StuCoSReC

Proceedings of the 2018 5th Student Computer Science Research Conference

University of Primorska Press

StuCoSReC
Proceedings of the 2018 5th Student Computer Science Research Conference

Edited by
Iztok Fister jr. and Andrej Brodnik

Published by
University of Primorska Press
Titov trg 4, si-6000 Koper

Editor-in-Chief
Jonatan Vinkler
Managing Editor
Alen Ježovnik

Koper, 2018

ISBN 978-961-7055-26-9 (pdf)
www.hippocampus.si/ISBN/978-961-7055-26-9.pdf
ISBN 978-961-7055-27-6 (html)
www.hippocampus.si/ISBN/978-961-7055-27-6/index.html
DOI: https://doi.org/10.26493/978-961-7055-26-9

Kataložni zapis o publikaciji (CIP) pripravili v Narodni in univerzitetni knjižnici v Ljubljani
COBISS.SI-ID=296769024
ISBN 978-961-7055-26-9 (pdf)
ISBN 978-961-7055-27-6 (html)

Preface

It is no longer necessary to emphasize that computer science is omnipresent today. Terms as digitalization, Industry 4.0 etc. are frequently heard in the mass media and are becoming a part of our everyday life. The rapid advances in computer science require a large investment in research and development. In particular, it is important to educate young scientists, which is one of the objectives of the StuCoSReC conference.

In front of you is the proceedings of the fifth StuCoSReC conference. The StuCoSReC is intended for masters & PhD students to show their research achievements. Socializing is also important and the conference is an excellent opportunity for students to get to know each other. The conference connects students of computer science and goes beyond the invisible limits of faculties. The uniqueness of StuCoSReC conference is that it is organized each year by different higher education institution. The University of Ljubljana - Faculty of Computer and Information Science is proud to host the conference this year.

Twelve papers addressed this conference, covering several topics of the computer science. All the papers were reviewed by two international reviewers and accepted for the oral presentation.

ref. prof. dr. Nikolaj Zimic

Contents

Preface II

Papers

Regulacija plovca v zračnem tunelu (zračna levitacija) na osnovi mehke logike 5–13
- Primož Bencak, Dominik Sedonja, Andrej Picej, Alen Kolman, Matjaž Bogša

Razpoznavanje literature v dokumentih in določanje njihovih sklicev v besedilu 15–19
- Matej Moravec, David Letonja, Jan Tovornik, Borko Boškovič

Klasifikacija samomorilskih pisem 21–25
- Dejan Rupnik, Denis Ekart, Gregor Kovačevič, Dejan Orter, Alen Verk, Borko Boškovič

Načrtovanje poprocesorja za pretvorbo NC/ISO G-kode v translacije za 3/5 osnega robota ACMA ABB XR701 27–33
- Gregor Germadnik, Timi Karner, Klemen Berkovič, Iztok Fister

Using constrained exhaustive search vs. greedy heuristic search for classification rule learning 35–38
- Jamolbek Mattiev, Branko Kavšek

Real-time visualization of 3D digital Earth 39–42
- Aljaž Jeromel

Towards an enhanced inertial sensor-based step length estimation model 43–46
- Melanija Vezočnik, Matjaž Branko Jurič

Breast Histopathology Image Clustering using Cuckoo Search Algorithm 47–54
- Krishna Gopal Dhal, Iztok Fister Jr., Arunita Das, Swarnajit Ray, Sanjoy Das

Time series classification with Bag-Of-Words approach post-quantum cryptography 55–59
- Domen Kavran

The problem of quantum computers in cryptography and post-quantum cryptography 61–64
- Dino Vlahek

Named Entity Recognition and Classification using Artificial Neural Network 65–70
- Luka Bašek, Borko Boškovič

Context-dependent chaining of heterogeneous data 71
- Rok Gomišček, Tomaž Curk

Time series classification with Bag-Of-Words approach

Domen Kavran
Faculty of Electrical Engineering and Computer Science
Koroska cesta 46
Maribor, Slovenia
domen.kavran@um.si

ABSTRACT

The amount of data generated every day increases each year and the pace is accelerating with development of Internet of Things (IoT). Gathered data solely doesn't contain much information, but with machine learning additional informations and hidden patterns can be obtained to contribute to time series analysis.

