

USB Proceedings

The 19th International Conference on

Modeling Decisions for Artificial Intelligence

MDAI 2022, Sant Cugat

Vicenç Torra, Yasuo Narukawa, Jordi Nin, Núria Agell



Image from wikipedia.org

USB Proceedings

The 19th International Conference on

Modeling Decisions for Artificial Intelligence

MDAI 2022, Sant Cugat, Catalonia
30 August - 2 September, 2022

Editors:

Vicenç Torra
Umeå University
Umeå, Sweden
E-mail: vtorra@ieee.org

Yasuo Narukawa
Department Management Science
Tamagawa University
Tokyo
Japan
E-mail: nrkwy@eng.tamagawa.ac.jp

Jordi Nin
ESADE
Department of Operations, Innovation and Data Sciences
Av. Torreblanca, 59
E-08172 Sant Cugat
E-mail: jordi.nin@esade.edu

Núria Agell
ESADE
Department of Operations, Innovation and Data Sciences
Av. Torreblanca, 59
E-08172 Sant Cugat
E-mail: nuria.agell@esade.edu

ISBN: 978-91-527-4611-0

Preface

This volume contains papers that had to be presented at the 19th International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2022), in Sant Cugat, August 30th - September 2nd, 2022. The rest of papers as well as invited papers have been separately published in the Lecture Notes in Artificial Intelligence, Vol. 13408 (by Springer).

This conference followed MDAI 2004 (Barcelona), MDAI 2005 (Tsukuba), MDAI 2006 (Tarragona), MDAI 2007 (Kitakyushu), MDAI 2008 (Sabadell), MDAI 2009 (Awaji Island), MDAI 2010 (Perpinyà), MDAI 2011 (Changsha), MDAI 2012 (Girona), MDAI 2013 (Barcelona), MDAI 2014 (Tokyo), MDAI 2015 (Skövde), MDAI 2016 (Sant Julià de Lòria), MDAI 2017 (Kitakyushu), MDAI 2018 (Mallorca), MDAI 2019 (Milano), MDAI 2020, and MDAI 2021 (Umeå).

The aim of MDAI is to provide a forum for researchers to discuss different facets of decision processes in a broad sense. This includes model building and all kinds of mathematical tools for data aggregation, information fusion, and decision-making; tools to help make decisions related to data science problems (including, e.g., statistical and machine learning algorithms as well as data visualization tools); and algorithms for data privacy and transparency-aware methods so that data processing procedures and the decisions made from them are fair, transparent, and avoid unnecessary disclosure of sensitive information.

The MDAI conference included tracks on the topics of (a) data science, (b) machine learning, (c) data privacy, (d) aggregation functions, (e) human decision-making, (f) graphs and (social) networks, and (g) recommendation and search.

The conference was supported by ESADE-Institute for Data-Driven Decisions (esadeD3), the European Society for Fuzzy Logic and Technology (EUSFLAT), the Catalan Association for Artificial Intelligence (ACIA), the Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT), and the UNESCO Chair in Data Privacy.

Vicenç Torra, Yasuo Narukawa
June, 2022

General Chairs

Jordi Nin, ESADE, Universitat Ramon Llull

Program Chairs

Vicenç Torra, Umeå University, Sweden
Yasuo Narukawa, Tamagawa University, Japan

Advisory Board

Didier Dubois, Institut de Recherche en Informatique de Toulouse, CNRS, France
Jozo Dujmović, San Francisco State University, USA
Lluís Godó, IIIA-CSIC, Spain
Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences, Poland
Sadaaki Miyamoto, University of Tsukuba, Japan
Pierangela Samarati, Università degli Studi di Milano, Italy
Sandra Sandri, Instituto Nacional de Pesquisas Espaciais, Brazil
Michio Sugeno, Tokyo Institute of Technology, Japan
Ronald R. Yager, Machine Intelligence Institute, Iona College, USA

Program Committee

Kayode S. Adewole, Umeå University, Sweden
Laya Aliahmadipour, Shahid Bahonar University, Iran
Cláudia Antunes, Universidade de Lisboa, Portugal
Eva Armengol, IIIA-CSIC, Spain
Edurne Barrenechea, Universidad Pública de Navarra, Spain
Gloria Bordogna, Consiglio Nazionale delle Ricerche, Italy
Humberto Bustince, Universidad Pública de Navarra, Spain
Alina Campan, North Kentucky University, USA
Francisco Chiclana, De Montfort University, UK
Susana Díaz, Universidad de Oviedo, Spain
Josep Domingo-Ferrer, Universitat Rovira i Virgili, Spain
Yasunori Endo, University of Tsukuba, Japan
Vladimir Estivill-Castro, Griffith University, Australia
Zoe Falomir, Universitat Jaume I, Spain
Javier Fernandez, Universidad Pública de Navarra, Spain
Giorgos Flouris, FORTH-ICS, Greece
Camilo Andres Franco De Los Rios, Universidad de los Andes, Colombia
Katsushige Fujimoto, Fukushima University, Japan
Joaquín García-Alfaro, Institut Mines-Télécom and Institut Polytechnique de

Paris, France
Michel Grabisch, Université Paris I Panthéon-Sorbonne, France
Yukihiro Hamasuna, Kindai University, Japan
Tove Helldin, University of Skövde, Sweden
Enrique Herrera-Viedma, Universidad de Granada, Spain
Aoi Honda, Kyushu Institute of Technology, Japan
Van-Nam Huynh, JAIST, Japan
Masahiro Inuiguchi, Osaka University, Japan
Simon James, Deakin University, Australia
Aránzazu Jurío, Universidad Pública de Navarra, Spain
Yuchi Kanzawa, Shibaura Institute of Technology, Japan
Sema Kayapinar Kaya, Munzur University, Turkey
Hiroaki Kikuchi, Meiji University, Japan
Petr Krajča, Palacky University Olomouc, Czech Republic
Marie-Jeanne Lesot, Université Pierre et Marie Curie (Paris VI), France
Giovanni Livraga, Università degli Studi di Milano, Italy
Jun Long, National University of Defense Technology, China
Beatriz López, University of Girona, Catalonia, Spain
Jean-Luc Marichal, University of Luxembourg, Luxembourg
Michael Mayo, University of Waikato, New Zealand
Radko Mesiar, Slovak University of Technology, Slovakia
Andrea Mesiarová-Zemánková, Slovak Academy of Sciences, Slovakia
Anna Monreale, University of Pisa, Italy
Pranab K. Muhuri, South Asian University, India
Toshiaki Murofushi, Tokyo Institute of Technology, Japan
Guillermo Navarro-Arribas, Universitat Autònoma de Barcelona, Spain
Tomaharu Nakashima, Osaka Prefecture University, Japan
Shekhar Negi, Umeå University, Sweden
Miguel Nunez-del-Prado, Universidad del Pacífico, Peru
Anna Oganyan, National Institute of Statistical Sciences (NISS), USA
Gabriella Pasi, Università di Milano Bicocca, Italy
Oriol Pujol, University of Barcelona, Catalonia, Spain
Maria Riveiro, Jönköping University, Sweden
Julian Salas, Universitat Oberta de Catalunya, Catalonia, Spain
H. Joe Steinhauer, University of Skövde, Sweden
László Szilágyi, Sapientia-Hungarian Science University of Transylvania, Hungary
Aida Valls, Universitat Rovira i Virgili, Spain
Paolo Viappiani, Université Paris Dauphine, France
Zeshui Xu, Southeast University, China

Local Organizing Committee Chair

Núria Agell, ESADE, Universitat Ramon Llul, Catalonia, Spain

Additional Referees

Ashneet Khandpur Singh, Sergio Martinez Lluís, Fabio Stella, Rami Haffar, Najeeb Moharram Salim Jebreel

Supporting Institutions

ESADE-Institute for Data-Driven Decisions
The European Society for Fuzzy Logic and Technology (EUSFLAT)
The Catalan Association for Artificial Intelligence (ACIA)
The Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)
The UNESCO Chair in Data Privacy

Table of Contents

Regular Papers

Anomaly Detection leveraging Dimensionality Reduction of Multivariate Spatio-Temporal Data using Pre-attentive Features	1
<i>Rui Maia, Cláudia Antunes</i>	
Detecting Mild Cognitive Impairment Using Fisher Vectors	11
<i>José Vicente Egas-López, Réka Balogh, Nóra Imre, Ildikó Hoffmann, László Tóth, Magdolna Pákáski, János Kálmán, Gábor Gosztolya</i>	
Null Additivization of a Monotone Measure on a Finite Set	21
<i>Ryoji Fukuda, Aoi Honda, Yoshiaki Okazaki</i>	
DDoS Attack Detection in Software Defined Networks using Convolutional Tsetlin Machine	32
<i>Rasoul Jafari Gohari, Laya Aliahmadipour, Marjan Kuchaki Rafsanjani</i>	
Using machine learning and natural language processing for fake news detection in WhatsApp	43
<i>Leonardo Miranda de Brito, Marília Caroline de A. C., Silva, Natália Oliveira, Tuby D. Neto, Nicksson Ckayo Arrais de Freitas, Tiago Da Silva Vinuto</i>	
DANKFE: a domain knowledge based algorithm for feature engineering ...	55
<i>Tiago Afonso, Cláudia Antunes</i>	
Parametric Extension of q-Divergence-based Fuzzy Clustering for Spherical Data	68
<i>Yuchi Kanzawa</i>	
Aggregation for Hesitant Fuzzy sets	79
<i>Yasuo Narukawa, Vicenç Torra</i>	

Anomaly Detection leveraging Dimensionality Reduction of Multivariate Spatio-Temporal Data using Pre-attentive Features

Rui Maia¹[0000–0003–2100–3203] and Cláudia Antunes²[0000–0002–9467–2515]

¹ Instituto Superior Técnico, Avenida Rovisco Pais 1, Lisboa, 1049-001, Portugal
rui.maia@tecnico.ulisboa.pt

² Instituto Superior Técnico, Avenida Rovisco Pais 1, Lisboa, 1049-001, Portugal
claudia.antunes@tecnico.ulisboa.pt

Abstract. The growing number of spatio-temporal enabled systems leverage the interest in mining data, whether for maritime and land traffic management or to improve human safety. Different complexities characterize these systems by the evolving characteristics of data sensors, natural and human introduced noise, legal restrictions or domain specificities such as weather and sea conditions. While someone can easily interpret space and time dimensions in human reasoning patterns, it is still a hard task to use the potential of these referential dimensions along with other categorical or real-valued dimensions in computational tasks. In this work, we propose a new symbolic representation of multivariate spatio-temporal datasets supported by the research on human reasoning and cognitive capabilities. The proposed approach uses pre-attentive features in order to transform complex spatio-temporal multivariate information into highly representative two-dimensional images. By experimenting with a real world Automatic Identification Systems dataset in a binary classification task using Convolutional Neural Networks as a classifier, we prove the effectiveness of our approach as a new multivariate spatio-temporal data representation.

Keywords: Dimensionality, Multivariate, Temporal, Spatial, Image, Perception

1 Introduction

Multivariate temporal data series are nowadays being collected around the globe in very different contexts such as networks of radars, maritime or land sensors, weather and climate monitoring devices or medical sensors. The huge amount of available data can be very different in terms of dimensions, noise, class balance, consistency or completeness. These differences are examples of the complexities that must be dealt with when solving a particular problem in a specific domain.

In order to apply computational approach to a determined task, either to classify observations or forecast a natural system behaviour, data is prepared applying domain knowledge rules and generic rules, applicable to most of the

problems. It has also been underlined [10] that there is an over-the-top usage of data subsets that are more likely to prove a certain idea.

The specific case of Multivariate Spatio-Temporal data includes not only the most common data complexities but also the ones related with the presence of two referential dimensions: space and time. The information on these two specific dimensions is easily acquired by the cognitive capabilities of humans, who can visually identify different data patterns when this data is represented by images. Although, humans cannot understand images representing a large number of dimensions, which is a limitation that computational approaches try to tackle although lacking the advantage human cognition on the referential dimensions.

Several symbolic representation approaches have been published, but they frequently maintain the number of dimensions [12] which is still seen as challenge to different mining algorithms do not scale properly. By identifying this gap between human capabilities and the state-of-art-knowledge on series symbolic representation and image processing, we propose a new method to represent multivariate spatio-temporal information in rich images. Each of these images is generated using a complete series and all the available dimensions, real-valued or categorical. The proposed method can be seen as a dimensionality reduction technique, from a multi dimensional to a two-dimensional space.

The next section of this paper presents the related work, followed by the introduction of relevant visual perception concepts supporting our method in section 3. Section 4 details the proposed approach while the experimental process and results analysis are done in Section 5. Finally conclusions, are drawn in Section 6.

2 Related Work

Multivariate temporal analysis still encompasses relevant challenges as underlined by Moskovitch [13]. The research on multivariate time series representation has been addressed by authors on different perspectives. Lin et al. [12] proposed the Symbolic Aggregate Approximation (SAX) to convert the result of a Piecewise Aggregate Approximation (PAA) [9] into a symbol string. The series space is split into equally sized number of regions (both predefined), each one identified by a different symbol. Each segment of a series is then mapped into a symbol, corresponding to the region in which it resides, which together builds a string of symbols that represent the complete series.

Baydogan and Runner [3] proposed Symbolic Multivariate Time Series (SMTS) as an approach to represent series data by the individual attributes and their relationships using a letter codebook. It is a dimensionality reduction technique based on a Random Forest approach that will partition the series into leaf nodes, each one represented by a letter. The words are latter used to classify new series. The same authors later proposed a similar approach, the Learned Pattern Similarity (LPS) [4]. Segments are extracted from a multivariate time series in order to train Regression Trees that find relations and dependencies between them.

Each node is also represented by a word, and the complete series is represented by the collection of leaf node words.

Recently, Bei et al. [2] noted limitations on SAX representation since it only extracts the mean feature of subsequences to build the symbolic representation. These authors proposed a new ensemble method called TBOPE based on SAX. It leverages time series mean feature and trend histograms to represent the complete series.

Frequent patterns in time series, typically identified through the analysis of a fixed size rolling window, are called Motifs and support different symbolic approaches. After the identification of motifs in a set of time series it is possible to verify which specific motif is part of a series by calculating the distance between the motif and a segment of a series, or, by converting the series into subsequences and comparing each subsequence with the identified motifs. They can be identified in a single series or in the complete series dataset. They can be contiguous, non-contiguous and multigranular, when they might occur in different window sizes. Aggarwal [1] underlined that 2-dimensional motifs (or patterns) can be useful for image processing when considering a spatial scenario. Nonetheless, to the extent of our knowledge, no works have been published on the representation of complete multivariate series in a two-dimensional image space.

Several other approaches to time series representation can be found in literature, for example, Fourier Transformations, Wavelets, Polynomial Models, or Shapelets.

3 Perception Background Knowledge

In the context of computing tasks, multivariate series data preparation and representation raise relevant complexities. Besides the real-valued and categorical dimensions, data series may include spatio-temporal dimensions, increasing the overall process complexity since these are considered as reference dimensions in almost any domain.

Charts or plots are frequently used to represent information in human friendly form. Our cognitive system can interpret these graphical representations of information being also able to naturally express in the same terms, manually or using computer tools. This interpretation process consists in the decoding of both categorical and quantitative information from images [6]. This process, pre-attentive processing [14], enables humans with a fast cognitive capability where a large number of visual attributes can instantly be interpreted. These attributes have been leveraging experiments [11] [5] on the representation of real-time multivariate data [8]. In these experimental research works, data series observations are represented using multiple pre-attentive visual attributes such as form, color, position and movement (see Figure 1).

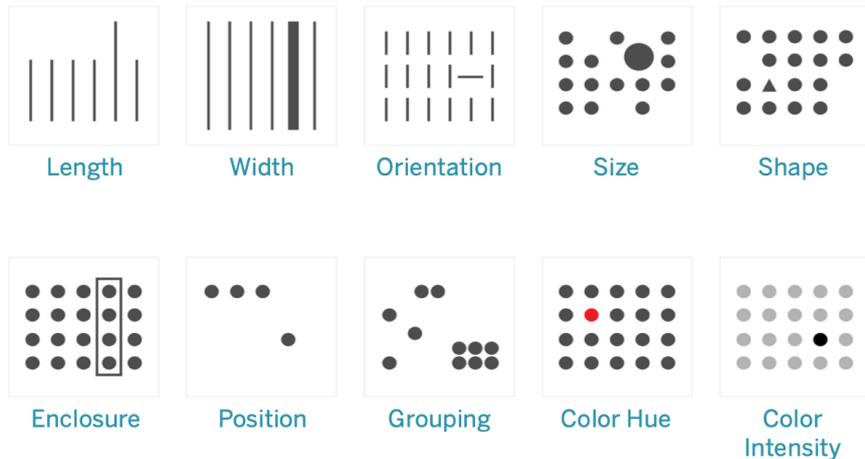


Fig. 1: Preattentive Visual Attributes.

4 Proposed Approach

This research builds on the possibility of generating images for multivariate temporal data that express the multitude of factors affecting the series behaviour. This framework is specific for spatio-temporal domain, where relevant dimensions are the ones regarding location, time, speed and course.

Figure 2 illustrated this three-phase process. It starts with (1) the generation of two-dimensional images using the complete data series of an object movement, i.e. the spatio-temporal and all other categorical and real-valued dimensions. The spatial features determine where each position is plotted in the two-dimensional space. Then, each mark depends on the time lag between observations and other feature values, which will affect the mark’s color, size, transparency or dimension. Therefore, regarding the pre-attentive features presented in Figure 1, this process includes: Position, Size, Color Hue and Color Intensity. The hyper-parameters were defined by testing and validation iterations. Algorithm 1 introduces the image generation process for each object data series (movement).

Obtained images are then (2) processed in a Convolutional Neural Network whose weights were previously trained in a classification task using the ImageNet [7] database, featuring 14,197,122 labelled images. We use only the feature extraction part of this network in order to get information about the image of an object movement. We used the VGG16 model for image recognition made available by the Visual Geometry Group of the University of Oxford. This model has 16 weight layers and a total of 138,357,544 parameters. Figure 3 illustrates the architecture of VGG16.

Finally, in order to validate our approach, we run a clustered based (3) unsupervised anomaly detection process over the complete dataset, including with

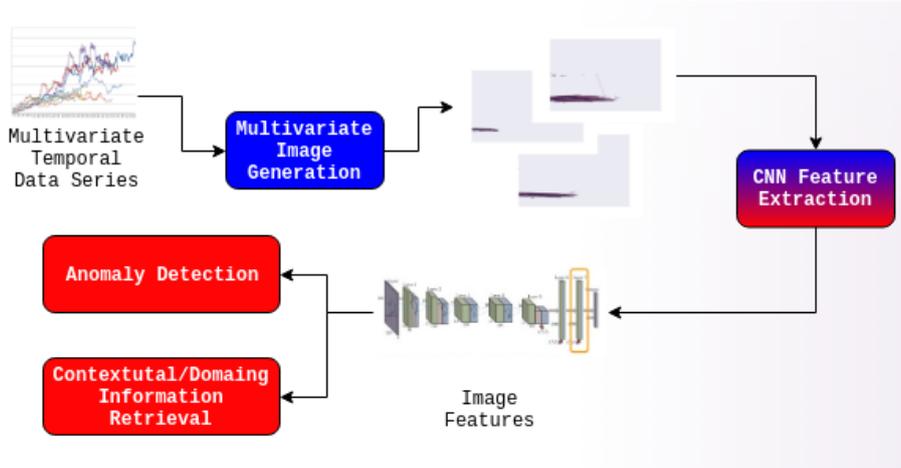


Fig. 2: Experimental framework for multivariate series image generation based anomaly detection.

all observation vectors. In parallel we are researching the possibility of retrieving information on the context or domain (not subject of this paper).

5 Experimental Analysis

This work's refers to the possibility of representing spatio-temporal multivariate data series in two-dimensional images. Having applicability in several domains, we chose to experiment in the maritime vessels Automatic Identification Systems (AIS) tracks. The similarities between this and other domains such as road traffic and public transportation is indisputable, as they both have referential spatio-temporal dimensions on the top of several other common object (vehicle) sensors information. They also have contextual similarities that may appear as relevant such as virtually identical notions of traffic lanes, crossing rules, priority passing, collision risk or coordinated movements.

Experiments were run over a freely available AIS dataset by National Oceanic and Atmospheric Administration (NOAA) of the United States of America (USA). In this dataset each observation consists of a multidimensional structure containing both real-valued and categorical features, including also spatial and time dimensions.

