams, N., 2001. An Algorithm for Segmenting Categorical Time Series into Meaningful Episodes, *Proceedings of the Fourth Symposium on Intelligent Data Analysis, Lecture Notes in Computer Science*.

Cohen, P. R., Adams, N., and Heeringa, B, 2007. Voting Experts: An Unsupervised Algorithm for Segmenting Sequences. *In Journal of Intelligent Data Analysis*.

Chen, J., Huang, K., Wang, F., and Wang, H., 2009. E-learning Behavior Analysis Based on Fuzzy Clustering. *In Proceedings of International Conference on Genetic and Evolutionary Computing*.

Goldin, D., Kanellakis, P., 1995. On similarity queries for time-series data: Constraint specification and implementation. *1st International Conference on the Principles and Practice of Constraint Programming*, pages 137–153, France.

Hewlett, D., Cohen P., 2009. Bootstrap Voting Experts. *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence (IJCAI)*.

Horsburgh, J.S., Tarboton, D.G., Maidment, D.R., Zaslavsky, I. 2008. *A relational model for environmental and water resources data. Water Resources Research Hydrological Sciences 44* (5).

Keogh, E., Pazzani, M., 2000. Scaling up dynamic time warping for datamining applications. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. Pages 285-289

Kocyan, T., Martinovič, J., Kuchař, Š., and Dvorský J., 2012. Unsupervised Algorithm for Post-Processing of Roughly Segmented Categorical Time Series. *Proceedings of Databases, Texts, Specifications, and Objects*.

Kocyan, T., Martinovič, J., Dráždilová, P., and Slaninová K., 2013. Recognizing Characteristic Patterns In Distorted Data Collections. *Proceedings of The European Modeling and Simulation 2013*.

Kocyan, T., Podhoranyi, M., Fedorčák, D., and Martinovič, J., 2014. Generating Rainfall-runoff Data Collection For Calibration Of Machine Learning Driven Models. *Proceedings of SGEM Multidisciplinary Scientific Conference*.

Lin, J., Keogh, E., Wei, L., Lonardi, S., 2007. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, v.15 n.2, p.107-144, October 2007

Maheras, P., Kolyva-Machera, F., 1990. Temporal and spatial characteristics of annual precipitation in the twentieth century. *Int. J. Climatol*. 10: 495-504.

Martinovič J., Kuchař, S., 2010. Multiple scenarios computing in the flood prediction system FLOREON+. *24th European Conference on Modelling and Simulation, Kuala Lumpur*.

Miller, M., Stoytchev, A, 2008. Hierarchical Voting Experts: An Unsupervised Algorithm for Hierarchical Sequence Segmentation. *Proceedings of the 7th IEEE International Conference on Development and Learning (ICDL)*.

Miller M., Wong P., and Stoytchev A., 2009. Unsupervised Segmentation of Audio Speech Using the Voting Experts Algorithm. *Proceedings of the Second Conference on Artificial General Intelligence (AGI)*.

Muller, M., 2007. Dynamic Time Warping. *Information Retrieval for Music and Motion*, Springer, ISBN 978-3-540-74047-6, 69--84.

Watson, I., 1997. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*, 15-38.