General purpose and problem specific time series feature extraction methods have been developed over the years. New feature extraction approach, derived from Bag-Of-Words, is presented in this paper. Main part of the approach is obtaining a dictionary of time series segments – the so-called words. K-Means clustering is used to form a dictionary containing K words, which is then used to define a feature vector of an individual time series as a histogram of word occurances inside it. Described approach can be used for feature extraction of time series without prior knowledge of data's nature. Moreover, the approach is robust and produces good classification results. Highest accuracy of 99.96% was achieved using datasets, presented in Results.

Keywords

time series, classification, machine learning, bag-of-words

1. INTRODUCTION

Technological progress made in industries, like automotive, healthcare and electronics, over the past years resulted in increased amount of produced data. Large complex datasets, often referred to as Big Data, must be analysed quickly and efficiently to reveal additional informations and hidden patterns. Data analysis aids companies with their product development, research and customer services [5]. Different technologies and programming libraries have been developed to help engineers in creating pipelines for manipulating data and extracting features. Unsupervised machine learning algorithms are often used for analysing data and finding correlation between variables. Various advanced supervised techniques have been developed to solve regression and classification tasks. Time series data have an important role in today's largest industries and this paper presents a general purpose time series feature extraction approach.

Broader interest in time series classification began at the start of the century [9], which led to development of many different feature extraction methods [16]. For machine learning algorithms to be successful at classifying time series, appropriate features must be selected. Features can be obtained through statistical analysis or by calculating features specifically selected for a given dataset, though expert knowledge is needed. Presented alternative approach, derived from Bag-Of-Words [17], needs no prior knowledge about provided data. Main part is the definition of dictionary words – segments, which are extracted from training time series data and later clustered. Segments of individual time series are then compared with dictionary words and histogram of word occurances is formed. Histogram is a feature vector representing individual time series.

Bag-Of-Words approach was originally intended for document and image classification. Same concepts can be applied to time series with great results. Presented approach uses elements of well-known image patch extraction algorithm to obtain overlapping windows or segments, containing parts of time series data. To speed up the training phase and classification process, discrete wavelet transform was used. The transform is known for its role in data compression and image processing [3]. That provided a suitable way of reducing each segment's feature vector dimensionality. Mini Batch K-means clustering was used to speed up the dictionary creation process.

In second section of the paper, procedure of feature extraction is described. Classification was done by 1-nearest neighbor algorithm, using Chi-squared distance measure and then compared with the results of support vector machines and random decision forests in Results.

2. TIME SERIES CLASSIFICATION

Classification pipeline is shown on Figure 1. Time series must be described with features, which appropriately present occured events in time series and are independent of its length. Beforehand, input time series dataset has to be split on training and test datasets. Words inside time series are segments feature vectors, with features being approximation coefficients of discrete wavelet transform. Dictionary words are the result of clustering segments feature vectors, extracted from training time series set. Feature vector of each time series is the created histogram of dictionary words occurances.

Original approach has a name Bag-Of-Words for classifying documents and Bag-Of-Features for image classification [6]. Advantage of the approach is its robustness, simplicity and good results on real-world data.