By representing tracks as images, and applying a clustering approach - for two clusters - we expected to prove the effectiveness of the approach. Because there is no information about abnormal samples or tracks, this experiment corresponds to an unsupervised anomaly detection task.

The dataset characteristics are summarized in the Table 1.

Figure 4 is an example of an image generated for a multivariate temporal series describing not only the location and movement but also other features

Algorithm 1: Multivariate Spatio-Temporal Series Image Generation

Data: List of Object Multivariate Spatio-Temporal Observations

Result: Object Movement High-Res Image

 α = Position Weight;

 β = Size Weight;

 γ = Color Hue;

 δ = Color Intensity (Transparency) ;

Circular Normalization of Latitude and Longitude ;

Normalization of Speed Over Ground and Course Over Ground ;

for $observation \in ListOfObjectObservations$ **do**

 observation Position = (Latitude and Longitude) * α ;

 observation Size = Default * $1/TimestampLag$ * $1/Speed\ Over\ Ground$ *

 β ;

 observation Color Hue = Course Over Ground * γ ;

 observation Color Intensity = $TimestampLag$ * δ ;

Plot observation

end

Table 1: Characterization of the experimented Dataset

Dataset	Samples	Dimensions	Real-Valued mensions	Di- mensions	Categorical	Di- mensions	Vessels	Count
USA (Zone 10)	279.773	8	Latitude, Longitude, Over Speed Ground,	Lon- Course Ground, Over	Vessel ID, Type,	38		

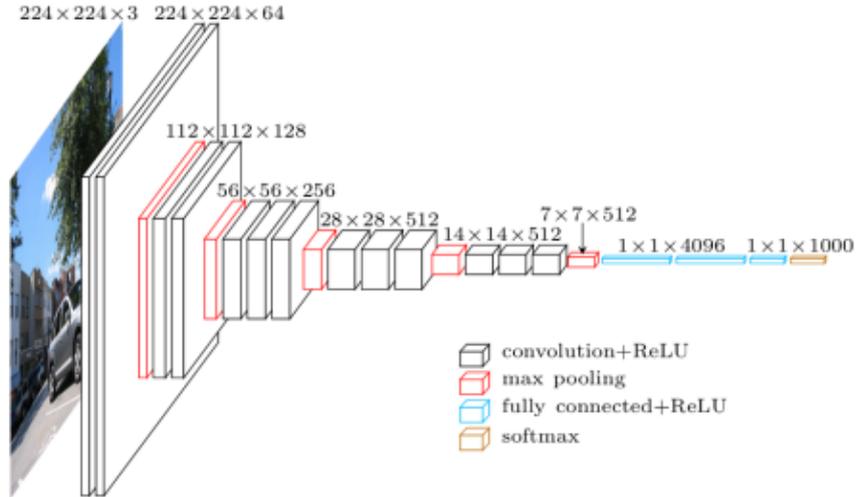


Fig. 3: VGG16 Architecture as published by the Visual Geometry Group (University of Oxford). Feature extraction layers are red coloured, ending at the layer with format $7 \times 7 \times 512$. The fully connect+ReLU and Softmax components are the classification components which are not used in this research.

influencing the series, namely: Latitude, Longitude, Timestamp, Course-over-Ground and Speed-over-ground.

Following the intuition that spatio-temporal data series heavily rely on positional features, similar positioned observations will appear near each others. Although, the detail of the series plot indicate very different signatures. Figure 5 zooms to the series detail.

The proposed multivariate spatio-temporal series image generation component, and the VGG16 model for feature extraction, showed to enable accurate results in the clustering tests, as illustrated in Figure 6 featuring clusters in red and blue colour of a small subset of the dataset series. We used a traditional k-means clustering with $k = 2$ and in order to analyse the clustering results we projected the vectors onto a two dimensional space.

6 Conclusions

Our preliminary results show that it is possible to generate what we identified as multivariate spatio-temporal series images that describe each series' behaviour using over 2 dimensions. Results also suggest that image features extracted from a pre-trained Convolutional Neural Network can be used on classification tasks, including unsupervised series classification or clustering.

Each part of this method should be further investigated: (1) the key part of our method - the image generation using pre-attentive features - should be



Fig. 4: Multivariate temporal series plot considering multiple dimensions

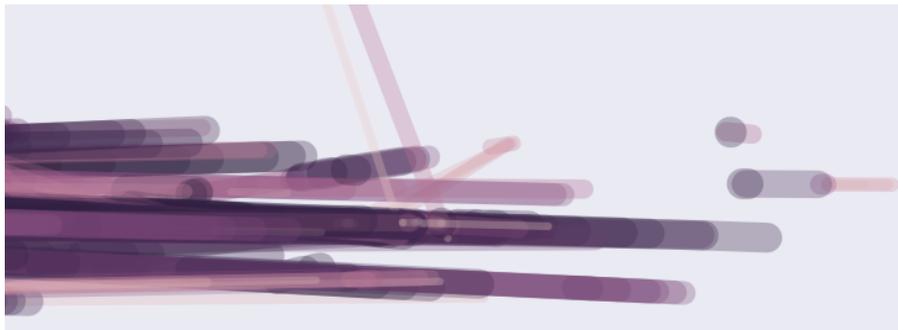


Fig. 5: Zoom over a region of the series

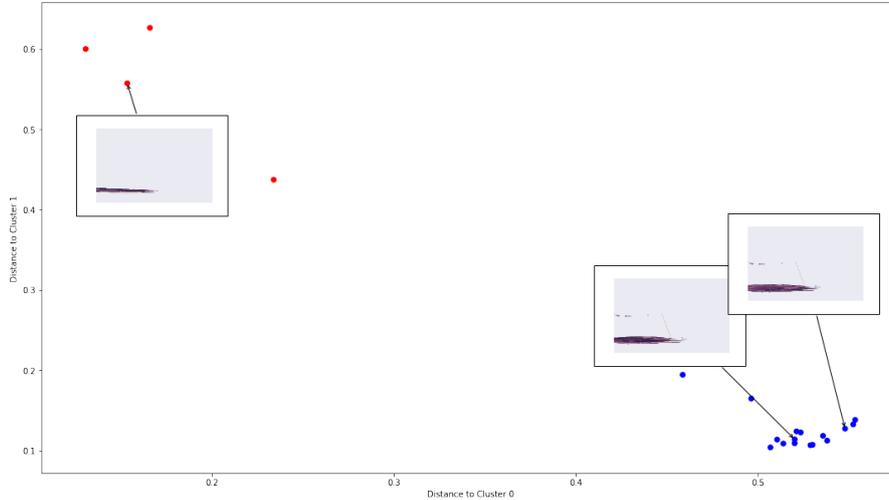


Fig. 6

expanded by comparing against the use of other features and considering the distribution of each series and the hidden structures or correlations in the dataset. (2) The convolutional network analysis may be the subject of a transfer learning approach by retraining at least the last layer of the network with a labelled dataset. The experiments should be run over a large dataset, increasing also the size of each image and applying spatial compression methods that encode the free space of each image (the spatio-temporal dimensions where the object track is inexistent).

We also consider necessary to extend the experiments to other similar domains, and further validate clustering results, using, for example, the Silhouette Coefficient score or the Elbow method, frequently applied in unsupervised models problems.

Soon, we plan to apply this approach to a labelled car traffic dataset to evaluate the possibility of automatically identify irregular driver behaviour.

7 Acknowledgments

This paper was supported by the project VizBig, financed by Fundação para Ciência e Tecnologia (FCT) under the Grant Agreement PTDC/CCI-CIF/28939/2017.

References

1. Aggarwal, C.C.: Data mining: the textbook. Springer (2015)

2. Bai, B., Li, G., Wang, S., Wu, Z., Yan, W.: Time series classification based on multi-feature dictionary representation and ensemble learning. *Expert Systems with Applications* **169**, 114162 (May 2021). <https://doi.org/10.1016/j.eswa.2020.114162>, <https://www.sciencedirect.com/science/article/pii/S0957417420309027>
3. Baydogan, M.G., Runger, G.: Learning a symbolic representation for multivariate time series classification. *Data Mining and Knowledge Discovery* **29**(2), 400–422 (Mar 2015). <https://doi.org/10.1007/s10618-014-0349-y>, <http://link.springer.com/10.1007/s10618-014-0349-y>
4. Baydogan, M.G., Runger, G.: Time series representation and similarity based on local autopatterns. *Data Mining and Knowledge Discovery* **30**(2), 476–509 (Mar 2016). <https://doi.org/10.1007/s10618-015-0425-y>, <https://doi.org/10.1007/s10618-015-0425-y>
5. Brath, R., Banissi, E.: Multivariate label-based thematic maps. *International Journal of Cartography* **3**(1), 45–60 (2017), publisher: Taylor & Francis
6. Cleveland, W.S., McGill, R.: Graphical perception: The visual decoding of quantitative information on graphical displays of data. *Journal of the Royal Statistical Society: Series A (General)* **150**(3), 192–210 (1987), publisher: Wiley Online Library
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
8. Healey, C.G., Booth, K.S., Enns, J.T.: Visualizing real-time multivariate data using preattentive processing. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **5**(3), 190–221 (1995), publisher: ACM New York, NY, USA
9. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Locally adaptive dimensionality reduction for indexing large time series databases. In: Proceedings of the 2001 ACM SIGMOD international conference on Management of data. pp. 151–162 (2001)
10. Keogh, E., Kasetty, S.: On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and knowledge discovery* **7**(4), 349–371 (2003)
11. Krekhov, A., Krüger, J.: Deadeye: A novel preattentive visualization technique based on dichoptic presentation. *IEEE transactions on visualization and computer graphics* **25**(1), 936–945 (2018), publisher: IEEE
12. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. *Data Mining and knowledge discovery* **15**(2), 107–144 (2007), publisher: Springer
13. Moskovitch, R.: Multivariate temporal data analysis-a review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **12**(1), e1430 (2022)
14. Treisman, A.: Preattentive processing in vision. *Computer vision, graphics, and image processing* **31**(2), 156–177 (1985), publisher: Elsevier

Detecting Mild Cognitive Impairment Using Fisher Vectors*

José Vicente Egas-López¹, Réka Balogh², Nóra Imre², Ildikó Hoffmann^{3,4},
László Tóth¹, Magdolna Pákáski², János Kálmán², and Gábor Gosztolya^{1,2}

¹ University of Szeged, Institute of Informatics, Szeged, Hungary

² ELKH-SZTE Research Group on Artificial Intelligence, Szeged, Hungary

³ University of Szeged, Department of Psychiatry, Szeged, Hungary

⁴ Research Center for Linguistics, ELRN, Budapest, Hungary

egasj@inf.u-szeged.hu

Abstract. Mild Cognitive Impairment (MCI) is a neurological disorder primarily affecting elderly people and it is frequently considered as a prodromal stage of dementia. MCI is characterized by a slight but measurable deficit in cognitive abilities like memory, thinking, and reasoning; and it has an effect on the patient’s speech. In this study, to represent the utterances, we employ the Fisher vector (FV) feature extraction approach, which proved to be a powerful technique in several other tasks before. Although there exist several studies related to MCI assessment via speech, to the best of our knowledge, this is the first study that employs the Fisher vector approach to assess dementia. Our experiments indicate that FVs are indeed a good feature extraction technique for this task as well, as we were able to outperform the standard i-vector and x-vector techniques. In the last part of our study, we demonstrate that the extracted feature vectors of the FV approach can be efficiently compressed by PCA, while retaining an identical classification performance.

Keywords: mild cognitive impairment · dementia · Fisher vectors · i-vectors · x-vectors · speech disorders

1 Introduction

Mild Cognitive Impairment (MCI) is a heterogeneous clinical syndrome, often considered as the transitional stage between normal cognitive aging and dementia. Its symptoms are similar to those of dementia (including deficits of memory, reasoning and problem-solving), but in MCI, impairment in the ability to carry

* This research was supported by the Hungarian Ministry of Innovation and Technology (grants no. NKFIH-1279-2/2020 and TKP2021-NVA-09), by the NRDI Office with grant FK-124413 and within the framework of the Artificial Intelligence National Laboratory Program (MILAB). G. Gosztolya was also funded by the János Bolyai Scholarship of the Hungarian Academy of Sciences and by the Hungarian Ministry of Innovation and Technology New National Excellence Program ÚNKP-21-5-SZTE.

out the activities of daily living is rarely significant [1]. Still, MCI is a rather serious condition, especially considering the fact that persons with MCI have a quadrupled risk of developing dementia later on [2]. The transition of MCI to dementia is a slow process, which can last for 15 years or more, during which subtle signs of cognitive decline can be detected [3]. The limited efficacy and applicability of currently available screening tools makes the recognition of MCI particularly challenging. Most screening measures are time consuming, and they may require trained clinicians to administer them.

Patients at this stage of cognitive decline may not show any clear signs of impairment during testing which makes the case identification even more difficult [4]. And, since current pharmacological agents seem to be more effective in the early or even in the pre-clinical stage of dementia [5], early identification is a primary concern. Thus, there is a growing need for a sensitive tool that can detect even the subtle changes indicating cognitive decline. Changes in language performance are also present in MCI, even before the manifestation of distinctive cognitive symptoms [6]. Monitoring these changes can be extremely beneficial, since speech production requires the parallel functioning of several domains, which gradually deteriorates during the course of disease progression (such as lexical semantic abilities, memory and executive functions) [7].

In addition, since speech can act as a cheap, easy-to-collect, non-invasive biomarker, there is a growing pool of acoustic parameters that have the potential to distinguish between persons with intact cognition and those with MCI. One set of these parameters may be treated as the temporal parameters of speech, including different types of pauses, duration, or speech rate. It has been shown that compared to healthy controls (HC), MCI patients tend to have a lower speech rate, and increased hesitation number and time. The proportion of pause and disfluencies is also higher in the case of patients with MCI [8]. Earlier [9] our team also found that there are certain temporal speech features which seem to characterize MCI patients relative to those for healthy people; and similar Automatic Speech Recognition (ASR)-based solutions for detecting dementia were proposed by other groups as well [10,11].

Besides focusing on temporal or ASR-based parameters which differ for the (spontaneous) speech of MCI and HC subjects, another approach is to employ more general techniques, which were shown to provide good choices for speaker groups in similar tasks. For example, i-vectors, originally introduced for speaker recognition, have been successfully applied for detecting Parkinson’s Disease [12,13] and Alzheimer’s Disease [14,15]. Likewise, x-vectors have been used to detect pathological speech [16].

In this study we use the utterance representation technique of Fisher vectors (FV, [17]) to perform a classification of utterances into two categories (MCI and HC). We will show that Fisher vectors, originally developed for image classification and image retrieval (see e.g. [18]), are an efficient feature extraction approach for this task as well. Despite its potential, however, FVs have been employed so far in audio processing relatively rarely; the handful of studies we found cover a variety of tasks like categorizing audio files as speech, music and

miscellaneous [19], emotion detection [20], and for determining food type from eating sounds [21]. Based on to our experimental results, FVs might lead to a better classification performance in MCI detection than those provided by the i-vector and x-vector techniques.

2 Data

The utterances were recorded at the Memory Clinic at the Department of Psychiatry of the University of Szeged, Hungary. A total of 50 subjects, selected from a larger pool of test participants, were used in the current study: 25 MCI patients and 25 healthy controls. These subjects were selected to ensure that the two study groups did not differ statistically from each other with regard to gender ($p = 0.734$), age ($p = 0.150$) and years of education ($p = 0.214$). All the subjects were right-handed and native speakers of Hungarian. The exclusion criteria were drug or alcohol consumption; being under pharmacological treatment affecting cognitive functions; depression; a medical history of head injuries or psychosis; and visual or auditory deficits. MCI patients were selected after a medical diagnosis supported by neuropsychological tests and CT or MRI. As it was found that MCI affects the *spontaneous* speech of the subjects more than their planned speech, we recorded and processed spontaneous speech. According to our protocol, the subjects were asked to talk about their previous day. The responses were recorded with a digital voice recorder and a tie clip microphone. Since our previous studies (e.g. [11,22]) and studies performed by other groups (e.g. [23,24]) found that MCI affect the *spontaneous* speech of the subjects more than their planned speech, we recorded and processed spontaneous speech.

3 Fisher Vectors

The Fisher Vector approach was originally developed for image representation to pool local image descriptors (e.g. SIFT, describing occurrences of rotation- and scale-invariant primitives). By modelling the low-level attributes by a Gaussian Mixture Model (GMM), it extracts a *fixed-sized* feature representation from each image, regardless of the number of local features. It can be applied to audio processing in a quite straightforward way, by substituting the SIFT descriptors of the actual image with the frame-level feature vectors of the actual utterance.

3.1 The Fisher Kernel

The Fisher Kernel (FK) seeks to measure the similarity of two objects from a parametric generative model of the data (X), which is defined as the gradient of the log-likelihood of X :

$$G_{\lambda}^X = \nabla_{\lambda} \log v_{\lambda}(X), \quad (1)$$

where $X = \{x_t, t = 1, \dots, T\}$ is a sample of T observations $x_t \in \mathcal{X}$, v represents a probability density function that models the elements in \mathcal{X} , and λ stands for

the parameter vector of v_λ [17]. Thus, the G_λ^X gradient describes the way the parameter v_λ should be changed in order to best fit the data X . The similarity of X and Y by the FK can be expressed as [25]

$$K_{FK}(X, Y) = G_\lambda^{X'} F_\lambda^{-1} G_\lambda^Y. \quad (2)$$

The Cholesky decomposition $F_\lambda^{-1} = L_\lambda' L_\lambda$ can be utilized to rewrite the Eq. (2) in terms of the dot product:

$$K_{FK}(X, Y) = \mathcal{G}_\lambda^{X'} \mathcal{G}_\lambda^Y, \quad (3)$$

where

$$\mathcal{G}_\lambda^X = L_\lambda G_\lambda^X = L_\lambda \nabla_\lambda \log v_\lambda(X). \quad (4)$$

Such a normalized gradient vector is the so-called *Fisher vector* of X [17]. Both the Fisher vector \mathcal{G}_λ^X and the gradient vector G_λ^X have the same dimension.

3.2 Fisher Vectors

Let $X = \{x_t, t = 1 \dots T\}$ be the set of D -dimensional local low-level descriptors, and us assume that the x_t samples are independent. Then Eq. (4) becomes

$$\mathcal{G}_\lambda^X = \sum_{t=1}^T L_\lambda \nabla_\lambda \log v_\lambda(x_t). \quad (5)$$

The assumption of independence allows the FV to become a sum of normalized gradients statistics $L_\lambda \nabla_\lambda \log v_\lambda(x_t)$ calculated for each x_t ; i.e.

$$X_t \rightarrow \varphi_{FK}(X_t) = L_\lambda \nabla_\lambda \log v_\lambda(X_t), \quad (6)$$

describing an operation that can be considered as a higher dimensional space embedding of the local descriptors X_t , describing the direction to which the model parameters should be changed in order to best fit the data X . When the distribution is modelled by a GMM with a diagonal covariance matrix, consisting of N Gaussian components, the FV will practically store one value for the mean and one for the variance for each component (i.e. it will have a size of ND) [17].

4 Other Feature Extraction Methods

4.1 i-vectors

GMM supervectors and JFA (Joint Factor Analysis) are successful approaches that were once the state-of-the-art systems for robust speaker recognition. In an attempt to combine of both techniques, JFA speaker factors were used as features for SVM classifiers. It was found that the channel factors estimated with JFA not only contain channel effects but speaker-dependent information as well; hence, speaker and channel factors were combined into a single space.

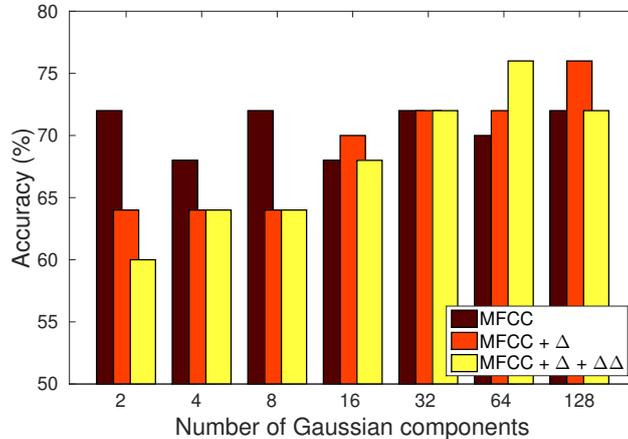


Fig. 1. Classification accuracy scores with the Fisher Vector approach as a function of the number of GMM components.

Factor Analysis (FA), which is used as a feature extractor, defines a new low-dimensional *total variability space* in which a speech utterance is defined by a new vector called *i-vector* [26] which contains the estimates of the *total factors*:

$$M = m + Tw, \quad (7)$$

where M is the GMM speaker supervector for a given signal; m is the speaker/channel-independent component (the UBM supervector); T is the Total Variability matrix; and w is a normal distributed hidden variable (i.e. the *i-vector*).

4.2 x-vectors

The x-vector approach is a neural network-based feature extraction technique providing fixed-dimensional *embeddings* for variable-length utterances, describing the observations at the utterance level rather than at the frame-level. We employed a deep network with an identical structure to that described by Snyder et al. [27]. It comprises five frame-level layers with time-delay architecture, a statistics pooling layer, two segment-layers and a final softmax output layer. The *stats pooling* layer gets the T frame-level activations from the last frame-level layer, and aggregates over the input segment by computing mean and standard deviation. These mean and standard deviation values are concatenated and then used as the input of the subsequent *segment* layers. The neurons of the final *softmax* layer correspond to the speakers in the training set. Since it only has a role during the training phase, it can be discarded after the training process has finished [27].

The embeddings produced by this network (i.e. the *x-vectors*) capture information from the speakers over the whole audio-signal. x-vectors can be extracted

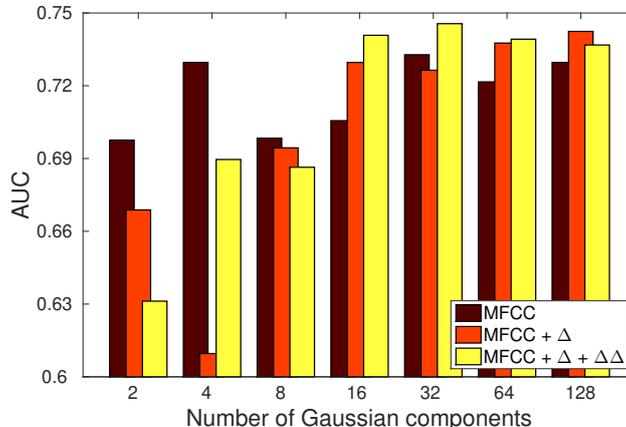


Fig. 2. AUC scores with the Fisher Vector approach as a function of the number of GMM components.

from any *segment* layer, although those from the *segment6* layer have shown to lead to a better performance than those from *segment7* [27].

5 Experiments and Results

All three tested feature extraction approaches rely on the frame-level features of the utterances. For this purpose, we used 13 dimensional Mel Frequency Cepstral Coefficients (MFCCs) with a frame-length of 25ms and a step size of 10ms, extracted by Kaldi. We also used Kaldi for i-vector and x-vector extraction, while the Fisher vector encodings were calculated by the VLFeat library.

A linear SVM classifier was utilized to discriminate the audio-signals (MCI and healthy controls). We used the libSVM implementation; the C complexity parameter was set in the range $10^{-5}, 10^{-4}, \dots, 10^1$. We trained our classifier model in 25-fold stratified cross-validation (CV): each fold consisted of the utterances of one healthy and one MCI subject. Performance was measured by classification accuracy besides equal error rate (EER), and by the area under the ROC curve (AUC).

Our preliminary tests revealed that i-vectors and Fisher vectors did not require standardization before classification, while for x-vectors this pre-processing step helped classification. The frame-level GMM model was trained on a subset of the BEA corpus [28], containing Hungarian spontaneous speech. We trained diagonal-covariance GMMs with $N = 2, 4, \dots, 128$ components; the same GMMs were used both for i-vector and Fisher vector extraction. The neural network for extracting the x-vectors embeddings was trained using the same data as the GMM models. We extracted the embeddings from the *segment6* layer with a dimension of 512.

Table 1. Classification accuracy and AUC scores achieved with the different feature extraction methods; N denotes the number of Gaussians components where applicable. Δ and $\Delta\Delta$ represent the first and second derivatives of the MFCCs, respectively.

Approach	Features	N	Acc.	AUC
i-vectors	MFCC	128	72%	0.733
	MFCC + Δ	2	52%	0.554
	MFCC + Δ + $\Delta\Delta$	2	68%	0.702
FV	MFCC	32	72%	0.733
	MFCC + Δ	128	76%	0.742
	MFCC + Δ + $\Delta\Delta$	32	72%	0.747
x-vectors	MFCC	—	60%	0.584

5.1 Results

Figure 1 shows the accuracy scores got using the FV features as a function of the number of Gaussian components N . When the features used the Δ and/or the $\Delta\Delta$ s, we can see a pattern where the accuracy results increased as the number of components increased, although there was a slight decrease in the score with the full feature set at $N = 128$. However, this is not true for the features with *no-deltas*, where the scores were quite similar (i.e. in the range 68...72%) in the whole examined area. The best accuracy results were got by using MFCCs along with their first order derivatives and with 128 Gaussian components.

In Figure 2 we see how the number of Gaussian components affects the Area-Under-Curve score. As in the other illustration, here, the color of the bars represent the use of MFCCs with a different number of derivatives. It can be seen that, as the number of components increases, the AUC scores do so as well. Here, the best AUC score was obtained by FVs using 32 Gaussian components and using both the Δ and $\Delta\Delta$ values. From both graphs, we can state that the more components there are, the better the GMM is able to model the distribution of the data. Hence, the FV encoding was able to exploit the GMM model to extract more meaningful features.

The best values attained are listed in Table 1, along with the performances of the i-vector and x-vector techniques. (For the sake of clarity, the best values and those close to it are shown as **bold**.) In terms of classification accuracy, when not using any derivatives, Fisher vectors and i-vectors gave the same performance (72%); however, adding the Δ and $\Delta\Delta$ values made the scores of the i-vector approach worse, while FVs continued to provide a good performance: in the MFCC + Δ case, the classification accuracy even increased to 76%. When inspecting the AUC scores, we can draw similar conclusions; using FVs gives the best results when using all or just the first order derivatives.

Surprisingly, the performance of x-vectors was lower for this particular dataset: such representations led to an accuracy of 60% and an AUC score of 0.584. We think that this particular performance is mainly due to the fact that the network

Table 2. Comparison of performances of the original features with the features after applying PCA.

Features	Full		PCA 95%	
	Acc.	AUC	Acc.	AUC
MFCC	72%	0.733	72%	0.733
MFCC + Δ	76%	0.742	76%	0.746
MFCC + Δ + $\Delta\Delta$	72%	0.747	72%	0.746

was trained using a relatively small amount of data (slightly less than 2 hours). GMMs turned out to be more robust using the same amount of data.

As for the Fisher vector representations, they attained the best accuracy and AUC scores for all our experiment configurations. However, it is worth pointing out that, as can be observed in Table 1, i-vectors achieve the same accuracy and AUC scores when no Δ values are used. We can attribute this behavior to the size of the dataset used for UBM training. However, when using Δ and $\Delta\Delta$, FVs encodings outperformed the other representations significantly, achieving the best AUC score of 0.747.

6 Applying Principal Component Analysis

Despite the classification performance, one drawback of the Fisher vector encoding is the size of features generated. Although we only used 13 MFCC components (in the literature, 20 and 23 are also common values), also including the Δ s with $N = 128$ Gaussian components led to 6656 features, while with the $\Delta\Delta$ values, this increases to 9984. Next, we will show that this feature vector size can be efficiently reduced by Principal Component Analysis (PCA), while retaining the same classification performance. In this experiment, we kept 95% of the information present in the Fisher vectors. For the sake of comparability, we used the same N values which proved to be optimal in our previous experiment.

Table 2 lists the accuracy and AUC scores for the Fisher vector features, obtained with the full feature vectors and after applying PCA. Clearly, the shown scores are practically identical: while there are insignificant differences in the AUC values (i.e. ≤ 0.004), the classification accuracy scores remained unchanged in each case. However, the size of the feature vector was reduced to 3 – 16% of its original length, allowing a more compact SVM model and a faster prediction.

7 Conclusions and Discussion

In this study we showed that Fisher vector encodings are efficient for representing speech utterances in order to detect Mild Cognitive Impairment, while the i-vector and x-vector approaches allowed slightly or significantly lower scores. In the next part of our study, we noted that FVs tend to extract quite large feature vectors. To decrease the number of attributes, we applied Principal Component

Analysis; and based on our results, this dimensionality reduction step could be performed without any real loss in performance.

Evidently, we carried out our experiments on a limited amount of data. Although working with the utterances of 50 subjects (which was also a result of using demographically aligned groups) is quite normal in the medical speech processing area, from a machine learning perspective this is considered a very small dataset. In the near future we plan to repeat our experiments on the utterances of more subjects. The sensitivity of the examined methods to acoustic or other factors (e.g. language) is also an open question. These, however, are clearly the subjects of future work.

References

1. R. C. Petersen, B. Caracciolo, C. Brayne, S. Gauthier, V. Jelic, and L. Fratiglioni, "Mild Cognitive Impairment: A concept in evolution," *Journal of Internal Medicine*, vol. 275, no. 3, pp. 214–228, 2014.
2. R. F. de Bruijn, S. Akoudad, L. G. Cremers, A. Hofman, W. J. Niessen, A. Van Der Lugt, P. J. Koudstaal, M. W. Vernooij, and M. A. Ikram, "Determinants, MRI correlates, and prognosis of Mild Cognitive Impairment: the Rotterdam study," *Journal of Alzheimer's Disease*, vol. 42, no. s3, pp. S239–S249, 2014.
3. C. Laske, H. R. Sohrabi, S. M. Frost, K. López-de Ipiña, P. Garrard, M. Buscema, J. Dauwels, S. R. Soekadar, S. Mueller, C. Linnemann *et al.*, "Innovative diagnostic tools for early detection of Alzheimer's disease," *Alzheimer's & Dementia*, vol. 11, no. 5, pp. 561–578, 2015.
4. C. F. Eliassen, I. Reinvang, P. Selnes, R. Grambaite, T. Fladby, and E. Hessen, "Biomarkers in subtypes of Mild Cognitive Impairment and subjective cognitive decline," *Brain and Behavior*, vol. 7, no. 9, p. e00776, 2017.
5. G. Szatlóczki, I. Hoffmann, V. Vincze, J. Kálmán, and M. Pákáski, "Speaking in Alzheimer's disease, is that an early sign? Importance of changes in language abilities in Alzheimer's disease," *Frontiers in Aging Neuroscience*, vol. 7, p. 195, 2015.
6. K. C. McCullough, K. A. Bayles, and E. D. Bouldin, "Language performance of individuals at risk for Mild Cognitive Impairment," *Journal of Speech, Language, and Hearing Research*, vol. 62, no. 3, pp. 706–722, 2019.
7. D. Beltrami, G. Gagliardi, R. Rossini Favretti, E. Ghidoni, F. Tamburini, and L. Calzà, "Speech analysis by natural language processing techniques: A possible tool for very early detection of cognitive decline?" *Frontiers in Aging Neuroscience*, vol. 10, p. 369, 2018.
8. B. S. Kim, Y. B. Kim, and H. Kim, "Discourse measures to differentiate between Mild Cognitive Impairment and healthy aging," *Frontiers in Aging Neuroscience*, vol. 11, p. 221, 2019.
9. L. Tóth, I. Hoffmann, G. Gosztolya, V. Vincze, G. Szatlóczki, Z. Bánréti, M. Pákáski, and J. Kálmán, "A Speech Recognition-based solution for the automatic detection of Mild Cognitive Impairment from spontaneous speech," *Current Alzheimer Research*, vol. 15, no. 2, pp. 130–138, 2018.
10. K. L. Fors, K. C. Fraser, and D. Kokkinakis, "Automated syntactic analysis of language abilities in persons with mild and subjective cognitive impairment." in *Proceedings of MIE*, 2018, pp. 705–709.

11. G. Gosztolya, V. Vincze, L. Tóth, M. Pákási, J. Kálmán, and I. Hoffmann, "Identifying Mild Cognitive Impairment and mild Alzheimer's disease based on spontaneous speech using ASR and linguistic features," *Computer, Speech & Language*, vol. 53, no. Jan, pp. 181–197, 2019.
12. N. García, J. R. Orozco-Arroyave, L. F. D'Haro, N. Dehak, and E. Nöth, "Evaluation of the neurological state of people with Parkinson's Disease using i-vectors," in *Proceedings of Interspeech*, 2017.
13. N. García, J. Vásquez-Correa, J. R. Orozco-Arroyave, and E. Nöth, "Multimodal i-vectors to Detect and Evaluate Parkinson's Disease," *Proceedings of Interspeech*, pp. 2349–2353, 2018.
14. J. Weiner and T. Schultz, "Selecting features for automatic screening for dementia based on speech," in *Proceedings of SPECOM*. Leipzig, Germany: Springer International Publishing, 2018, pp. 747–756.
15. J. V. Egas-López, L. Tóth, I. Hoffmann, J. Kálmán, M. Pákási, and G. Gosztolya, "Assessing Alzheimer's Disease from Speech Using the i-vector Approach," in *Proceedings of SPECOM*. Springer, 2019.
16. C. Botelho, F. Teixeira, T. Rolland, A. Abad, and I. Trancoso, "Pathological speech detection using x-vector embeddings," 2020.
17. J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher Vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, pp. 222–245, 2013.
18. Y. Song, J. J. Zou, H. Chang, and W. Cai, "Adapting Fisher vectors for histopathology image classification," in *Proceedings of ISBI*. IEEE, 2017, pp. 600–603.
19. P. J. Moreno and R. Rifkin, "Using the Fisher kernel method for web audio classification," in *Proceedings of ICASSP*, Dallas, TX, USA, 2010, pp. 2417–2420.
20. G. Gosztolya, "Using the Fisher vector representation for audio-based emotion recognition," *Acta Polytechnica Hungarica*, vol. 17, no. 6, pp. 7–23, 2020.
21. H. Kaya, A. A. Karpov, and A. A. Salah, "Fisher Vectors with cascaded normalization for paralinguistic analysis," in *Proceedings of Interspeech*, 2015, pp. 909–913.
22. G. Gosztolya, L. Tóth, T. Grósz, V. Vincze, I. Hoffmann, G. Szatlóczki, M. Pákási, and J. Kálmán, "Detecting Mild Cognitive Impairment from spontaneous speech by correlation-based phonetic feature selection," in *Proceedings of Interspeech*, San Francisco, CA, USA, Sep 2016, pp. 107–111.
23. B. Roark, M. Mitchell, J.-P. Hosom, K. Hollingshead, and J. Kaye, "Spoken language derived measures for detecting Mild Cognitive Impairment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2081–2090, 2011.
24. A. Satt, R. Hoory, A. König, P. Aalten, and P. H. Robert, "Speech-based automatic and robust detection of very early dementia," in *Proceedings of Interspeech*, Singapore, 2014, pp. 2538–2542.
25. T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Proceedings of NIPS*, Denver, CO, USA, 1998, pp. 487–493.
26. N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.
27. D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker verification," in *Proceedings of ICASSP*, 2018, pp. 5329–5333.
28. T. Neuberger, D. Gyarmathy, T. E. Grácsi, V. Horváth, M. Gósy, and A. Beke, "Development of a large spontaneous speech database of agglutinative Hungarian language," in *Proceedings of TSD*, Brno, Czech Republic, Sep 2014, pp. 424–431.

Null Additivization of a Monotone Measure on a Finite Set

Ryoji Fukuda¹, Aoi Honda², and Yoshiaki Okazaki³

¹ Oita University, 700 Dan-noharu, Oita city, Oita 870-1192, JAPAN
rfukuda@oita-u.ac.jp

² Kyushu Institute of Technology, 680-4 Kawazu, Iizuka city, Fukuoka 820-8502, JAPAN
aoi@ai.kyutech.ac.jp

³ Fuzzy Logic Systems Institute, 680-41 Kawazu, Iizuka city, Fukuoka 820-0067, JAPAN
okazaki@flsi.or.jp

Abstract. Massless points with respect to a monotone measure are inconspicuous, however, they sometimes play important roles. In this article, we try to treat such roles along with standard elements of the monotone measure space. By using Möbius transform, every inconspicuous role can be expressed by a certain element. We provide a method to construct a null additive space in which inconspicuous elements and standard elements in the given monotone measure space are mixed. In this space, a function on the original space corresponds to a certain function on the null additive space, and Choquet integral values are preserved under this translation.

Keywords: Fuzzy measure theory, Set functions, measures and integrals with values in ordered spaces.

1 Introduction

In this study, we discuss the null additivity of monotone measures, or set functions more generally. For a set function μ defined on a measurable space (X, \mathcal{B}) , we always assume that $\mu(\emptyset) = 0$. We define $A \in \mathcal{B}$ as a μ -null set (or simply a null set) if any measurable subset B of A satisfies $\mu(B) = 0$. We define that μ is null additive if $\mu(B) = \mu(B \cup A)$ for any null set A and a measurable set $B \in \mathcal{B}$, and that μ is weakly null additive if $A \cup B$ is a μ -null set for any null sets $A, B \in \mathcal{B}$. In the case where μ is a monotone measure, the null additivity is defined in [1] or [2], and these definitions are delicate and differ slightly from study to study.

The concept of null and weak null additivities are important in analyses of measurable function on monotone measure spaces. For example, considering the relation $\mu(\{x : f(x) \neq g(x)\}) = 0$ (we denote it briefly by $\mu(\{f \neq g\}) = 0$ in the sequel) for a pair of measurable functions f, g , a monotone measure μ is weakly null additive if and only if this relation is an equivalent relation. Moreover, under

the condition of null-continuity from below, the weak null additivity implies the existence of completion of the σ -algebra \mathcal{B} , and the completion is unique when μ is null additive. ([3],[4]). For a non-discrete monotone measure space, equivalent conditions for the null and weak null additivities can be described by the generalized Möbius transform (for the definition of Möbius transform, see [5]; for the equivalence conditions, [4]). Using the relation between classical and generalized Möbius transforms, which we show in this study, we describe conditions for the null and weak null additivities using the classical Möbius transform.

A set function with constructive k -additivity ($k \in \mathbb{N}$), the definition of which was given in [6], can be described using a signed measure on the space of finite subsets with cardinalities not more than k . In the case where X is a finite set, any set functions are (constructively) k -additive for some k ($k \leq |X|$). Each point mass of the above measure with respect to a point A (the measure is defined on a set family) is equivalent with the corresponding Möbius transform τ_A . Then, if there is a massless point $a \in X$, some influence factor may be represented using a Möbius transform with respect to some finite subset of X . Thus, for an arbitrary monotone measure space, we aim to construct a new non-additive measure space with no massless point by replacing some points with suitable influence factors, each of which has a one-to-one correspondence with some subset of X .

The new non-additive measure space inherit certain properties of the original monotone measure space. As the first step, for a given monotone measure space (X, μ) (X is a finite set and μ is a monotone measure) and a nonnegative function f on X , we aim to construct a non-additive measure space $(\tilde{X}, \tilde{\mu})$ and a function \tilde{f} on \tilde{X} , for which two Choquet integrals $\int^{Ch} f d\mu, \int^{Ch} \tilde{f} d\tilde{\mu}$ are equivalent. Moreover, the transformed $(\tilde{X}, \tilde{\mu})$ and a function \tilde{f} also have same distribution functions. Then, for all distribution function type integrals, the Sugeno integrals or the Shilkret integrals, two integral values are equivalent.

2 Basic Properties

Let (X, \mathcal{B}) be a measurable space and μ be a set function ($\mu(\emptyset) = 0$). We define the null set, null additivity, and null additivity as follows.

Definition 1. (a) $A \in \mathcal{B}$ is a μ -null set (simply null set) if $\mu(B) = 0$ for any $B \subset A, B \in \mathcal{B}$.

(b) μ is null additive if $\mu(B) = \mu(B \cup A)$ for any null set A and $B \in \mathcal{B}$.

(c) μ is weakly null additive if $\mu(B \cup A) = 0$ for any null sets A and B .

The generalized Möbius transform $\tau(\cdot \cdot \cdot)$ is defined in Definition 2. Consider a partition $\mathbb{D} = \{D_j\}_{j=1}^n$ of a general measurable space, then a set function can be regarded as a set function defined on the n -point space \mathbb{D} . Under this restriction, the generalized Möbius transform coincides with the classical one.

Definition 2. ([5])

(a) $\mathcal{D} = \{\mathbb{D} = \{D_j\}_{j=1}^n, n \in \mathbb{N}, D_j \in \mathcal{B}, D_j \cap D_k = \emptyset, \forall j, k \leq n, j \neq k\}$.