DOI: https://doi.org/10.26493/978-961-7055-26-9.55-59

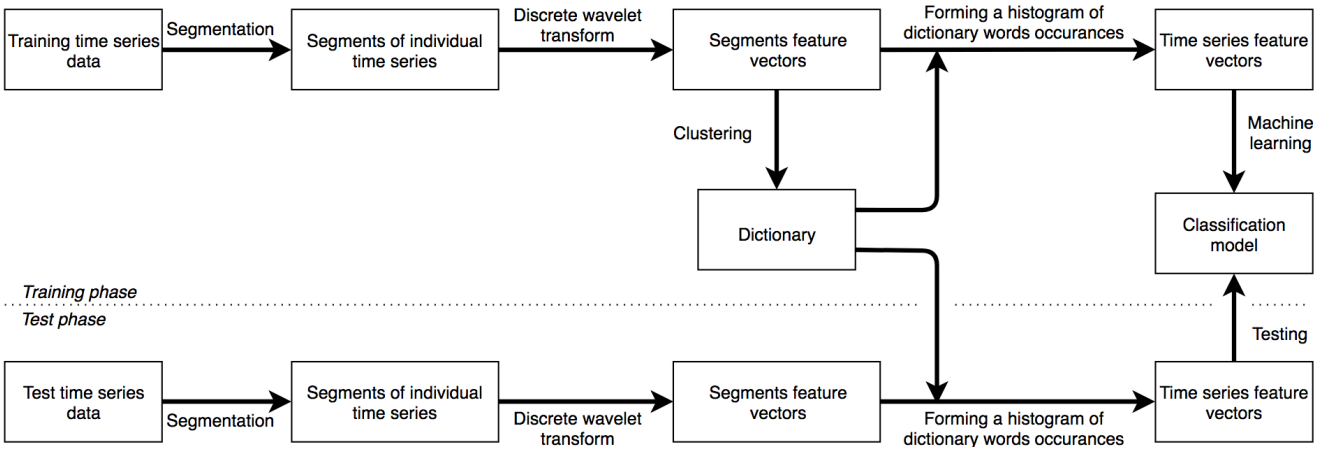

Figure 1: Classification pipeline

Histograms are often used in natural language processing and computer vision problems, because local informations of observed object are presented. Those characteristics justify use of histograms as time series feature vectors.

2.1 Extraction of segments

2.1.1 *Segmentation with sliding window algorithm*

Both training and test time series are split into overlapping segments with sliding window of length W_c and step length $t_c < W_c$.

Example of segments extraction is shown on Figure 2. Time series of length 3072 values is split onto segments of length $W_c = 256$ with step $t_c = 192$. Segments are shown in orange color with grey overlapping parts. In most cases, individual segment's length should not exceed 10% of average time series length, although this percantage can differ between datasets. Selected window length has a big impact on classification accuracy and should be carefully selected.

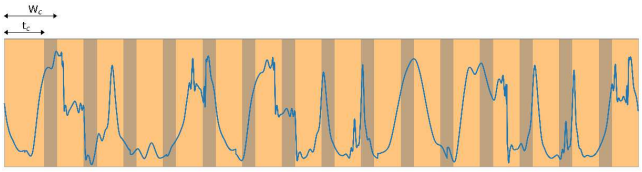

Figure 2: Segmentation of a time series

2.1.2 *Feature extraction*

Every segment is described with features being approximation coefficients of discrete wavelet transform (DWT) function for a selected decomposition level. With every increase of decomposition level, number of approximation coefficients is reduced by nearly a factor of two, but enough information is preserved to reconstruct original data. With selected transform, frequency informations at lower frequencies are analyzed and time informations in higher frequencies are gathered [17]. Additional benefit of the transform is data noise reduction. Db3 discrete wavelet transform function at first decomposition level was used to define segment's feature vector. That results in almost half of segment's length number of approximation coefficients [17]. Figure 3 shows application of discrete wavelet transform on a time series.

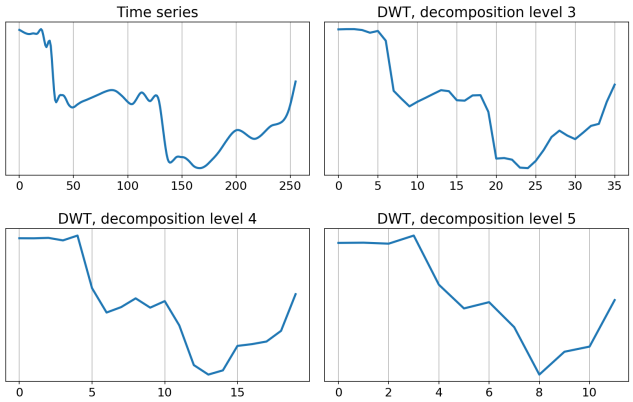

Figure 3: Approximation coefficients of applied db3 discrete wavelet transform with different decomposition levels