- (b) $\tau(\mathbb{D}) = \tau(D_1, \dots, D_n)$, ($\mathcal{D} \rightarrow \mathbb{R}$) is defined by inductively:
1. $\tau(\{D\}) = \mu(D)$ for any $D \in \mathcal{B}$.
 2. $\tau(\mathbb{D}) = \mu(\bigcup_{D \in \mathbb{D}} D) - \sum_{\mathbb{D}' \subsetneq \mathbb{D}} \tau(\mathbb{D}')$.

The null and weak null additivities are described by the generalized Möbius transform.

Proposition 1. ([4]) Let (X, \mathcal{B}, μ) be a non-additive measure space.

(a) μ is null additive if and only if

$$\mathbb{D} \in \mathcal{D}, \exists A \in \mathbb{D}, A \text{ is a null set.} \Rightarrow \tau(\mathbb{D}) = 0.$$

(b) μ is null additive if and only if

$$\mathbb{D} \in \mathcal{D}, \forall A \in \mathbb{D}, A \text{ is a null set.} \Rightarrow \tau(\mathbb{D}) = 0.$$

Originally, the above proposition was proven for the case in which μ is a monotone measure. The same method is valid for this case.

The classical Möbius transform for a set function on a finite set is well known (see for example [7]) and it was used in various situations. In the case where X is a finite set, we consider the classical Möbius transform $\{\tau_A\}_{A \subset X}$ of a set function $\mu : (2^X \rightarrow \mathbb{R})$. The generalized Möbius transform can be represented by the classical one.

Proposition 2. Let X be a finite set, μ be a set function with $\mu(\emptyset) = 0$, τ be the (generalized) Möbius transform, and $\{\nu_B\}_{B \subset X}$ be the classical Möbius transform. For an element $\mathbb{D} = \{A_1, \dots, A_n\}$, we define the family of finite subsets $\Gamma(A_1, \dots, A_n)$ as follows.

$$\Gamma(A_1, \dots, A_n) = \{B \subset \bigcup_{j \leq n} A_j : \forall j \leq n, A_j \cap B \neq \emptyset\}.$$

Then, we have:

- (a) $\tau_A = \tau(\{a_1\}, \dots, \{a_n\})$, $A = \{a_1, \dots, a_n\}$.
(b) $\tau(A_1, \dots, A_n) = \sum_{B \in \Gamma(A_1, \dots, A_n)} \tau_B$

Proof. (a) We can describe the definition of classical Möbius transform by a similar method with Definition 1, and this may be easily verified using the above definition. It was also checked in [5]. (b) We prove this formula by induction on the cardinality $|\mathbb{D}| = n$. For a set $A \subset X$, by the definition of the Möbius transform,

$$\mu(A) = \sum_{B \subset A} \tau_B.$$

Then, this proves the case $n = 1$ because $\tau(\{A\}) = \mu(A)$, and $\Gamma(A) = \{B : B \subset A, B \neq \emptyset\}$.

Assume this formula when n is not more than $n_0 - 1$. For any $n \in \mathbb{N}$ and a disjoint set family $\{A_1, \dots, A_n\}$, the following formula was proven in [5] for any n .

$$\begin{aligned} & \tau(A_1, \dots, A_{n-2}, A_{n-1} \cup A_n) \\ &= \tau(\dots, A_{n-2}, A_{n-1}) + \tau(\dots, A_{n-2}, A_n) + \tau(\dots, A_{n-2}, A_{n-1}, A_n). \end{aligned}$$

Then,

$$\begin{aligned} & \tau(\dots, A_{n-2}, A_{n-1}, A_n) \\ &= \tau(\dots, A_{n-2}, A_{n-1} \cup A_n) - \tau(\dots, A_{n-2}, A_{n-1}) - \tau(\dots, A_{n-2}, A_n). \end{aligned}$$

Set

$$\begin{aligned} \mathbb{D}_0 &= \{\dots, A_{n_0-2}, A_{n_0-1} \cup A_{n_0}\}, & \mathbb{D}_1 &= \{\dots, A_{n_0-2}, A_{n_0-1}\}, \\ \mathbb{D}_2 &= \{\dots, A_{n_0-2}, A_{n_0}\}, & \mathbb{D}_3 &= \{\dots, A_{n_0-2}, A_{n_0-1}, A_{n_0}\}. \end{aligned}$$

Then,

$$\tau(\mathbb{D}_3) = \sum_{B \in \Gamma(\mathbb{D}_0)} \tau_B - \sum_{B \in \Gamma(\mathbb{D}_1)} \tau_B - \sum_{B \in \Gamma(\mathbb{D}_2)} \tau_B.$$

For an element of $F \in \Gamma(A_1, \dots, A_{n_0-2}, A_{n_0-1} \cup A_{n_0})$ satisfies one and only one of the following (1) \sim (3).

- (1) $F \cap A_{n_0-1} \neq \emptyset$ and $F \cap A_{n_0} \neq \emptyset$, that is, $F \in \Gamma(A_1, \dots, A_{n_0-2}, A_{n_0-1}, A_{n_0})$.
- (2) $F \cap A_{n_0-1} \neq \emptyset$ and $F \cap A_{n_0} = \emptyset$, that is, $F \in \Gamma(A_1, \dots, A_{n_0-2}, A_{n_0-1})$.
- (3) $F \cap A_{n_0-1} = \emptyset$ and $F \cap A_{n_0} \neq \emptyset$, that is, $F \in \Gamma(A_1, \dots, A_{n_0-2}, A_{n_0})$.

Moreover, $\Gamma(\mathbb{D}_1)$, $\Gamma(\mathbb{D}_2)$, and $\Gamma(\mathbb{D}_3)$ are disjoint from each other. This implies that

$$\sum_{B \in \Gamma(\mathbb{D}_0)} \tau_B - \sum_{B \in \Gamma(\mathbb{D}_1)} \tau_B - \sum_{B \in \Gamma(\mathbb{D}_2)} \tau_B = \sum_{B \in \Gamma(\mathbb{D}_3)} \tau_B,$$

and this concludes the proof. \square

The above proposition asserts that generalized Möbius transform can be expressed by classical one when $|X| < \infty$. The conditions for the null and weak null additivities are expressed as follows.

Proposition 3. *Let X be a finite set and μ be a set function defined on 2^X . Then, we have the following.*

(a) μ is null additive if and only if

$$A \subset X, \exists x \in A, \mu(\{x\}) = 0, \Rightarrow \tau_A = 0. \quad (1)$$

(b) μ is null additive if and only if

$$A \subset X, \forall x \in A, \mu(\{x\}) = 0, \Rightarrow \tau_A = 0.$$

Proof. (a) Let $\mathbb{D} = \{A_1, \dots, A_n\}$ be an arbitrary element of \mathcal{D} . Then, by Proposition 1, we have only to prove that

$$\mu(A_1) = 0 \Rightarrow \tau(\{A_1, \dots, A_n\}) = 0, \quad (2)$$

using the condition (1). Because the condition (2) clearly implies (1), then we have that conditions (1) and (2) are equivalent.

By Proposition 2,

$$\tau(\mathbb{D}) = \sum_{B \in \Gamma(\mathbb{D})} \tau_B.$$

For any element $B \in \Gamma(\mathbb{D})$, there exist $a \in B \cap A_1$. By the assumption of (a), we have $\tau_B = 0$ for each $B \in \Gamma(\mathbb{D})$. Hence, $\tau(\mathbb{D}) = 0$.

(b) We need only to prove that $\tau(\{A_1, \dots, A_n\}) = 0$ ($\mathbb{D} = \{A_1, \dots, A_n\} \in \mathcal{D}$) if A_k is a null set for each $k \leq n$, under the assumption of (b).

To do so, we use the Proposition 2 again:

$$\tau(\mathbb{D}) = \sum_{B \in \Gamma(\mathbb{D})} \tau_B.$$

By the definition of $\Gamma(\mathbb{D})$, any element $B \in \Gamma(\mathbb{D})$ satisfies

$$B \subset \bigcup_{k \leq n} A_k.$$

Hence, each point $b \in B$ satisfies $\mu(\{b\}) = 0$ because $b \in A_k$ for some $k \leq n$ and A_k is a null set for any $k \leq n$. Then, we also have $\tau_B = 0$ for any $B \in \Gamma(\mathbb{D})$, and this concludes the proof. \square

3 Influence Factor

By the arguments in the previous section, a set function on a finite set is null additive if there are no massless points. The purpose of this section is to construct a null additive non-additive measure space, by removing massless points and adding some influence elements.

Definition 3. Let X be a finite set and μ be a set function on 2^X . Then, $A \subset X$ is an influence factor if there exists $\exists a \in A$ such that

- (a) $\mu(\{a\}) = 0$,
- (b) $\tau_A \neq 0$, and
- (c) $a \in B \subsetneq A \Rightarrow \tau_B = 0$.

N denotes the family of all massless points, and \mathcal{A} denotes the family of all influence factors. We consider an influence element ι_A corresponding to an influence factor $A \in \mathcal{A}$.

For each $B \subset X$, set

$$I_B = \{A \mid A \in \mathcal{A}, A \subset B\},$$

$$\mathbb{I} = \{B : B \cap N \neq \emptyset, B = (B \setminus N) \cup \bigcup_{A \in I_B} A\}.$$

Remarks: For a massless point a satisfies that $\tau_A = 0$ if $a \in A$, then we can remove a from X . Then, we assume that, for any point a , there exists A which satisfies $\tau_A \neq 0$ and $a \in A$. An influence factor is a minimal subset satisfying $a \in A$ and $\tau_A \neq 0$. However, in general, the relation between a massless point and an influence factor is not a one-to-one correspondence. The relation between “factor” and “element” is similar to that between “fuzzy set” and “membership function”.

Proposition 4. *Let X be a finite set, μ be a monotone measure on 2^X , and A be an influence factor. Then, we have $\tau_A > 0$.*

Proof. Let $a \in A$ be a point satisfying the condition in Definition 3

$$\begin{aligned}\mu(A) &= \sum_{B \subsetneq A} \tau_B + \tau_A \\ &= \sum_{B \subsetneq A, a \notin B} \tau_B + \sum_{B \subsetneq A, a \in B} \tau_B + \tau_A \\ &= \mu(A \setminus \{a\}) + \tau_A.\end{aligned}$$

The last equality holds by the condition (c) in Definition 3. Using the monotonicity, we have $\tau_A > 0$. \square

Proposition 5. *Let X be a finite set, μ be a monotone measure on 2^X . Then, we have the following.*

- (a) *If $\tau_B \neq 0$, then $B = (B \setminus N) \cup \bigcup_{A \in \mathcal{A}, A \subset B} A$.*
(b) *Let $B, B' \subset X$ be any subsets of X . Then $B = B'$ if and only if $(B \setminus N) \cup I_B = (B' \setminus N) \cup I_{B'}$.*

Notation and Remark: Set $I_B = \{A \in \mathcal{A}, A \subset B\}$ and \mathbb{I} is defined by $\mathbb{I} = \{B : B = (B \setminus N) \cup \bigcup_{A \in \mathcal{A}, A \subset B} A\}$.

(a) of this proposition implies that $\tau_B = 0$ if $B \cap N \neq \emptyset$ and $B \notin \mathbb{I}$.

Proof. (a) Consider an element a satisfying $a \in B \cap N$. We prove that there exist $A \in \mathcal{A}$ such that $a \in A \subset B$. If all proper subsets $C \subsetneq A$ with $a \in C$ satisfies $\tau_C = 0$ then $B \in \mathcal{A}$. If there exist $C \subsetneq B$ with $a \in C$ and $\tau_C \neq 0$, by replacing B with C and iterating this process until we find an element of \mathcal{A} .

This property implies that, for all $a \in B \cap N$, there exist $C \in \mathcal{A}$ such that $a \in C \subset B$. Therefore, we have

$$B = (B \setminus N) \cup \left(\bigcup_{C \subset B, C \in \mathcal{A}} C \right),$$

and this concludes the proof of (a).

(b) As the “only if” part is clear, and we prove the contrapositive of the “if” part. Assume that $B \neq B'$. If $B \cap N \neq B' \cap N$, clearly we have $(B \setminus N) \cup I_B \neq (B' \setminus N) \cup I_{B'}$. Then, we consider the case $B \cap N = B' \cap N$, and this implies $B \cap N \neq B' \cap N$. Assume that $\exists a \in (B \cap N) \setminus B'$. Then, there exists $A \in \mathcal{A}$ satisfying $a \in A \subset B$, because $B \supset \bigcup_{A \in \mathcal{A}, A \subset B} A = \bigcup_{A \in I_B} A$. Assuming that $A \in I_{B'}$ then $a \in A \subset B'$ contradicts to the assumption $a \notin B'$. Therefore, we have $I_B \neq I_{B'}$, and this concludes the proof. \square

4 Construction of null additivization space.

Using the argument in previous sections, we define the null additivization of monotone measure space as follows.

Definition 4. Let X be a finite set and μ be a monotone measure on 2^X . Set $\tilde{X} = (X \setminus N) \cup \{\iota_A : A \in \mathcal{A}\}$. We define the transformed set function $\tilde{\mu}$ by defining its Möbius transform $\tilde{\tau}$ as follows.

$$\begin{aligned} A \in \mathcal{A} &\Rightarrow \tilde{\tau}_{\{A\}} = \tau_A, \\ B \subset X \setminus N &\Rightarrow \tilde{\tau}_B = \tau_B, \\ B \in \mathbb{I}, B = \bigcup_{C \in I_B} C &\Rightarrow \tilde{\tau}_{B \cup I_B} = 0, \\ B \in \mathbb{I}, B \neq \bigcup_{C \in I_B} C &\Rightarrow \tilde{\tau}_{B \cup I_B} = \tau_B, \\ \text{otherwise, } U \subset \tilde{X} &\Rightarrow \tilde{\tau}_U = 0. \end{aligned}$$

Let $\tilde{\mu}$ be the set function with the Möbius transform is $\tilde{\tau}$. Then, $(\tilde{X}, \tilde{\mu})$ is the null additivization of (X, μ) .

Definition 5. Let (Y, ν) be a (not necessary monotone) non-additive measure space on a finite set, $\{\tau_B\}_{B \in 2^Y}$ be its Möbius transform, and f be a nonnegative function on Y . Then, we define a Choquet integral of f on (Y, ν) as follows.

$$\int^{Ch} f d\nu = \sum_{B \in 2^Y} \min_{y \in B} f(y) \tau_B.$$

It is well-known that the above Choquet integral is identical with the standard version, when μ is a monotone measure. Then, the following theorem holds.

Theorem 1. Let $(\tilde{X}, \tilde{\mu})$ and \tilde{f} be the null additivization of a monotone measure space (X, μ) and a nonnegative function f on X , which are given in Definition 4. Then, we have

$$\int_X^{Ch} f d\mu = \int_{\tilde{X}}^{Ch} \tilde{f} d\tilde{\mu}.$$

Proof.

By Proposition 5,

$$\begin{aligned} \int_X^{Ch} f d\mu &= \sum_{B \subset X} \min_{x \in B} f(x) \tau_B \\ &= \sum_{B \subset X \setminus N} \min_{x \in B} f(x) \tau_B + \sum_{B \in \mathbb{I}} \min_{x \in B} f(x) \tau_B \\ &= \sum_{B \subset X \setminus N} \min_{x \in B} f(x) \tau_B + \sum_{A \in \mathcal{A}} \min_{x \in A} f(x) \tilde{\tau}_{\{A\}} + \sum_{B \in \mathbb{I}, \emptyset \neq A} \min_{y \in (B \setminus N) \cup I_B} f(x) \tilde{\tau}_{\{(B \setminus N) \cup I_B\}} \\ &= \int_{\tilde{X}}^{Ch} \tilde{f} d\tilde{\mu}. \end{aligned}$$

□

Example 1. Set $X = \{a, b, c\}$ and define a monotone measure μ on X as follows.

$$\begin{aligned}\mu(\{a\}) &= \mu(\{b\}) = 0, \quad \mu(\{c\}) = 1, \\ \mu(\{a, b\}) &= 1, \quad \mu(\{a, c\}) = 2, \quad \mu(\{b, c\}) = 3, \\ \mu(\{a, b, c\}) &= 3.\end{aligned}$$

Then, the Möbius transform is given by

$$\begin{aligned}\tau_{\{a\}} &= \tau_{\{b\}} = 0, \quad \tau_{\{c\}} = 1, \\ \tau_{\{a, b\}} &= \tau_{\{a, c\}} = 1, \quad \tau_{\{b, c\}} = 2, \\ \tau_{\{a, b, c\}} &= -2.\end{aligned}$$

In this case, the null additivization is given by:

$$\begin{aligned}\tilde{X} &= \{c, \iota_{\{a, b\}}, \iota_{\{a, c\}}, \iota_{\{b, c\}}\} \\ &= \{c, \alpha, \beta, \gamma\}, \\ N &= \{a, b\}, \quad \mathcal{A} = \{\alpha, \beta, \gamma\}.\end{aligned}$$

Then, the translated Möbius transform is calculated as follows.

$$\begin{aligned}\tilde{\tau}_{\{c\}} &= 1, \quad \tilde{\tau}_{\{\alpha\}} = 1, \quad \tilde{\tau}_{\{\beta\}} = 1, \quad \tilde{\tau}_{\{\gamma\}} = 2, \\ \tilde{\tau}_{\{c, \alpha\}} &= \tilde{\tau}_{\{c, \beta\}} = \tilde{\tau}_{\{c, \gamma\}} = 0, \\ \tilde{\tau}_{\{c, \alpha, \beta\}} &= \tilde{\tau}_{\{c, \beta, \gamma\}} = \tilde{\tau}_{\{c, \alpha, \gamma\}} = 0, \\ \tilde{\tau}_{\{\alpha, \beta\}} &= \tilde{\tau}_{\{\beta, \gamma\}} = \tilde{\tau}_{\{\alpha, \gamma\}} = 0, \\ \tilde{\tau}_{\{\alpha, \beta, \gamma\}} &= 0, \\ \tilde{\tau}_{\{c, \alpha, \beta, \gamma\}} &= (\tau_{a, b, c}) - 2.\end{aligned}$$

Let $\tilde{\mu}$ be a set function defined by the above Möbius transform. We have $\tilde{\mu}(\{\alpha, \beta, \gamma\}) = 4$ and $\tilde{\mu}(\{c, \alpha, \beta, \gamma\}) = 3$. Hence, this set function is not monotone.

Consider a function f on X , $f(a) = 1, f(b) = 2, f(c) = 3$. Then,

$$\begin{aligned}f(\alpha) &= f(a) \wedge f(b) = 1, \quad f(\beta) = f(a) \wedge f(c) = 1, \\ f(\gamma) &= f(b) \wedge f(c) = 2.\end{aligned}$$

$$\begin{aligned}\int^{Ch} f d\mu &= \sum_{B \subset X} \min_{x \in B} f(x) \tau_B \\ &= f(c) \tau_{\{c\}} + (f(a) \wedge f(b)) \tau_{\{a, b\}} + (f(a) \wedge f(c)) \tau_{\{a, c\}} + \\ &\quad (f(b) \wedge f(c)) \tau_{\{b, c\}} + (f(a) \wedge f(b) \wedge f(c)) \tau_{\{a, b, c\}} \\ &= 3 \times 1 + 1 \times 1 + 1 \times 1 + 2 \times 2 + 1 \times (-2) = 7.\end{aligned}$$

On the other hand, Choquet integral of translated function is calculated as follows.

$$\begin{aligned}
\int^{Ch} f d\tilde{\mu} &= \sum_{U \subset \tilde{X}} (\min_{x \in U} f(x)) \tilde{\tau}_U \\
&= f(c)\tilde{\tau}_{\{c\}} + f(\alpha)\tilde{\tau}_{\{\alpha\}} + f(\beta)\tilde{\tau}_{\{\beta\}} + f(\gamma)\tilde{\tau}_{\{\gamma\}} + \\
&\quad (f(\alpha) \wedge f(\beta) \wedge f(\gamma) \wedge f(c))\tilde{\tau}_{\{c,\alpha,\beta,\gamma\}} \\
&= 3 \times 1 + 1 \times 1 + 1 \times 1 + 2 \times 2 + 1 \times (-2) = 7.
\end{aligned}$$

Thus, two integral values are the same.

The translated set function in the above example is not monotone. Therefore, this example illustrates that a translated set function is not necessary monotone even if the original set function is monotone.

Lemma 1. *Let X be a finite set, μ be a monotone measure on 2^X , and f be a nonnegative function on X . \tilde{X} , $\tilde{\mu}$, and \tilde{f} are null additivization of X , μ , and f respectively. Then,*

- (a) *If $\int^{Ch} f d\mu = 0$, we have $f(x) = 0$ for any $x \in X \setminus N$, and $\min_{x \in A} f(x) = 0$ for any $A \in \mathcal{A}$, where N is the set of all massless points and \mathcal{A} is the set of all influence factors.*
- (b) *$\mu(\{f(x) > r\}) = \tilde{\mu}(\{\tilde{f}(x) > r\})$ for any $r > 0$.*

Remark: By Lemma 1 (b), the distribution function of f is same with that of \tilde{f} . This implies that the Sugeno and Shilkret integrals take same values on the both spaces, where these integrals are defined on (not necessarily monotone) non-additive measure space using the distribution functions.

Proof. (a) Using the monotonicity of μ and the Choquet integral with respect to μ ,

$$\begin{aligned}
0 &= \int^{Ch} f(x) d\mu \geq \int^{Ch} f(x) 1_{\{b\}} d\mu = f(b)\mu(\{b\}) \geq 0, \\
0 &= \int^{Ch} f(x) d\mu \geq \int^{Ch} f(x) 1_A d\mu \geq (\min_{x \in A} f(x))\mu(A) \geq 0,
\end{aligned}$$

for any $b \in X \setminus N$, and $A \in \mathcal{A}$. We have $f(b) = 0$.

By the definition of \mathcal{A} , there exists $a \in A \cap N$ such that $\tau_B = 0$ if $a \in B \subsetneq A$. Then,

$$\mu(A) = \sum_{B \subset A \setminus \{a\}} \tau_B + \tau_A = \mu(A \setminus \{a\}) + \tau_A.$$

By Proposition 4 and $\mu(A \setminus \{a\}) \geq 0$, we have $\mu(A) > 0$, and this concludes the proof of (a).

(b) Fixing $r > 0$ and setting $A_r = \{x : f(x) > r\}$, we have:

$$\begin{aligned}
\mu(A_r) &= \sum_{B \subset A_r} \tau_B \\
&= \sum_{B \subset A_r \setminus N} \tau_B + \sum_{B \subset A_r, B \in \mathbb{I}} \tau_B \\
&= \sum_{B \subset A_r \setminus N} \tilde{\tau}_B + \sum_{A \subset A_r, A \in \mathcal{A}} \tilde{\tau}_{\{A\}} + \sum_{B \subset A_r, B \notin \mathcal{A}, B \in \mathbb{I}} \tilde{\tau}_{(B \setminus N) \cup I_B} \quad (3)
\end{aligned}$$

We remark that:

- on $B \subset N$, \tilde{f} coincides with f ,
- on a one point set $\{\iota_A\}$, $A \in \mathcal{A}$, $\tilde{f}(\iota_A) = \min_{x \in A} f(x)$, then, $\iota_A \subset A_r$ if and only if $\{\iota_A\} \subset \{\tilde{f}(y) > r\}$, and
- on $B \setminus N \cup I_B$, $y \in \{\tilde{f}(y) > r\}$ if and only if $y \in B \setminus N$ or $y \in I_B$, $\tilde{f}(y) = \min_{x \in y} f(x) > r$.

Hence, we have:

$$\begin{aligned}
(3) &= \sum_{B \subset \{\tilde{f}(y) > r\} \setminus \mathcal{A}} \tilde{\tau}_B + \sum_{A \in \{\tilde{f}(y) > r\} \cap \mathcal{A}} \tilde{\tau}_{\{A\}} + \sum_{(B \setminus N) \cup I_B \subset \{\tilde{f}(y) > r\}} \tilde{\tau}_{(B \setminus N) \cup I_B} \\
&= \mu(\{\tilde{f}(y) > r\}).
\end{aligned}$$

□

5 Conclusion

In this study, we have analyzed relations between classical and non-discrete Möbius transforms, and given equivalence conditions for null and weak null additivity for set functions on a finite set. We have also defined a method to construct a null additive set function from a general monotone measure. Along with a translated function, in the new non-additive measure space, the distribution function and the distribution function type integrals remain unchanged under this translation.

Our construction of null additive spaces is based on the argument to express the target monotone measure by using a certain σ -additive signed measure on the family of finite subsets of target space. A similar situation can be found in [2], this provides an expression method to describe a monotone measure by σ -additive (non-negative) measure on some set family space. Our problem may be improved or developed by using this idea.

References

1. E. Pap, Null-Additive Set Functions, Mathematics and Its Applications, Kluwer Academic Publishers, 1995.
2. T. Murofushi, M. Sugeno, A theory of fuzzy measures: Representations, the choquet integral, and null sets, *Journal of Mathematical Analysis and Applications* 159 (1991) 532–549.
3. J. Kawabe, The topology on the space of measurable functions that is compatible with convergence in nonadditive measure, *Fuzzy sets and systems*(Available online) (2021).
4. R. Fukuda, A. Honda, Y. Okazaki, On relaxations for the subadditivity of a monotone measure, Preprint (2021).
5. A. Honda, R. Fukuda, Y. Okazaki, Non-discrete k -order additivity of a set function and distorted measure, *Fuzzy sets and systems*(Available online) (2021).
6. R. Fukuda, A. Honda, Y. Okazaki, Constructive k -additive measure and decreasing convergence theorems, *Lecture Notes in Artificial Intelligence* 12256 (2020) 104–116.
7. G. Rota, On the foundations of combinatorial theory, I. Theory of Möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 2 (1964) 340–368.

DDoS Attack Detection in Software Defined Networks using Convolutional Tsetlin Machine

Rasoul Jafari Gohari¹, Laya Aliahmadipour², and Marjan Kuchaki Rafsanjani³

¹ Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran

`rjafari@math.uk.ac.ir`

² Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran

`l.aliahmadipour@uk.ac.ir`

³ Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran

`kuchaki@uk.ac.ir`

Abstract. Software Defined Network (SDN) has gained a huge popularity in the world of computer networks for its vast functionalities and its strength in providing scalability and maneuverability in the network. However, security challenges have always been among the top concerns that are required to be addressed. Many approaches regarding the security of SDN have been provided that seemed to be effective, approaches that utilized machine learning for detecting and preventing network attacks. Among them, the Distributed Denial of Service (DDoS) attack has always been the primary issue in terms of security and reliability. In this paper, we propose a new method for detecting DDoS attacks in the SDN environment using the Convolutional Tsetlin Machine (CTM). Moreover, the data generation process is also another considerable contribution of this work. We have been able to collect data and create an independent dataset with 20 novel features in our SDN environment in order to use it for training the model to detect three types of DDoS attacks; SYN flood attack, ICMP flood attack and UDP flood attack. Finally, the result of CTM accuracy and its performance is compared with the Convolutional Neural Network (CNN) algorithm. The result demonstrated that CTM with accuracy of 100% had the highest performance and yet the lowest memory usage compared to CNN.

Keywords: Software Defined Network · Distributed Denial of Service · Machine Learning · Convolutional Tsetlin Machine.

1 Introduction

Computer networks have always been evolving and after decades of improvement, Software Defined Network (SDN) [1] and Software Defined Wide Area Network (SD-WAN) are considered the pinnacle of this evolution by many since it enhances the network by providing numerous advantages [2] that were considered almost impossible in traditional networks. Providing a broad overview

over the network enables the network administrators to easily manage their networks via programming and cost-effective applications [3]. Moreover, the idea of a network that is mainly based on software develops the ability in the network that can create a room for growth and more novel ideas to be implemented without the unnecessary expenditure on network devices. Hence, a new idea can be implemented as an application in the network through Network Function Virtualization (NFV) [4].

In addition, one of the most important aspects of SDN is its integration with machine learning [5], which gives the network a unique and dynamic maneuverability to fend off cyber security attacks in the network. Not only the cost substantially reduces but also innovative approaches can leverage machine learning algorithms to detect network attacks, especially infamous and disruptive threats such as Distributed Denial of Service (DDoS) attacks [6] that can cause irreparable damage to SDN networks since they are heavily dependent on SDN controllers [7]. That is why creating a detection mechanism for DDoS attacks in an SDN environment is imperative.

Numerous detection approaches are proposed to detect DDoS attacks in an SDN environment [8, 9]. However, classification algorithms and detection methods have always been among the main concerns of researchers since acquiring an ideal solution is not easy. Therefore, this work focuses on a new approach that can acquire a great accuracy as well as maintaining a low memory usage. The main contributions of this work are as follows:

1. Proposing a new detection approach in SDN environments against DDoS attack
2. Using Convolutional Tsetlin Machine (CTM) for the classification of the DDoS attack
3. Providing a new dataset of 20 features in the simulated SDN environment
4. Comparing the outcome of our work with the Convolutional Neural Network (CNN) algorithm

In the following sections, we are going to discuss the preliminaries, which are required to understand the main components of an SDN network as well as DDoS attacks. Afterwards. Furthermore, we will focus on the CTM algorithm and then we will discuss how we were able to create a novel dataset. We will also deep dive into all the details in association with the attributes of our dataset and ultimately we will make a comprehensive comparison between all the other classification methods and CTM algorithm to understand each algorithm's accuracy as well as its performance.

2 Preliminaries

In this section, we will discuss the main concepts and preliminaries in order to better understand the architecture of an SDN network. Furthermore, we will discuss the DDoS attack within the SDN environment in order to comprehend the risks that pose a threat for the network.

2.1 SDN Architecture

All network devices inside a conventional network function with both the data plane and control plane [10]. This however has completely changed with the birth of SDN, which has created room for a remarkable way to provide defensive mechanisms against numerous types of attacks, especially against DDoS attacks. The architecture of an SDN network is based upon three separate planes; namely physical Plane, control Plane and management Plane. This architecture has had a great impact on security as well [11]. The physical plane is where all the devices are located. In other words, the separation of control plane and physical plane leads to a single physical plane that is meant to be for switches, routers and other network devices. The control plane on the other hand creates a room for the SDN controller so that it can have an uplifted overview of the entire network. The softwares and applications are put into the management plane so they can mount on the SDN controller for management purposes.

The communication of each plane takes place via Application Programming Interfaces (API) [12] to send and receive data flows between network devices and the SDN controller as well as data flows between the SDN controller and the NFVs in the management plane. Devices in the network send packets throughout the network back and forth via flow tables [13]. This comprehensiveness of the entire architecture allows the network engineers to shape the network through softwares to finally be able to perform some of the most challenging functionalities more easily using applications and NFVs, tasks such as load balancing, Quality of Service (QoS), security enhancement through software firewalls [14], Intrusion Detection Systems (IDS) and DDoS attack detections.

In order to detect DDoS attacks in the SDN environment, we must first understand the threats against the SDN and what parts of the network can be potentially targeted, so that the DDoS detection can be implemented more efficiently. We discuss the DDoS attack inside the SDN environment in the following section.

2.2 DDoS Attack in SDN

In spite of all the benefits that SDN provides, the growth of these types of networks has created a new paradigm for security researchers to deal with security threats such as DDoS attacks that can cripple the entire SDN and ultimately lead to unwanted risks that not only can stop the network from functioning but it can also cause detrimental damages to the network and cost an unprecedented amount of time and money for recovery [15]. That is why it is extremely paramount to detect DDoS attacks in the SDN environment.

The work of Ali et al. [16] focuses on the vulnerabilities of Openflow protocol that allows hosts to exploit this weakness and perform DDoS attacks in SDN. Their approach revolves around analyzing packet headers that are unique in a compromised host. This forces the switch to forward numerous request packets towards the controller, which as a result creates a DDoS attack. Their approach yielded a high number of true positives using MATLAB. Fouladi et al. [17]

proposed a new approach that utilizes traffic feature forecasting that is combined with chaos theory to detect DDoS attacks. Although their proposed approach is complex, the final result yielded 98.82% accuracy using MATLAB as their main tool and MAWI Working Group Traffic as their training dataset [18].

The work of Kaur and Gupta [19] focuses on optimization of the existing models. They propose a hybrid approach that takes advantage of multiple machine learning models that are combined with each other. Their approach uses these models to implement a real-time dataset. The algorithms are finally implemented in MATLAB and the final results showed higher accuracy compared to previously mentioned approaches.

In order to understand what parts of SDN network are ideal targets, we need to specify what those potential targets are. DDoS attacks can target different planes of SDN and all of them can be categorized into the followings:

1. DDoS Attack in the Data Plane: This type of attack occurs when the attacker floods the flow table of switches and devices. The ultimate purpose of the attacker is to over-flow the flow-table and cripple the data plane
2. DDoS Attack in the Control Plane: Since the controller is responsible for a broad overview of the network, the attacker can send a huge number of requests from spoofed IP addresses to the controller, which in return forces the controller to process illegitimate messages in as a result the controller delays in denies the legitimate requests
3. DDoS Attack in the Application Plane: The attack takes place in the application that is on the controller, which as a result leads to resource exhaustion which in return, legitimate users will not be able to use the proper application
4. DDoS attack for Bandwidth Exhaustion: This attack, similar to the attack in the application plane, is meant to consume all the bandwidth between the controller and the devices in the network. As a result, the depletion of bandwidth will not allow any request to respond to messages to be delivered across the network

As it can be seen, network intruders can perform DDoS attacks to cripple significant sections of the SDN environment. These security challenges in SDN need to be addressed and among the most prominent solutions lies machine learning models. In the following section we will discuss the CTM and its functionalities to perform the classification task of DDoS attack detection.

3 Convolutional Tsetlin Machine

3.1 Use Case

Training machine learning models can be an arduous task given the learning process and learning algorithms whose memory usage is significantly high. Hence, we propose using the CTM [20] for detecting DDoS attacks in the SDN, which is a less computationally intensive model. The CTM algorithm is based upon the

Tsetlin Machine (TM) [21], which is unique due to certain aspects of its nature. The most prominent area where CTM is currently used is in image classification as well as text categorization [22] and Natural Language Processing [23]. Firstly it solves complex pattern recognition problems. This can be essential for computationally intensive decisions. Moreover, it is capable of online learning, which can increase both the test and the training accuracy. Secondly, as the algorithm deals with boolean data, the memory usage decreases since this method is close to hardware language, namely binarized bits.

Since CTM algorithm is mostly used in image and text classification tasks, comparing CTM with CNN can help us understand how performant our proposed approach is since CNN is currently among the top deep learning models. Moreover, CNN produces high accuracies in classification tasks, which makes it ideal for us to understand how accurate new models are.

3.2 Scientific Notation

CTM algorithm basically leverages the TM learning process. When it comes to rule-based filters that are interpretable, the CTM uses filters with spatial dimensions $W \times W$ that uses a Z binary layers and more interestingly, the clauses of CTM take the role of filters. Each clause in the CTM algorithm is composed of $W \times W \times Z \times 2$ literals. Each literal in the TM algorithm can be defined as the two possible forms of a boolean outcome, whether it is true or false. Whatever the outcome is, TM feedback will affect the outcome based on the type of feedback. The TM utilizes two kinds of feedback: Type I and Type II. Type I feedback jointly combats false negatives and overfitting by stimulating recognition of frequent patterns. However, Type II feedback, increases the discrimination power of the patterns learnt in order to suppresses false positives. It is worth mentioning that Type I feedback is trickled down into two parts, type Ia and type Ib. Type Ia reinforces include actions to make the patterns finer. Type Ib feedback is meant to reinforce excluded actions to stop over-fitting.

As shown in the first formula below, when it comes to recognition in the CTM algorithm, the basic principles that are used in the TM algorithm are again used in CTM. For every clause in the CTM, B is used as the output of each clause per image during the image classification. Moreover, the output of a positive clause j on a patch b is denoted as $c_j^{b,+}$. Considering multiple outputs $c_j^{1,+}, \dots, c_j^{B,+}$ for clause j , we can turn this into a single input denoted as c_j^+ by using an OR operator as shown in the Eq. (1):

$$c_j^+ = \bigvee_{b=1}^B c_j^{b,+}. \quad (1)$$

4 The Proposed Method

As we discussed in previous sections, the SDN controller's ability to have a complete overview of the network has paved the way to implement machine learning

algorithms inside the SDN environment. The CTM algorithm’s capability has enabled us to propose a method with low computational power. In this section, the process of data generation and the features in the outcome dataset as well as the utilization of the CTM algorithm are discussed.

Figure. 1 illustrates the topology of our work. As it can be seen the SDN network has 6 switches and 18 hosts. Host 3 and host 14 are meant to send illegitimate data inside the network to cause DDoS attack. 3 hosts are connected to each switch and each switch is connected to the SDN controller via a datapath in order to send all the data flows to the controller. The two modules that are in the management plane are Data Gathering Module and CTM Classifier Module. Data gathering process took 2 days after creating the intended scenario. It is

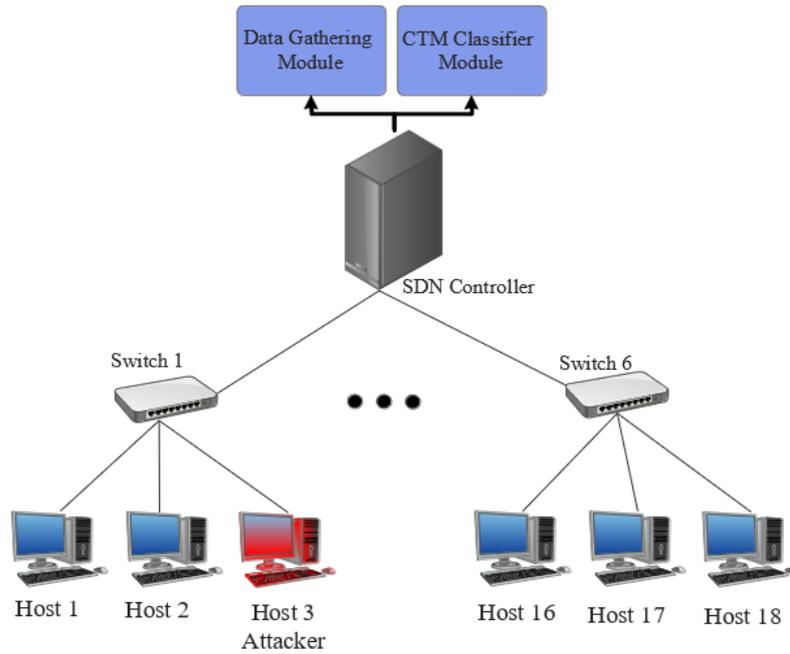


Fig. 1. The proposed topology of SDN environment under DDoS attack.

worth mentioning that the legitimate traffic was annotated as 0 and illegitimate traffic was annotated as 1. Moreover, the simulated DDoS attack covers three types of attacks, namely ICMP flood attack, UDP flood attack and SYN flood attack. We started by gathering the statistics of each flow from each switch individually through every single datapath. All the flow statistics in association with event requests and the reply handlers were collected by utilizing OFPFLOWStat-

sRequest and OFPFlowStatsReply methods in the controller. In addition, flow statistics that were related to port numbers in the network were collected by utilizing OFPPortStatsRequest method. The dataset contains 763250 records of data containing 20 features, covering ICMP flood attack, UDP flood attack and SYN flood attack. Each of these attacks contains about 250 thousand records of data in the dataset. We split the dataset into three parts in order to implement the detection on each attack separately. For the purpose of preventing overfitting, we used the 10-fold cross validation technique as well.

4.1 Generated Dataset

As shown in Table. 1, the dataset contains 20 features. We only used 16 of those features that are src_ip, dst_ip, src_port, dst_port, datapath_id, ip_protocol, flow_duration_sec, icmp_code, icmp_type, idle_timeout, hard_timeout, flag, packet_count, byte_count, packet_count_per_sec and byte_count_per_sec. As the table illustrates the details of each feature, the size and number of packets as well as timeout attributes have numerical values while attributes such as source and destination IP have categorical values. This is important to know because the input for the CTM algorithm must be a binarized dataset in the end. That is why we used binary feature-encoding methods [24] to normalize the data.

Table 1. Descriptions of the features of our dataset.