2.2 Dictionary words

K-means clustering has been applied on a set of segments feature vectors, extracted from training time series in previous step. Result of clustering is K number of clusters with centroids being dictionary words. In terms of speed, K-means algorithm is not suitable for large datasets. Because of the latter fact, alternative algorithm Mini Batch K-means, was used. Algorithm uses subset of the dataset per iteration and, consequently, less computations are needed.

For visualization purposes, conversion of segments multidimensional feature vectors into three dimensional was done, using principal component analysis (PCA). Mathematical transformation converts large number of potentially dependant variables to fewer independant variables – principal components, which hold the most information. Applying principal component analysis is one of the first steps in analysis of large, multivariate datasets [13]. Result of clustering PCA output is visible on Figure 4 with individual cluster being presented in unique color and have a centroid marked with a black dot.

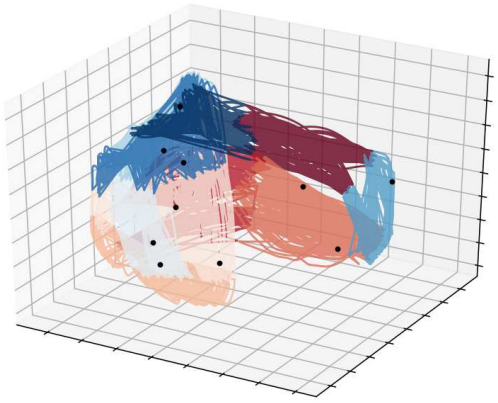

Figure 4: Vizualization of ten clusters as a result of applying clustering on first three principal components of training segments feature vectors

Recommended size of dictionary is ranging from 1000 to 3500 words. The reason is that classification done with fewer number of words doesn't reach the highest accuracy possible, but it starts to stabilize with a dictionary size above 1000 words [17].

2.3 Time series features

Defining time series features starts with segmentation and feature extraction, described in Chapter 2.1. Segments feature vectors are compared with dictionary words using 1-nearest neighbors algorithm using Euclidean distance as distance measure. Histogram of dictionary word occurances is created and normalized with L^2-norm. Example of time series histogram is shown in Figure 5.

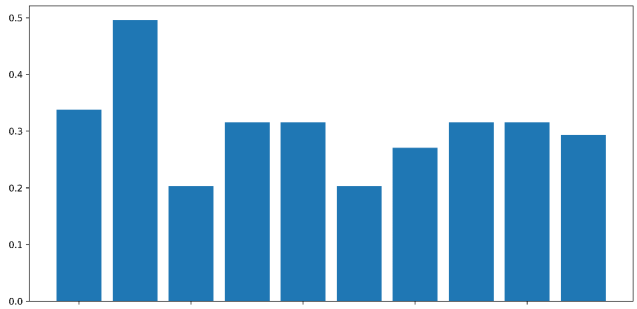

Figure 5: Normalized histogram based on a dictionary of ten words

Training time series, described with feature vectors, are used to train a classification model. For classification model testing, time series from test dataset are used.

2.4 Classification algorithms

Training classification models was done using the following described classification models.

2.4.1 K-Nearest neighbors

K-Nearest neighbors (KNN) algorithm classifies a sample based on distance between training samples and unclassified sample. Euclidean distance is usually used as a distance measure and label is determined based on K nearest training samples. Odd number should be selected for parameter K value, because classification label decision-making is done by majority vote [11]. Selection of parameter K value has a big effect on final classification accuracy. The best approach to select appropriate parameter K is by trial and error. Time complexity of the algorithm is large, because most computational operations are being done during classification and not in training phase [10].

Chi-Square distance measure [18] was used, because it can measure differences between histograms. It is frequently used in classification of textures, objects and shapes. Chi-square distance measure takes distribution of values and their frequencies into account and also has an import characteristic of weighing rarely occured values in histogram [1]. Equation of Chi-square distance measure χ is

$$\chi^2(x_i, x_j) = \frac{1}{2}\sum_{k=1}^{d}\frac{(x_{ik} - x_{jk})^2}{x_{ik} + x_{jk}} \quad (1)$$

where:

x = feature vector
i, j = feature vectors indices
d = length of feature vectors
k = feature index

2.4.2 Support vector machine

Algorithm is a part of linear classifiers group. It forms an optimal hyperplane to maximize distance between parallel planes, placed on outer boundaries of training samples feature vectors, based on their labels [12, 15]. Support vector machine (SVM) is a binary classifier, but can also be used for multilabel classification. Multiple binary classifiers are formed, with i-th label being treated as positive and the rest as negative. Described approach is called One-Vs-All [14]. Alternative approach has a name One-Vs-One, where a binary classifier is formed for every pair of labels. One-Vs-One approach is useful at dealing with unbalanced datasets, but that comes at a cost of higher time complexity during training and classification process [8].

2.4.3 Random forest

Training starts by forming a set of decision trees with induced randomness. A sample is classified based on majority vote of decision trees [7]. Number of trees, their depth, minimal number of training samples to split a node and minimal number of samples required to be at leaf node must be adjusted to optimize classification accuracy. Random forest is a fast learning and general-purpose classification algorithm, which is resistant to overfitting at large number of features [4]. To achieve high accuracy, large set of decision trees has to be created, but that also means slower classification process.

3. RESULTS

Presented feature extraction approach was implemented in Python using scientific computing library NumPy. Machine learning library scikit-learn was used for clustering and classification. For discrete wavelet transform, open-source soft-

ware PyWavelets was used. Training and testing was done on the following computer hardware:

- Intel Core i7-6700K processor
- 16GB of DDR4 3200MHz memory

Various open-source time series datasets [2] were used for testing and are described in Table 1. Each dataset contains time series of equal length and split in two classes or more. Testing was done using 5-fold cross-validation with F1 score as classification accuracy measurement for all classification algorithms, presented in Chapter 2.4. Dictionaries containing 2000 words were used. Parameters W_c and moving step t_c were choosen individually for every time series. The reason being is that their lengths differ and experimenting with different parameter values was needed to reach best possible classification score. Classification results are presented in Table 2.

Table 1: Time series descriptions

Time series	Description
CBF	Simulated time series
ECG5000	ECG signal of a patient having a heart failure
FordA	Engine sounds
Mallat	Simulated time series
ShapesAll	Binary images contours distances from center
UWaveGestureLibraryAll	Accelerometer data generated during hand movement
StarlighCurves	Brightness of astronomical objects through time
Wafer	Measurements recorded from sensors during the processing of silicon wafers

Highest average classification F1 score 91.82% was achieved with SVM classification model, followed by random forest and 1-nearest neighbor using Chi-square distance measure. Biggest difference in classification results can be seen at UWaveGestureLibraryAll, where 1-NN classifier achieved F1 score of 25.53%, opposed to SVM's score of 80.64%. F1 scores of all classification models are closest at classifying time series CBF, with a difference of only 0.75% between them.

4. CONCLUSION

Highest achieved classification accuracy on test datasets was 99.96%, with average across all three classification models being 85.75%. Feature extraction approach is appropriate for use in various industry related problems and at analysis of stock market, but medical use is questionable, because achieved accuracy must be at highest level possible to ensure medical safety. Best such case are classification scores for time series ECG5000, where highest achieved F1 score is 86.89%, much lower than satisfying for critical medical data.

Parallel computing can be used for extraction of segments and at defining time series feature vectors. Presented feature extraction approach is appropriate for integration in real-time systems and IoT devices, however training phase should be done separately from working environment. Higher classification accuracy could be achieved with use of multivariate time series.