Number	Feature	Description
1	src_ips	Source IP of each host
2	dst_ip	Destination IP of each host
3	src_port	Source port number
4	dst_port	Destination port number
5	datapath_id	The datapath of the incoming flow to the controller
6	ip_protocol	The type of protocol, TCP, UDP or ICMP
7	flow_duration_sec	Duration of the flow in the network
8	flow_duration_nsec	Duration of the flow in the network in nanoseconds
9	flow_id	A combination of source IP, destination IP, source port number and destination port number
10	icmp_code	Whether ICMP is used or not
11	icmp_type	The type of ICMP used
12	idle_timeout	The idle timeout of the Host in the SDN environment
13	hard_timeout	The hard timeout of the Host in the SDN environment
14	flag	Whether or not the flow has any flags
15	packet_count	Number of packets per flow
16	packet_count_per_sec	Number of packets per flow in seconds
17	packet_count_per_nsec	Number of packets per flow in nanoseconds
18	byte_count	Size per flow in bytes
19	byte_count_per_sec	Size per flow in bytes in each second
20	byte_count_per_nsec	Size per flow in bytes in each nanosecond

5 Experimental Result

Our proposed method was implemented using an Ubuntu 20.04 Focal Fossa with 16 GB RAM and VirtualBox 6.1 to create separate Virtual Machines (VM) to complete the topology. We used Mininet for implementing the topology of our SDN environment inside a VM and Ryu as the SDN controller inside another separate VM. We used Kali Linux as a VM to generate DDoS attack traffic. Inside Kali we used hping and mgen to generate the UDP, ICMP and SYN flood traffic along with normal traffic in 3 days. Table. 2 shows the summary of our infrastructure.

Table 2. Parameters of the our Infrastructure.

Number	Parameter
1	Main Operating System: Ubuntu 20.04
2	Hardware: Intel Corei7-4720HQ and 16 GB RAM
3	Virtualization Technology: VirtualBox 6.1
4	SDN Controller: Ryu Controllerr
5	Protocol: Openflow
6	DDoS Traffic Generator: Kali Linux, mgen, hping
7	Emulator: Mininet

5.1 Performance Parameters

In this work we used Accuracy as well as Memory Usage to measure the performance of our model. Table. 3 demonstrates the final result of our model. We split the dataset into three separate parts, namely ICMP flood attack, UDP flood attack and SYN flood attack. As it can be seen in Table. 3, CTM has gained the highest accuracy in all three attacks and yet it shows far less memory usage. CNN uses almost three times the memory for gaining less accurate outcomes compared to CTM.

Table 3. Performance of the Proposed Model compared to CNN.

Algorithm	Dataset	Accuracy (%)	Memory Usage (MB)
CTM	ICMP	99.98	205.29
	UDP	100	205.15
	SYN	99.98	319.60
CNN	ICMP	99.91	599.71
	UDP	99.88	599.26
	SYN	99.97	731.04

6 Conclusion and Future Work

SDN environments and SDN architecture are on the rise. This signifies the importance of cyber security among these types of networks. As discussed, many researchers have proposed different solutions to make sure SDN environments are capable of detecting these threats. In this paper, we proposed a new method for detecting DDoS attacks within the SDN environment where the network was under three of the most notorious DDoS attacks; ICMP flood attack, UDP flood attack and SYN flood attack. We also contributed a new dataset within the SDN network that consisted of more than 750 thousand rows of data, which was finally used by CTM algorithm to classify legitimate traffic from illegitimate DDoS traffic in the network. In the end the results demonstrated that the CTM algorithm has the highest accuracy while utilizing the lowest memory usage.

Since this algorithm has shown promising results on both accuracy and memory usage, we will focus on creating more records of data in order to perform the same attack on a bigger dataset. This allows us to see how the CTM algorithm will perform on much bigger datasets and how much memory will the algorithm require when it comes to large datasets. This can also provide insight in order to see if the algorithm is capable of working in real-time situations. In addition, we will test the results with a benchmark dataset in order to guarantee the quality of the dataset that we created.

Moreover, since this algorithm has shown substantial results for optimized memory consumption, we will also focus on using CTM algorithm on wireless networks, especially Mobile Ad-hoc Networks (MANETs) and Vehicular Ad-hoc Networks (VANETS) where wireless devices are in need of such an ideal algorithm to decrease their computational resources. In that case, the mobility of such networks will be much more operable under severe conditions.

In addition to those, CTM algorithm can also be used in other types of networks as well. We hope to use the same algorithm in other types of networks and compare our results with other machine learning models. This helps us widen our horizon when it comes to new algorithms such as CTM.

References

1. Greenberg, A., Hjalmtysson, G., Maltz, D.A., Myers, A., Rexford, J., Xie, G., Yan, H., Zhan, J., Zhang, H.: A clean slate 4D approach to network control and management, *SIGCOMM Comput. Commun. Rev.* 35, 5 , 41–54. (2005)
2. Kim, H., Feamster N.: Improving network management with software defined networking, *IEEE Communications Magazine*, 51, 114-119, (2013)
3. Lin, T., Kang, J., Bannazadeh, H., Leon-Garcia A.: Enabling SDN applications on Software-Defined Infrastructure, *IEEE Network Operations and Management Symposium (NOMS)*, 1-7, (2014)
4. Papavassiliou, S.: Software Defined Networking (SDN) and Network Function Virtualization (NFV)., *Future Internet*, 12, 7. (2020)
5. Nanda, S., Zafari F., DeCusatis C., Wedaa E., Yang B.: Predicting network attack patterns in SDN using machine learning approach, 2016 IEEE Conference on Net-

- work Function Virtualization and Software Defined Networks (NFV-SDN), 167-172, (2016)
6. Banitalebi Dehkordi, A., Soltanaghaei, M., Boroujeni, F.Z.: The DDoS attacks detection through machine learning and statistical methods in SDN, *J Supercomput* 77, 2383-2415 (2021)
 7. Revathi, M., Ramalingam, V.V., Amutha, B.: A Machine Learning Based Detection and Mitigation of the DDOS Attack by Using SDN Controller Framework, *Wireless Pers Commun*, (2021)
 8. Novaes, M. P., Carvalho, L. F., Lloret, J., Proença, M. L.: Adversarial Deep Learning approach detection and defense against DDoS attacks in SDN environments, *Future Generation Computer Systems*, 125, 156-167, (2021)
 9. Deepa, V., Sudar K. M., Deepalakshmi, P.: Detection of DDoS Attack on SDN Control plane using Hybrid Machine Learning Techniques, 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), 299-303, (2018)
 10. Karakus, M., Durresi A.: A survey: Control plane scalability issues and approaches in Software-Defined Networking (SDN), *Computer Networks*, 12, 279-293, 1389-1286, (2017)
 11. Ganzha, M., Maciaszek, L., Paprzycki, M.: Position Papers of the 2014 Federated Conference on Computer Science and Information Systems, *ACSIS*, 3, 143-148, (2014)
 12. Zhou, W., Li, L., Luo M., Chou, W.: REST API Design Patterns for SDN Northbound API, 2014 28th International Conference on Advanced Information Networking and Applications Workshops, 358-365, (2014)
 13. Kuźniar, M., Perešini, P., Kostić, D.: What You Need to Know About SDN Flow Tables. Mirkovic, J., Liu, Y. (eds) *Passive and Active Measurement*, 8995, (2015)
 14. Krongbaramee, P., Somchit, Y.: Implementation of SDN Stateful Firewall on Data Plane using Open vSwitch, 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE), 1-5, (2018)
 15. Ribin, J. S. B., N., Kumar: Precursory study on varieties of DDoS attacks and its implications in Cloud Systems, 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 1003-1008, (2019)
 16. Granmo, O. C., Glimsdal, S., Jiao, L., Goodwin, M., Omlin, C., Berge, G: The Convolutional Tsetlin Machine, (2019)
 17. Ali, S., Alvi, M. K., Faizullah, S., Khan, M. A., Alshantqiti, A., Khan, I: Detecting DDoS Attack on SDN Due to Vulnerabilities in OpenFlow, 2019 International Conference on Advances in the Emerging Computing Technologies (AECT), 1-6, 2020
 18. Fouladi, R. F., Ermis, O., Anarim, E: A DDoS attack detection and defense scheme using time-series analysis for SDN, *Journal of Information Security and Applications*, 54, 2020
 19. MAWI Working Group Traffic Archive, <http://mawi.wide.ad.jp/mawi>, Last accessed 24 May 2022
 20. Kaur, G., Gupta, P: Hybrid Approach for detecting DDOS Attacks in Software Defined Networks, 2019 Twelfth International Conference on Contemporary Computing (IC3), 1-6, 2019
 21. Granmo, O. C.: The Tsetlin Machine - A Game Theoretic Bandit Driven Approach to Optimal Pattern Recognition with Propositional Logic., (2018)
 22. Berge, G. T., Granmo, O. C, Tveit, T. O., Goodwin, M., Jiao, L., Matheussen and B. V.: Using the Tsetlin Machine to Learn Human-Interpretable Rules for High-Accuracy Text Categorization With Medical Applications. *IEEE Access*, 7, 115134-115146, (2019)

23. Saha, R., Granmo, O. C., Goodwin, M.: Using Tsetlin Machine to discover interpretable rules in natural language processing applications, *Expert Systems*, (2021)
24. Uchida, S., Image processing and recognition for biological images. *Develop. Growth Differ.*, 55, 523-549, (2013)

Using machine learning and natural language processing for fake news detection in WhatsApp

Leonardo Miranda de Brito^{1,2}[0000-0002-0598-5115], Marilia Caroline de A. C. Silva^{1,2}, Natália Oliveira^{1,2}, Tuby D. Neto^{1,2}, Nicksson Ckayo Arrais de Freitas²[0000-0001-6693-5886], and Tiago Da Silva Vinuto²

¹ Centro de Informática – Universidade Federal de Pernambuco (UFPE), Brazil

{lmb3,mcacs,ntso,tdn2}@cin.ufpe.br

² SiDi, Recife, PE, Brazil

{nicksson.a,t.vinuto}@sidi.org.br

Abstract. The progressive propagation of misinformation on a large scale, volume and velocity through social networks has become an imminent problem, impacting the legitimacy of information in social, democratic and public health instances. Nowadays, one of the most popular digital tools by the Brazilian population is WhatsApp. WhatsApp has become an easily accessible vehicle for generating disinformation in Brazil since it contains millions of users worldwide, and any citizen has the autonomy to publish. Intelligence systems became an alternative to minimize the damage caused by the proliferation of misinformation. In this sense, research on methods for detecting misinformation on WhatsApp for messages written in Brazilian Portuguese is still scarce among the scientific community. In this paper, we propose a new approach to classify information and misinformation in WhatsApp. We conducted extensive experiments using machine learning methods and natural language techniques from FakeWhatsApp.Br. We achieved a more significant F1-score considering the previous approaches in the literature, up to 14%.

Keywords: Fake News · Machine Learning · Natural Language Processing · Classification

1 Introduction

Nowadays, people generally prefer to use social networks rather than traditional media as a source of information because social networks represent a more dynamic and cheaper way to spread the news and are faster and easier to share and discuss among friends than television or printed newspapers [12].

Despite the advantages of social media, the quality of the news shared is lower than that presented in traditional news organizations, such as newspapers, due to the lack of review and curation of the propagated content. This wide sharing makes the distribution of the so-called fake news much greater, that is, news whose content is false and/or without foundation, which aims to deceive and manipulate the reader, besides producing financial or political gains. Situations

like these, can negatively influence the authenticity of other news channels, as well as deliberately persuade those who consume them to accept false beliefs and change the way real news is perceived by the public [14,12].

As an attempt to minimize the negative impact of fake news, it is necessary to develop methods that help to detect them on social media, including messaging applications such as WhatsApp. In this context, some works were found in the literature in this scope, including [4], which aims to detect Fake News on Brazilian WhatsApp using Natural Language Processing techniques, however, the best F1-score achieved was 0.7330. In addition, it is worth noting that many similar articles have been identified, but most of these deal with other languages or other social networks, for instance, [5] deals with this type of classification, but focus on posts from the social network Twitter, getting a slightly better result with an F1-score of 0.7400.

In this paper, we propose an approach to detect whether a text message written in Portuguese-Brazilian language from WhatsApp groups contains misinformation or not, focusing on achieving a higher F1-score than those found in the literature. In order to achieve this goal we applied pre-processing, text vectorization, data augmentation and attribute selection techniques to prepare the data for further application in machine learning models. This approach, was applied to a database of messages collected from public groups of WhatsApp, labeled as containing misinformation or not. The experiments performed are available at ³.

The next sections are organized as follows: in Section 2, related works are presented and the differentials of the present paper are detailed; in Section 3, the methodology used is explained, more specifically, the Database and the performed processes - Pre-Processing, Vectorization, Data Augmentation, Attribute Selection, Application of Machine Learning Models and Evaluation Metrics - were detailed; And, in Section 4, the results and discussions of the experiment are presented and in Section 5 conclusions and possible future works are presented.

2 Related Works

In the literature, we investigated works that describe machine learning methods for fake news classification in Portuguese language using fake news disseminated in social networks and instant messaging applications. We highlight that few articles were found considering the context of fake news in Brazil and the messaging application WhatsApp. These papers are presented below:

Initially we highlight [13], in which the authors describe a series of analysis on machine learning methods for fake news detection applied to the Fake.Br corpus, which is a dataset composed by manually labeled news collected from several news sites, this dataset has 7,200 news (3,600 fake news and 3,600 legitimate news) [11]. In this work, a Vectorization process was applied, where linguistics-based attributes, Bag of Words, Word2Vec and FastText were used. Also in the

³ https://github.com/Giganoide01/FakeNews_Wpp_PtBr_Classifier

paper, the classification was performed using the following algorithms: logistic regression, support vector machines, decision trees, random forest, aggregation by bootstrap and adaptive boosting. The best result achieved by the authors was obtained by stacking logistic regression models for Bag of Words with linguistic-based attributes, achieving an F1-score of 0.9710.

On the other hand, the authors of the work presented in [5] describe an example of automatic detection of fake news in social network texts. In this paper, it is reported the construction of a dataset containing fake and true news extracted from Twitter and the application of machine learning models to classify this news. To do this, the authors used the unigram and bigram method with bag of words, along with several machine learning methods. The best performance obtained in this work using the Complement Naive Bayes algorithm was an F1-score of 0.7400.

In the scope of automatic classification of fake news spread by WhatsApp, the authors of the paper in [10] describe the development of a classifier of fake news about COVID-19 collected from messages on WhatsApp. For that, the authors used an architecture based on the BiLSTM neural network (Bidirectional Long-Short Term Memory) and obtained an F1-score of 0.8340.

Finally, the authors of the paper presented in [4] describe a database of true and fake news collected from messages in WhatsApp groups and manually annotated (FakeWhatsApp.Br⁴). To detect fake news, the authors used Bag-Of-Words (BOW) and Term Frequency-Inverse Document Frequency (TF-IDF) along with classical machine learning methods. The results obtained by the authors show that due to the peculiarities of WhatsApp messages, mainly the predominance of short messages followed by media files (audio, image or video), purely NLP-based methods using traditional vectorization features have limited performance. The result obtained by the authors was a F1-score of 0.7330 for the combination TF-IDF and Linear Support Vector Machine. The error analysis presented by the authors shows that the classification errors are mainly due to the predominance of short texts accompanying the media files. When filtering out texts with less than 50 words, the F1-score rose to 0.8700.

Our work also used the same dataset used in [4]. However, the difference between this work and ours is that we evaluated new methods of processing and data handling on the dataset, such as a vectorization technique that considers the semantic context of words (Doc2Vec), data augmentation, attribute selection for TF-IDF vectors, and the evaluation of machine learning-based classification models. With this, we hope that these techniques can result in an improved approach to the classification of fake news in Whatsapp messages.

3 Proposed Approach

In this paper, we present our approach to the proposed problem and in Figure 1 we have an overview of it. We performed an extensive experimentation ap-

⁴ <https://github.com/cabrau/FakeWhatsApp.Br>

proach using FakeWhatsApp.Br as our case study. The methods used during the development of the approach are described in the subsections below.

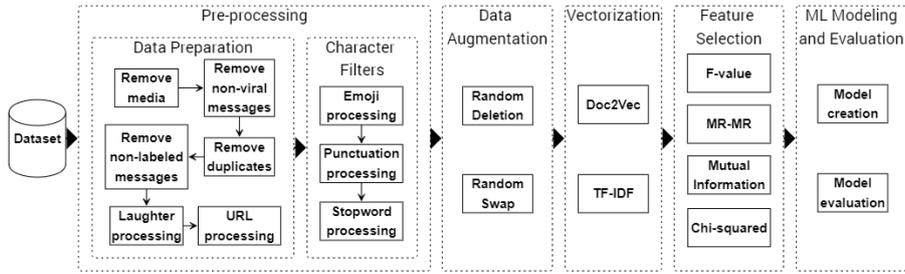


Fig. 1: Overview of the methodology used.

3.1 Database

The large-scale database containing messages from the app WhatsApp in Brazilian Portuguese was developed for the research and study reported in [4]. This database is composed of anonymous messages sent on WhatsApp in public groups collected in 2018; these messages include conversations, opinions, humorous and satirical texts, prayers, commercial offers, news, short texts, *emojis* and others.

For job comparison issues, we use the same bank made available at ⁵, which presents 4588 samples (2041 messages containing disinformation and 2547 messages without disinformation).

3.2 Pre-Processing

For data preparation, the same steps were followed as in the paper [4]. Regarding the preparation pipeline, respectively: samples that contained media, that is, audios, videos and images, were removed, keeping only those that exclusively presented textual content (1); records that did not go viral among the WhatsApp groups were excluded (that is, showed up two times in two different groups) (2); instances that had duplicate messages were deleted (3); the samples in which it was not possible to detect if they are Fake News or Real News were removed - in the database, there is a feature called “Misinformation”, which is equivalent to 0 when the news is true, 1 for Fake News and -1 for instances where this detection was not performed (disregarded for the analysis) (4); Subsequently, the laughs with varying amounts of the letter “k” were all transformed into “kkkk” (5); and, all links have been shortened to the website domain, discarding the specific protocols and directories that are usually present in web addresses (6).

⁵ <https://github.com/cabrau/FakeWhatsApp.Br>

Finally, one of the distinguishing factors of this work is the pre-processing applied that generated 6 subsets of the text messages through different character filters, in order to find out how much these variations would improve the performance of the machine learning models applied later. Namely, in the generated subsets: emojis, punctuation and stop words were preserved (1); stop words were removed, but emojis and punctuation were preserved (2); stop words and punctuation were removed, but emojis were preserved (3); stop words and emojis were removed, but punctuation was preserved (4); emojis and punctuation were removed, but stop words were preserved (5); and, emojis, punctuation and stop words were removed (6).

3.3 Vectorization

For this work, We use two vectorization techniques: TF-IDF and Doc2Vec. TF-IDF is a numerical measure of statistics that expresses how important a word is to a document. It is a frequency-based method that is not as naive as the Bag of Words (BoW), since it reduces the importance of words that are frequently repeated, such as articles, prepositions, and conjunctions are not considered to be more important than less frequent but important words [1]. We chose this method as our count-based vectorization method due to its better general performance in comparison to the BoW [3]. Also, we used the uni-bi-trigram extraction because it was the n-gram pattern that yielded the best results in the work of [4].

The other technique we use is Doc2vec, an NLP tool that serves to represent documents in the form of vectors, being a generalization of the word2vec model [9] - which creates numerical vectors for each word, considering semantic relationships between the words of the document, which are called attribute vectors. Some of those semantic relationships between words are synonyms, antonyms, or analogies. We chose this technique to obtain a representation that could encapsulate the semantic relationships between the words present in the documents, in order to try to increase the performance of the system.

3.4 Data Augmentation

One of the distinguishing factors of this work is the use of the Data Augmentation technique [16], that is, the number of records is increased without collecting more messages in other WhatsApp groups, but performing operations on the data that were already present in the database. In other words, the volume of information has been artificially increased.

In this context, we chose to apply two different strategies: Random Deletion, which consists of randomly removing each word from the sample with a given probability p ; and Random Swap, in which two words are swapped in document n times, with n being a parameterized integer value [15].

3.5 Feature Selection

Next, another addition in comparison to the baseline work presented in [4] was the application of feature selection methods, such as: F-value, Maximum Rel-

evance - Minimum Redundancy (MR-MR) [6], mutual information and Chi-Squared. Those techniques were applied in order to select the most relevant features according to the target variable.

For the TF-IDF sparse feature matrix, we use Chi-squared and mutual information as our feature selection methods due to their simple implementation and their efficiency to deal with TF-IDF sparse matrices [8,3]. For the Doc2Vec feature matrices we use the ANOVA F-value and MR-MR because they are more adequate to dense numeric feature matrices [3].

Additionally, after every feature selection method we applied the removal of perfect collinearity, where attributes are excluded if there is an exact linear relationship between two features. This allows the mitigation of redundancies prior to the development of the models.

3.6 Machine Learning Models

For the experiments with the machine learning algorithms, the dataset is split into 80% for training/validation and 20% for test. Validation is performed using the stratified 10-fold cross-validation method. The evaluation metric chosen was the F1-score, as it is the same used by [4], since our partial goal in this work is to compare the current results with those obtained by the former author.

The library used for the experiments was *pycaret* [2], which has a list of 16 classifiers that encompass different types of machine learning algorithms. For each experiment, the test of the best model was performed. At the end of the experiments, the best model among all underwent a parameter adjustment using random search and grid search methods in order to obtain the best possible result.

3.7 Evaluation Metrics

In order to evaluate the performance of the machine learning models, we made use of accuracy, precision, recall, F1-score, and AUC. The main metric for evaluation was the F1-score, due to its reliability as it needs the precision and the recall to be high to have a high value too [7]. Also, it was the most frequently used metric in the papers from the Related Works (Section 2).

4 Results and Discussions

Four experiments were carried out to evaluate the performance of different combinations of operations for the framework of the classification model, a summary of those experiments is presented in Figure 1.

4.1 Doc2Vec Vectorization

We started the experiments by applying the Doc2Vec vectorization to a dataset containing only the relevant words (without stopwords, punctuation, and emojis) and without data augmentation. The result obtained indicates that the algorithms based on boosting performed better, although, the performance was

poor since it did not reach an F1-score of 0.6000 during the test, as shown in the first sub-table of Table 1.

In order to identify whether this performance could be improved with more samples, we repeated the configuration of the previous experiment, with the difference that data augmentation was applied to the training/validation set, with $p = 0.5$ and n being half the string’s length. The results of the metrics for validation indicated an improvement in performance, however, the test indicated that the data augmentation may have caused an over-fitting scenario, as shown in the second sub-table of Table 1.

Other attempt in improving the performance of the Doc2Vec based methods was to apply feature selection methods, we tried the ANOVA F-value (third sub-table of table 1) and MR-MR (fourth sub-table of table 1). But none of those strategies enhanced the performance of the F1-score, with both achieving similar values as those achieved before, 0.6293 and 0.5843 in test, respectively.

4.2 TF-IDF Vectorization

We also experiment with the TF-IDF vectorization plus feature selection methods (mutual information and Chi-squared), as it was not something we have seen reported in the Related Works section 2. In order to obtain a fair comparison, the same pre-processing was used as in the four previous experiments (stopword, emoji, and punctuation removal).

The vectorization by TF-IDF produced a matrix with 293577 attributes. In order to reduce the amount of data, we apply the Chi-squared test to select the 10000 best attributes and then remove the attributes with perfect collinearity, resulting in 3790 attributes at the end. The best result obtained by this approach was through the Gaussian Naive Bayes classifier, which is superior to previous experiments both in validation and in testing, as shown in the fifth sub-table in Table 1.

We also experimented with the mutual information selection method. After the selection process, 6310 features were selected. The results obtained by this pipeline were inferior to those obtained by the Chi-squared selection, although, were far superior from those obtained by the Do2Vec vectorization and similar to the result obtained in [4], as shown in the sixth sub-table of Table 1.

In order to identify whether punctuation, emojis and stopwords could provide some additional information to the models, hence enhancing their performance according to the metrics, we repeat the process of TF-IDF vectorization, Chi-squared feature selection, and model training for texts containing punctuation, emojis, and stopwords.

The vectorization by TF-IDF results in 376296 attributes, after the selection of the top 10000 attributes through the Chi-squared test and removal of attributes with perfect collinearity, 4731 attributes remained. The results obtained by the Gaussian Naive Bayes classifier were slightly better in validation and test when compared to the fifth experiment, as shown in seventh sub-table of Table 1. It is possible to notice that the test performance was superior to the

validation, which may be caused by luck in the division of the training and test sets, or lack of tuning of model parameters.

Table 1: Table containing the seven pipelines executed along this project. It shows metrics of the top-3 models for the validation step and the test of the best validation model for each experiment, according to the F1-score, with the best results amongst all in bold.

Pipeline	Evaluation	Model	Accuracy	Recall	Precision	F1-score
A	Validation	lightbm	0.7128	0.6026	0.7085	0.6512
		catboost	0.7183	0.5824	0.7312	0.6481
		xgboost	0.6975	0.5971	0.6842	0.6374
B	Validation	lightbm	0.6795	0.5392	0.6687	0.5970
		knn	0.8735	0.7390	0.8767	0.8019
		xgboost	0.7880	0.7015	0.7978	0.7464
C	Validation	catboost	0.7847	0.6529	0.8270	0.7296
		knn	0.6500	0.6500	0.6400	0.6300
		xgboost	0.7136	0.5805	0.7225	0.6432
D	Validation	lightbm	0.7025	0.5971	0.6927	0.6410
		catboost	0.7046	0.5848	0.7021	0.6377
		xgboost	0.7048	0.5637	0.7121	0.6293
E	Validation	catboost	0.7123	0.5928	0.7133	0.6472
		lightbm	0.7117	0.5744	0.7215	0.6394
		xgboost	0.6951	0.5903	0.6823	0.6324
F	Validation	lightbm	0.6590	0.5392	0.6377	0.5843
		nb	0.8782	0.8512	0.8725	0.8613
		svm	0.8044	0.7182	0.8296	0.7630
G	Validation	ridge	0.7940	0.6571	0.8459	0.7391
		nb	0.8671	0.8064	0.8844	0.8436
		et	0.7790	0.7006	0.7811	0.7383
A	Test	et	0.7876	0.7108	0.7902	0.7484
		rf	0.7700	0.6675	0.7839	0.7205
		svm	0.7684	0.6539	0.7989	0.7138
B	Test	et	0.7876	0.7108	0.7902	0.7484
		nb	0.8883	0.8965	0.8590	0.8772
		svm	0.7978	0.6760	0.8427	0.7476
C	Test	ridge	0.7905	0.6455	0.8476	0.7327
		nb	0.9074	0.8897	0.9007	0.8952
		svm	0.7905	0.6455	0.8476	0.7327
Description						
A	Stopword, emoji and punctuation removal + Doc2Vec					
B	Stopword, emoji and punctuation removal + Doc2Vec + Data augmentation					
C	Stopword, emoji and punctuation removal + Doc2Vec + F-value					
D	Stopword, emoji and punctuation removal + Doc2Vec + MR-MR					
E	Stopword, emoji and punctuation removal + TF-IDF + Chi-squared					
F	Stopword, emoji and punctuation removal + TF-IDF + Mutual information					
G	Keep stopwords, emojis and punctuations + TF-IDF + Chi-squared					

4.3 Parameter Fine-tuning

The best result was obtained by the Gaussian Naive Bayes classifier in the last experiment, therefore, this model is chosen to undergo fine-tuning. Considering

that other similar Naive Bayes models usually perform well in this type of data, Complement and Multinomial Naive Bayes classifiers are also evaluated. After two rounds of fine-tuning, we found, through the F1-score for validation, that the best model was the Naive Bayes Gaussian with variance smoothing equal to 0.0005, as shown in Table 2.

As shown in Table 3, our model performed better in the task of classifying fake news from Whatsapp messages written in Brazilian Portuguese than the models reported in [4]. This difference can be attributed to the feature selection, given that this was the main difference between the F1-score of 0.8990 obtained in this paper and the 0.7330 obtained in the article [4], in which the author used only the 5000 most frequent attributes in the entire vocabulary to train the classifiers.

Additional information about the performance of the model is provided in Figure 2. From Figure 2b it is noticeable that the model has a slightly stronger tendency to let messages with misinformation pass as non-misinformation with 13% being wrongfully labeled as such. From Figure 2a we can observe the model’s separation ability through the AUC since it is greater than 0.9.

Table 2: Evaluation of the three Naive Bayes models that underwent two rounds of fine-tuning.

	Model	Accuracy	Recall	Precision	F1-score
Validation	Gaussian NB	0.9033	0.8990	0.8855	0.8921
	Complement NB	0.8480	0.9467	0.7668	0.8472
	Multinomial NB	0.8886	0.9149	0.8471	0.8795
Test	Gaussian NB	0.9129	0.8725	0.9271	0.8990

Table 3: Metrics comparison between the framework developed here and in the paper from [4]

Approach	Recall	Precision	F1-score
Ours	0.8720	0.9270	0.8990
[4]	0.7500	0.7170	0.7330

4.4 Error Analysis

In order to understand which text features were causing the model to commit errors, we analyzed the difference in frequency of occurrence of tokens between texts that were misclassified and correctly classified. We noticed that the terms most associated with Type I Errors were “lgbt”, “as fraudes” (the frauds) and

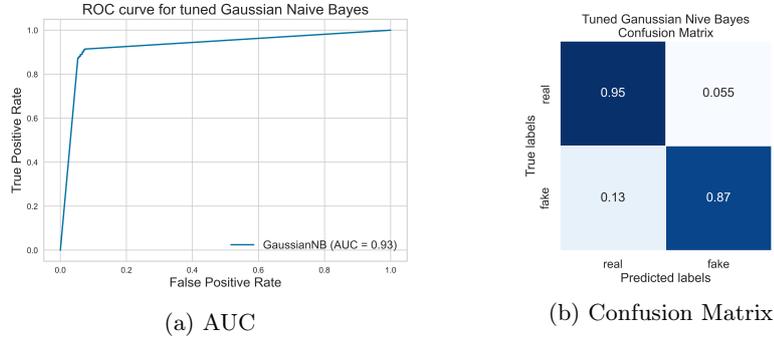


Fig. 2: AUC and Confusion Matrix for test performance of the tuned Gaussian Naive Bayes model.

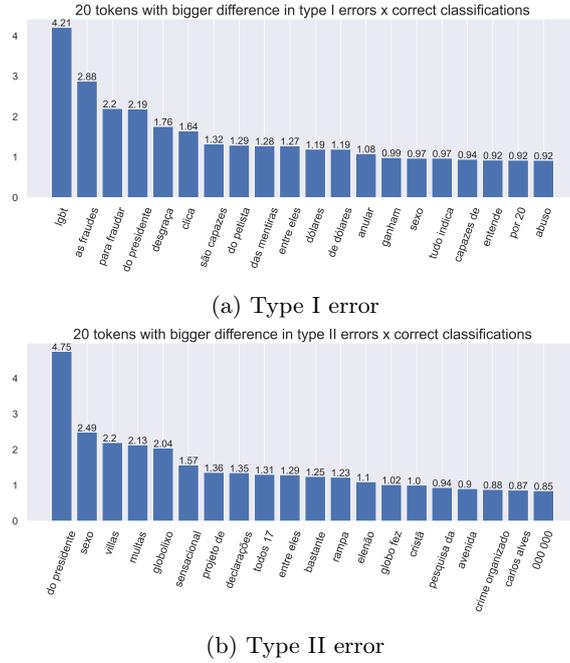


Fig. 3: Bar graph of the percentages of the twenty main attributes (words in Portuguese) with the greatest difference in the frequency of occurrence between texts misclassified by type I (a) and type II (b) errors and texts classified correctly.

“para fraudar” (to defraud) (Figure 3a), which indicates that the model associated these terms with the occurrence of fake news whenever they occur in the text, independently of the other words it contained.

By this same metric, it is possible to notice that the terms most associated with type II errors were “do presidente” (from the president) and “sexo” (sex) (Figure 3b), which were representative of political and moral agendas. This type of error may indicate that the model had difficulty distinguishing a true communication associated with the President of the Republic from a false speech or campaign piece associated with him.

5 Conclusions and Future Work

Our work contributes with an approach that has superior performance in the fake news classification task in the FakeWhatsApp.Br database, and with the exploration and evaluation of data science methods applied to this database, which can contribute to other fake news classification tasks in the Portuguese-Brazilian language. Achieving the goal of overcoming the F1-Score value reported so far in the literature for the FakeWhatsApp.Br database. We accomplished this objective through an extensive exploration of combinations of pre-processing methods, such as data augmentation, vectorization, feature selection and machine learning model evaluation.

At the end of the process, we identified that the result with the highest F1-Score was obtained by the approach composed of: a pre-processing that kept stopwords, emojis and punctuation; vectorization by TF-IDF, attribute selection by Chi-squared test with perfect collinearity removal, training of a Gaussian Naive Bayes classification model and parameter tuning by Random Search and Grid Search. The F1-Score obtained was 0.8990, which represents a 16.6% increase in F1-Score over that obtained previously in the literature for the classification of fake news from the FakeWhatsApp.Br dataset [4].

However, we emphasize that, despite being successful, the proposed approach can be modified to become even more robust, with the exploration of more data augmentation methods, and also with the addition of other methods such as random insertion and synonyms swap in combination with random swap and random deletion, which may be able to improve the performance of classifiers on small databases [15]. Other improvements that can be made and that we intend to perform as future work are: to make use of embedding vectorizations, and to evaluate deep learning methods in future trials of the approach.

Acknowledgement

The results presented in this paper, Using machine learning and natural language processing for fake news detection in WhatsApp have been developed as part of a project between SiDi, financed by Samsung Eletrônica da Amazonia Ltda., under the auspices of the Brazilian Federal Law of Informatics no. 8248/91.”

References

1. Aizawa, A.: An information-theoretic perspective of tf-idf measures. *Information Processing & Management* **39**(1), 45–65 (2003)

2. Ali, M.: Pycaret: An open source, low-code machine learning library in python. PyCaret version **2** (2020)
3. Bengfort, B., Bilbro, R., Ojeda, T.: Applied text analysis with python: Enabling language-aware data products with machine learning. " O'Reilly Media, Inc." (2018)
4. Cabral, L., Monteiro, J.M., da Silva, J.W.F., Mattos, C.L., Mourao, P.J.C.: Fake-whastapp. br: Nlp and machine learning techniques for misinformation detection in brazilian portuguese whatsapp messages (2021)
5. Cordeiro, P.R., Pinheiro, V.: Um corpus de notícias falsas do twitter e verificação automática de rumores em lingua portuguesa. In: STIL-Brazilian Symposium in Information and Human Language Technology. IEEE, Salvador, BA, Brazil. pp. 220–228 (2019)
6. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology* **3**(02), 185–205 (2005)
7. Géron, A.: Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. " O'Reilly Media, Inc." (2019)
8. Haddi, E., Liu, X., Shi, Y.: The role of text pre-processing in sentiment analysis. *Procedia computer science* **17**, 26–32 (2013)
9. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196. PMLR (2014)
10. Martins, A.D.F., Cabral, L., Mourao, P.J.C., Monteiro, J.M., Machado, J.: Detection of misinformation about covid-19 in brazilian portuguese whatsapp messages using deep learning. In: Anais do XXXVI Simpósio Brasileiro de Bancos de Dados. pp. 85–96. SBC (2021)
11. Monteiro, R.A., Santos, R.L., Pardo, T.A., Almeida, T.A.d., Ruiz, E.E., Vale, O.A.: Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In: International Conference on Computational Processing of the Portuguese Language. pp. 324–334. Springer (2018)
12. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter* **19**(1), 22–36 (2017)
13. Silva, R.M., Santos, R.L., Almeida, T.A., Pardo, T.A.: Towards automatically filtering fake news in portuguese. *Expert Systems with Applications* **146**, 113199 (2020)
14. Tandoc Jr, E.C., Lim, Z.W., Ling, R.: Defining “fake news” a typology of scholarly definitions. *Digital journalism* **6**(2), 137–153 (2018)
15. Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)
16. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation **34**(07), 13001–13008 (2020)

DANKFE: a domain knowledge based algorithm for feature engineering

Tiago Afonso, Cláudia Antunes

Instituto Superior Técnico, Av. Rovisco Pais 1 Lisboa, Portugal
tiago.francisco.a, claudia.antunes@tecnico.ulisboa.pt

Abstract. The research on the automation of the knowledge discovery process, usually known as AutoML, has been pursued in the last few years. There methods are one of the most promising ways for democratizing the use of data science. Its success will allow data science to be used as a universal tool over any data from any domain, available to anyone. However, its improvement is threatened by the difficulties on the automation of the data preparation step. Being the longest of the process stages, it requires domain knowledge to enrich the data, and hopefully improve models' results. In this paper, we propose DANKFE, a new algorithm for feature generation based on domain knowledge, represented through extended entity-relationship (EER) diagrams. It works by efficiently exploring the domain knowledge available regarding the data, and using a set of operators to increase the feature space and consequently increase the performance of the learnt models. The algorithm is validated over a COVID-19 dataset, comparing the models learnt with and without automatic generated features, along with its performance study.

Keywords: Feature Engineering · Feature Generation · AutoML · Domain Knowledge · Entity-Relationship diagrams.

1 Introduction

Throughout the years, the amount of data that is collected and processed has increased exponentially. Treating data manually has become intractable, which is why development in machine learning has also increased massively. In this era of Big Data, the more data that can be processed by a system, the better the information that can be retrieved from it and the more robust it can get [4]. Processes for turning raw data into functional knowledge have been defined and refined into what is known nowadays as *Data Science*. This growth is leading to the use of data science and machine learning in more and more domains, as every industry races to use data-driven approaches to find the best insights. This is leading to a large demand for data scientists that have experience in such domains and in handling large amounts of data [23]. To counteract this, research is inclining towards the automation of the data science pipeline in order to be able to gather valuable insights without the need for human intervention.

These Automated Machine Learning (AutoML) tools can be a solution to the high demand and low supply of data scientists, and are already tackling important parts of the pipeline, such as model selection and hyperparameter optimization. However, in the area of feature engineering, and especially feature generation, current AutoML frameworks are still lacking in automation, employing only simple or no solutions. These frameworks also try to remain domain agnostic. They frequently employ black-box approaches that work for a variety of data types and datasets, but can also cause some mistrust among data scientists in AutoML [26]. Nevertheless, harnessing domain knowledge can improve the performance of these frameworks, as models tend to be more robust with useful information [1].

In this paper, we propose an algorithm for the automation of the feature generation stage, making use of domain knowledge represented in the form of an extended entity-relationship (EER) diagram. These diagrams have been the *de facto* standard for representing databases' logic models, and are plenty available and easy to define. From a dataset containing the original features describing the problem and the corresponding EER diagram, the algorithm generates a new feature for each relation in the diagram and computes its values for each record in the dataset.

In order to validate our algorithm, we study both its efficacy and efficiency. First, we measure efficacy by comparing the performance of similar classifiers trained over the original dataset and over the dataset enriched by the features generated by the new algorithm, and then against the classifiers trained through auto-sklearn [8]. Second, we compare the time spent with each one of those approaches to assess their relative efficiency. Experimental results show an increase in model performance with the generation of features, with negligible time spent, especially when compared to an AutoML framework.

The rest of the paper is structured as follows: next (section 2), we provide some background on the automation of the knowledge discovery process, presenting a brief overview of automatic feature generation procedures. Following this, in section 3, our algorithm is described, addressing its main advantages and difficulties. Section 4 evaluates the new algorithm, either its efficiency and efficacy, by comparing the models learnt over the original data, and datasets enriched with the features automatically generated. The paper concludes in section 5, where a summary of the new approach and its results are presented, along with some guidelines for future work.

2 Background

Knowledge Discovery in Databases (KDD) is the predecessor term to what is nowadays usually called *data science*, and it represents the entire process of extracting and using valuable information from raw data. The process begins with understanding the domain and defining a goal, followed by finding a dataset. The data is then cleaned and pre-processed and manipulated through *feature engineering*. After having the right variables to describe the data, a goal is then

matched to the dataset, specifying the task to perform (classification, regression, clustering...), from which results a *model*. The process is then repeated iteratively, until the model is good enough to be deployed into production, documented and further optimized.

Features are the *variables* describing the data, and in order to ensure that our models are able to achieve strong results, they are transformed, reduced and extended. *Feature selection* techniques remove redundant and irrelevant variables, avoiding an exceeding number of features, which can lead to overfitting and high variance [12]. The data can also be rearranged in some new space [18] through supervised or unsupervised methods, where the variables are combined or transformed from the original space to a new one. This is known as *feature extraction*. Variables can also be added to the original dataset, either by exploring or not domain knowledge. Unlike feature extraction, *feature generation* analyzes relations among features, augmenting the feature space [18].

Feature engineering is usually the most time-consuming step of the KDD process [27], especially since it requires human interaction and intuition to obtain the best results. This can be subjective, costly and limits the process' repeatability. To counteract this, there have been numerous works on automating the generation of features, either with or without the use of domain knowledge to improve induction. Without domain knowledge, feature generation can follow a *data-driven* approach, that only uses the input data for guidance, and works by applying some operators to features (such as logarithm, exponential or extraction of certain parts of the value, like year and month from a date), by combining features of the same data types through n-ary operators (such as sum and average), or making aggregations (for example count, max, min) [13]. *hypothesis-driven* use an induced hypothesis for ranking the new features accordingly. Decision trees have been used as the majority of situations [20]. Other methods for feature generation without knowledge domain have been used, namely: hierarchical greedy search [16], neural networks [28], reinforcement learning [15], or genetic programming [19].