5. REFERENCES

[1] V. Asha, N. U. Bhajantri, and P. Nagabhushan. GLCM based Chi-square Histogram Distance for Automatic Detection of Defects on Patterned Textures. *International Journal of Computational Vision and Robotics*, 2(4):302–313, 2011.

[2] A. Bagnall, J. Lines, W. Vickers, and E. Keogh. The UEA and UCR Time Series Classification Repository. Available at `www.timeseriesclassification.com`.

[3] P.M. Bentley and J.T.E. McDonnell. Wavelet transforms: an introduction. *Electronics and Communication Engineering Journal*, 6(4):175–186, 1994.

[4] G. Biau. Analysis of a Random Forests Model. *Journal of Machine Learning Research*, 13:1063–1095, 2012.

[5] Thomas H. Davenport. *Big Data at Work*. Harvard Business Review Press, 2014.

[6] F. Fang, H. Lu, and Y. Chen. Bag of Features Tracking. *International Conference on Pattern Recognition*, 2010(46):153–156, 2010.

[7] K. Fawagreh, M. Gaber, and E. Elyan. Random forests: from early developments to recent advancements. *Systems Science and Control Engineering*, 2(1):602–609, 2014.

[8] A. Gidudu, G. Hulley, and T. Marwala. Image classification using SVMs: One-Against-One vs One-Against-All. *Clinical Orthopaedics and Related*

Table 2: Classification F1(%) results

Time series	No. of classes	Parameters	1-Nearest neighbor, Chi-square distance	SVM, polynomial kernel, degree = 2	Random forest
CBF	3	W_c=50, t_c=5	99.14	99.89	99.35
ECG5000	5	W_c=10, t_c=5	77.91	86.89	84.18
FordA	2	W_c=90, t_c=5	70.41	89.96	81.92
Mallat	8	W_c=10, t_c=5	98.78	99.96	97.08
ShapesAll	60	W_c=100, t_c=5	83.31	80.14	62.19
UWaveGestureLibraryAll	8	W_c=50, t_c=5	25.53	80.64	57.86
StarlightCurves	3	W_c=220, t_c=5	91.47	97.56	96.89
Wafer	2	W_c=10, t_c=5	98.52	99.54	99.04
F1 average	/	/	80.63	91.82	84.81

Research, 0711.2914, 2007.

[9] Juan J. Rodr Guez and Carlos J. Alonso. Boosting Interval-Based Literals: Variable Length and Early Classification. *Series in Machine Perception and Artificial Intelligence*, 57:149–171, 2002.

[10] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer. KNN Model-Based Approach in Classification. *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, pages 986–996, 2003.

[11] E. M. Kamel, M. Wafy, H. M. Ibrahim, and I. A. Badr. Comparison of different classification algorithms for certain weed seeds' species and wheat grains identification based on morphological parameters. *IJCSI International Journal of Computer Science Issues*, 12(5):110–116, 2015.

[12] G. Oreški and S. Oreški. An experimental comparison of classification algorithm performances for highly imbalanced datasets. *Central European Conference on Information and Intelligent*, pages 4–11, 2014.

[13] M. Richardson. Principal Component Analysis. 2009.

[14] R. Rifkin and A. Klautau. In Defense of One-Vs-All Classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

[15] D. Srivastava and L. Bhambhu. Data Classification using support vector machine. *Journal of Theoretical and Applied Information Technology*, 12(1), 2005.

[16] Michele A. Trovero and Michael J. Leonard. Time Series Feature Extraction. 2018.

[17] J. Wang, P. Liu, M. F. H. She, S. Nahavandi, and A. Kouzani. Bag-of-words representation for biomedical time series classification. *Biomedical Signal Processing and Control*, 8(6):634–644, 2013.

[18] W. Yang, L. Xu, X. Chen, F. Zheng, and Y. Liu. Chi-Squared Distance Metric Learning for Histogram Data. *Mathematical Problems in Engineering*, 2015:1–12, 2015.

StuCoSReC

University of Primorska Press
www.hippocampus.si
ISBN *978-961-7055-26-9*