Several works have been published throughout the years researching the incorporation of domain knowledge into feature generation. This knowledge does not need to be complete. As it has been proven, fragmentary knowledge can still be applied for searching new features, narrowing down the feature space [5]. These approaches go from asking for expert's domain knowledge [20], to the embedding of that knowledge into dedicated algorithms [21] or by exploring external knowledge representation formalisms. Some authors used a graph-based language for feature generation in linked data, by querying the relations inside the data. Those frameworks allow for extracting information from knowledge bases such as YAGO and DBpedia [3,11]. There are also examples of feature generation regarding textual data [10]. Other approaches use already available knowledge repositories, such as ontologies, finding candidate terms that match the dataset to increase the feature space [9,25].

The increase interest on the automation of the KDD process, usually called AutoML, led to a huge increase in AutoML frameworks in recent years, which all

have the goal of returning the best approach for a dataset with as little human intervention as possible [14]. In this context, Auto-sklearn [8] uses embeddings, clustering, matrix decomposition and one-hot encoding, as well as meta-features. Auto-Gluon [7] only uses simple data preprocessing techniques, as well as H2O [17]. TPOT [24] uses meta-features and polynomial combinations.

We can see that AutoML still has much room to improve, especially in feature generation based on domain knowledge. Indeed, *explanation* is one of the key topics to solve, and providing models easier for human interpretation is a path to explore. One way to achieve this, is through the generation of richer variables, able to describe interesting properties for the problem under analysis. [26].

3 DANKFE algorithm

The automatic exploration of domain knowledge requires one of two approaches: either embedding it in a dedicated algorithm or by using any external source. Ten years ago, domain driven data mining algorithms were proposed as the first approach, but despite their effectiveness, they required a different algorithm for each problem. The use of external sources, however, are general enough to be applied to any context. The choice of the knowledge representation formalism depends strongly on the domain knowledge available and need for expressiveness. For smaller domains, more expressive formalisms, like ontologies or higher-order logic, can be used as their performance will not become a hindrance. Larger domains demand simpler formalisms, without losing too much expressiveness.

Databases are the most usual data sources, and they are mostly designed through entity-relationship (ER) [2] or extended entity-relationship (EER) [6] diagrams and then formalized as relational schemas. ER diagrams have three main elements: rectangles to define concepts, named entities, ellipses for attributes and diamonds for relations among concepts. As an extension of ER, EER uses the same elements and allows for the representation of special relations such as inheritance and aggregation. In a certain way, we can look at those diagrams as graphical representations of simple ontologies, since they are able to express their simplest elements: concepts (entities), attributes and relations. Beside these, ontologies allow for representing inheritance through their taxonomies, which EER is also able to express. The exception is that these diagrams are not expressive enough to represent axioms. Nevertheless, their generalized use guarantees their availability for a large number of situations, but even if they do not exist in advance, they are very easy to create.

In this paper, we propose an algorithm, *DANKFE* (DomAiN Knowledge Feature Engineering), that given an EER diagram and a dataset, transforms the relations between entities into new variables to populate the dataset. The algorithm works as follows:

Given a dataset \mathcal{D} described by a set of d variables, $\mathcal{F} = \{v_1, \dots, v_d\}$ and an EER diagram, $\mathcal{KB} = (\mathcal{E}, \mathcal{R})$, where for each $v \in \mathcal{F}$ exists an $e \in \mathcal{E}$ such as e represents v , the algorithm generates a new variable v' for each relation $r \in \mathcal{R}$,

extending the set of features to F' and the given dataset to D' .

Therefore, in order to generate the new variables, the EER diagram has to be perfectly aligned with the dataset at hand. Indeed, the EER diagram defines each variable in the dataset as an entity, characterised by its *name* (used as an identifier), *type* (to help on determining the operations to perform over it), *description* (optional description of the entity) and possible *constraints*. We decided to represent every variable in the dataset as an entity in the diagram since in this manner we are able to manipulate and reason about them [22].

The diagram also has to specify the known relations between entities. To allow for the generation of new variables from them, each relation is described by its *name* (again to be used as an identifier), *inputs* (the list of entities to combine), *operations* corresponding to the sequence of operations to perform over its inputs to generate the new variable, and the *constraints* its inputs have to satisfy to make the generation possible.

Algorithm 1 DANKFE algorithm

```

procedure DANKFE( $\mathcal{D}$ ,  $\mathcal{F}$ ,  $\mathcal{KB}$ )
   $queue \leftarrow \mathcal{KB}[\text{'relations'}]$ 
  while  $queue$  is not empty do
     $current\_relation \leftarrow pop(queue)$ 
     $inputs \leftarrow current\_relation[\text{'inputs'}]$ 
     $constraint \leftarrow get\_constraint(current\_relation[\text{'constraint'}])$ 
     $operations \leftarrow reverse(current\_relation[\text{'operations'}])$ 
    if  $inputs \in \mathcal{F}$  then
       $args \leftarrow \mathcal{D}[inputs]$ 
      for  $operation$  in  $operations$  do
        for  $row$  in  $args$  do
          if  $satisfies(row, constraint)$  then
             $row \leftarrow operation(args)$ 
          else
             $row \leftarrow null$ 
          end if
        end for
      end for
    else
       $queue \leftarrow append(queue)$ 
    end if
  end while
end procedure

```

The DANKFE algorithm is illustrated in 1, and it works as follows: the relations are read from the EER model, and stored as a queue to be processed. The relations are processed one by one, if the inputs are already available. If part of the input is not yet available (is also generated and in the queue) that relation

is sent to the end of the queue. If the inputs are already available, the list of operations specified in the diagram for that relationship, is applied to any row in the dataset that satisfies the constraints imposed for the relation. Whenever any row does not meet the constraints, a null value is imputed. When all rows are processed, the relation is removed from the queue.

Since the operation used to generate the new variable is specified in the diagram, we could think this can be any operation imaginable. Unfortunately, that is not the case, since in order to do that we would need an interpreter to decode the operations to apply. Moreover, that would introduce additional processing time and complexity, which we shall avoid.

Nevertheless, whenever the operation only encompasses a single record (just one row in the dataset), we are able to directly implement it through a lambda function in Python. In this case, the function applies each operation in the list of operations in reverse (similarly to a composition of operations) sequentially over the corresponding inputs - the values assumed by each input variable in the record, and return the output value - the value to assign to the new variable for the given record. The input columns are fed to the algorithm for each relation, processing the defined operation row by row.

Let \mathcal{D} be a dataset, $\mathcal{F}' = \{v_1, \dots, v_d\}$ the set of d variable describing \mathcal{D} and $\mathcal{KB} = (\mathcal{E}, \mathcal{R})$ be a EER diagram, as defined before. Let also $r = (\Xi, \Gamma, \Psi)$ be any relation in \mathcal{R} , with Ξ the set of input variables, Γ the set of operations to perform and Ψ the set of constraints to be satisfied. For each $r \in \mathcal{R}$ a new variable v_r is generated and added to \mathcal{F}' . Then, for each record $x = x_1 \dots x_d$ in \mathcal{D} , v_r assumes a value as follows:

1. *if x does not satisfy some constraint $\psi \in \Psi$, a null value is assigned;*
2. *otherwise, the first operation $\gamma_1 \in \Gamma$ is applied over the values assumed by x over all variables in Ξ ; then its result is submitted to the following operations in Γ in the given order, and so on.*

Among the operations specified by relations in the EER diagram, we may consider the following ones:

- *Unary operations:* any operation with a single argument of any type, such as square root or absolute value for numeric variables, or extraction functions (year/month for dates, surname for extracting the last name, etc.);
- *Binary operations:* any operation with two arguments from the same type, such as difference, division, etc.;
- *Preparation operations:* any operation usual in the data preparation step, such as dummification, scaling, and missing value imputation.

The EER diagram was internally represented as a JSON file, since the algorithm was implemented in Python. Any other similar language such as XML, RDF or even OWL could be used.

3.1 Illustration

To better understand the proposed algorithm, consider the knowledge base represented by the EER diagram represented in Fig. 1.

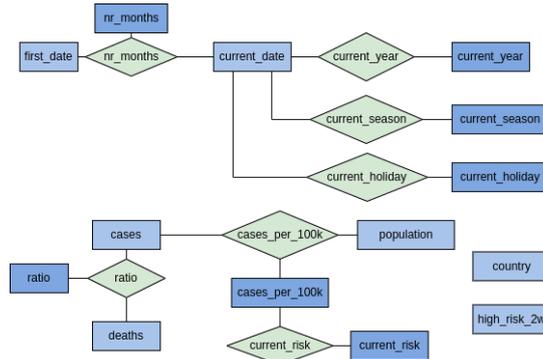


Fig. 1. Example of the EER diagram for feature generation.

Additionally, consider the data \mathcal{D} on Table 1, which is described by the set of variables $\mathcal{F} = \{current_date, cases, deaths, country, population, first_date, high_risk_2w\}$.

current_date	cases	deaths	country	population	first_date	high_risk_2w
2021/02/23	1032	63	PT	10295909	2020/03/03	TRUE
2022/02/14	20360	78	UK	10718565	2020/02/23	TRUE
2021/08/12	223	2	PL	37958138	2020/03/07	FALSE
2020/06/11	22	0	AT	8901064	2020/02/26	FALSE

Table 1. Illustration dataset, labeled by *high_risk_2w*

We can see that all variables in \mathcal{F} are present in the EER diagram as entities (represented as light blue rectangles). Beside them, we found six relationships (green diamonds) and seven additional entities (dark blue rectangles), corresponding to the variables to generate. Note, that each relationship is linked to a set of entities, where the lighter ones correspond to their inputs, and the darker ones to the new variables to generate.

Fig. 2 shows the definition of two relations *ratio* and *nr_months*. Here are specified the inputs for each relations and the list of operations for each one of them. The *ratio* relation divides *cases* by *deaths* whenever the second one is different from zero, imposed by its constraint. The relation *nr_months* applies the composition of a division by thirty (the average number of days per month), after computing the difference between and *current_date* and *first_date*.

<pre>{ "name": "ratio", "type": "float", "operations": ["/"], "inputs": ["cases", "deaths"], "constraint": "deaths != 0" },</pre>	<pre>{ "name": "nr_months", "type": "int", "operations": ["divide_by_30", "-"], "inputs": ["current_date", "first_date"], "constraint": "" },</pre>
---	---

Fig. 2. Specification of *ratio* (left) and *nr_months* (right) relations.

current_date	current_year	current_season	current_holiday	nr_months	ratio	cases_per_100k	current_risk
2021/02/23	2021	winter	FALSE	11	16.4	10.023	FALSE
2022/02/14	2022	winter	FALSE	24	261.0	189.951	TRUE
2021/08/12	2021	summer	FALSE	17	111.5	0.587	FALSE
2020/06/11	2020	spring	TRUE	3	null	0.247	FALSE

Table 2. Generated variables, indexed by *current.date*.

Table 2 summarizes the variables generated by our algorithm when applied to the data in \mathcal{D} , shown in the previous table, and using the EER diagram in Fig. 1. Beside *ratio* and *nr_months* explained above, *current_year* is computed by extracting the year over the *current_date* variable and similarly for *current_season*, but using a given function to extract the yearly seasons. The *current_holiday* variable is computed by consulting a calendar for each country. Moreover, *cases_per_100k* is just the number of cases by one hundred thousand population, and *current_risk* is derived from the previous one, just verifying if it is above 120 cases per one hundred thousand population.

4 Validation

In order to validate our proposal, we compared the quality of classification models trained over a given dataset and models trained over an extension of it, described both by the original and generated variables, using the algorithm proposed. Beside the performance, we studied the impact on the time spent processing the train (*fit*) and prediction (*pred*) of those models. Additionally, we compared those results with the resulting from training a model for the same problem through AutoSklearn [8], which uses a variety of model selection and hyperparameter optimization methods to return an ensemble of models, as well as some data preparation (meta-features, embeddings, one-hot encoding).

The dataset was collected from the European Centre for Disease Prevention and Control ¹. The dataset was labeled by a new variable, *high_risk_2w*, using the level of risk recorded fourteen days later (implying the loss of the first fourteen days registered). In this manner, the set of variables describing the dataset was

¹ <https://www.ecdc.europa.eu/en/publications-data/data-daily-new-cases-covid-19-eueea-country>, from which we ignored the redundant variables (*year*, *day*, *month*, *countriesAndTerritories*, *countryterritoryCode* and *continentExp*).

just $\mathcal{F}=\{dateRep, cases, deaths, geoID \text{ (country)}, popData2020 \text{ (population)}, high_risk_2w\}$.

We can state the analysis goal as to find the best model able to predict the level of risk at each country in two weeks. So, given the current date, its corresponding number of cases and deaths, we want to know which of the countries will be in high risk in fourteen days.



Fig. 3. Class distribution (left) and correlation among variables (right).

The original dataset is composed of 22211 records, one per country per day, containing the number of registered number of covid-19 cases and deaths. It is important to note the class is balanced, with 41% of the records representing a low risk day in 2 weeks, and 59% representing high risk, as seen in Fig. 3 (left).

From this dataset and the EER diagram depicted in Fig. 1, ten variables were generated (*current_year*, *current_month*, *current_day*, *current_season*, *current_holiday*, *current_weekday*, *nr_months*, *ratio*, *cases_per_100k* and *current_risk*), where the variables have the semantic described before. Fig. 3 (right) shows the correlation between the original and generated variables. We can see that the most correlated pair is *month* and *season*.

Results in Fig. 4 (left) show that all the techniques improve their performance over the extended dataset (original+generated), with the largest improvement reached by the ensembles tested (random forests and gradient boosting).

Additionally, we can see that the time spent also increases, as seen in Fig. 4 (right). For almost all techniques, the increase was equal to the amount of time spent on generating the variables (around 28.5 seconds), with the difference of time on dealing with the extra variables being neglectful. Only gradient boosting doubles the time spent by exploring the new variables. Another important remark is that the extended dataset shows as a good performance as the model trained by AutoSklearn, while also completing faster, since the time spent by the latter is constant and equal to one hour (3600 seconds).

The 28.5 seconds wasted on generating the variables, however, are not equally distributed. Fig. 5 shows that different variables required different amount of

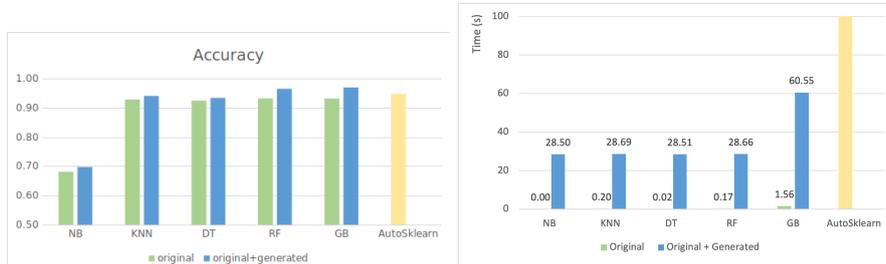


Fig. 4. Comparison of accuracy (left) and processing times (right) for different models.

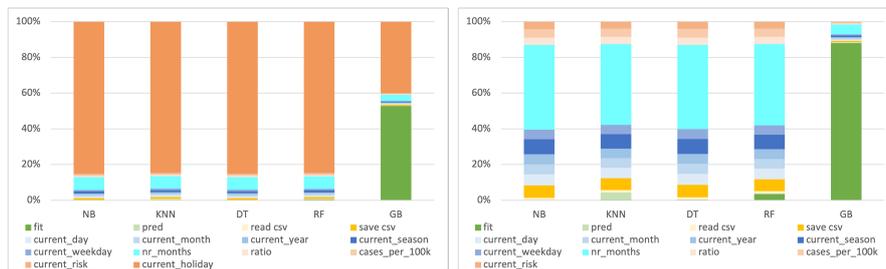


Fig. 5. Time comparison per variable generated against other procedures required, with (left) and without (right) holiday variable.

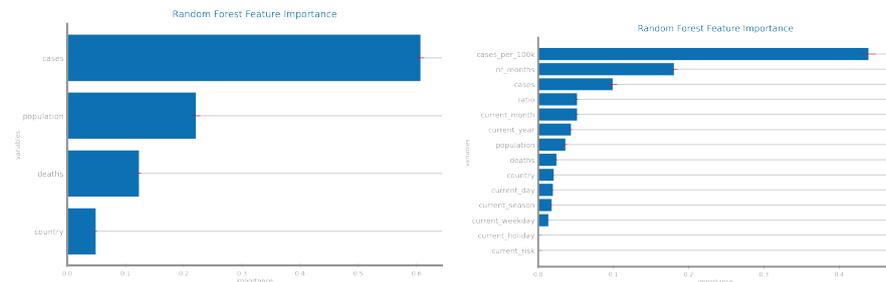


Fig. 6. Average feature importance for Random Forests on original (left) and original+generated variables (right).

time to be computed. It is clear that the *current.holiday* variable waste almost all the time required (5-left), spending more than 24 seconds. This is due to the need to access to Python calendars, in order to check if all given days are holidays in each country. Analyzing the remaining variables, we can see that only *nr_months* (5-right) spent more than 1 second, and all the others require as much as the time needed to save the dataset. Note that *nr_months* implements a composite function, requiring to apply two operations instead of one, which justifies the extra time. Results also show, that the time needed to generate all variables is less then the time required for training the gradient boosting models.

Another important result, is that models trained over the extended dataset favor the generated variables. Fig. 6 shows features' importance according to Random Forests, for both original and extended dataset, but similar results were obtained for the remaining models. In particular, the *cases_per_100k* variable replaces both variables *cases* and *population*, according to the domain practices. Beside it, the new models also explore the *nr_months* and *ratio*.

5 Conclusion

The results show that the use of domain knowledge to generate new features via the EER diagram has improved the performance of all models. Not only this generation of features helped increase the robustness of models (as seen by the performance increase) but also doing so without a significant time increase. Additionally, when compared to an AutoML framework, the use of domain knowledge to increase the feature space achieved better classifications results while also taking significantly less time.

As for the majority of related work, the features proposed were derived from a single record, which does not allow for making single aggregations, like the ones performed by group by clauses in SQL (average, min and max, for example). The next steps to follow are in two directions. First, to enlarge the operations to allow for aggregation operations; and second, to automatically propose some operations, instead of requiring the entire specification in the ER diagram. Preliminary results show the first line of work is accomplished without major problems and without hindering the algorithm performance.

6 Acknowledgements

This work was supported by national funds by Fundação para a Ciência e Tecnologia (FCT) through project VizBig (PTDC/CCI-CIF/28939/2017).

References

1. Antunes, C., Silva, A.: New trends in knowledge driven data mining. In: ICEIS (1). pp. 346–351 (2014)
2. Chen, P.P.S.: The entity-relationship model—toward a unified view of data. *ACM transactions on database systems (TODS)* **1**(1), 9–36 (1976)
3. Cheng, W., Kasneci, G., Graepel, T., Stern, D., Herbrich, R.: Automated feature generation from structured knowledge. In: Proc. 20th ACM international conference on Information and knowledge management. pp. 1395–1404 (2011)
4. Domingos, P.: A few useful things to know about machine learning. *Communications of the ACM* **55**(10), 78–87 (2012)
5. Donoho, S., Rendell, L.: Feature construction using fragmentary knowledge. In: Feature Extraction, Construction and Selection, pp. 273–288. Springer (1998)
6. Elmasri, R., Navathe, S.B.: The Enhanced Entity–Relationship (EER) Model, p. 107–135. Addison-Wesley (2000)

7. Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., Smola, A.: Autogluon-tabular: Robust and accurate automl for structured data. arXiv preprint arXiv:2003.06505 (2020)
8. Feurer, M., Eggenesperger, K., Falkner, S., Lindauer, M., Hutter, F.: Auto-sklearn 2.0: Hands-free automl via meta-learning. arXiv preprint arXiv:2007.04074 (2020)
9. Friedman, L., Markovitch, S.: Recursive feature generation for knowledge-based learning. arXiv preprint arXiv:1802.00050 (2018)
10. Gabrilovich, E., Markovitch, S.: Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research* **34**, 443–498 (2009)
11. Galhotra, S., Khurana, U., Hassanzadeh, O., Srinivas, K., Samulowitz, H., Qi, M.: Automated feature enhancement for predictive modeling using external knowledge. In: 2019 International Conference on Data Mining Workshops (ICDMW). pp. 1094–1097. IEEE (2019)
12. Hastie, T., Friedman, J., Tibshirani, R.: *High-Dimensional Problems*, p. 649–650. Springer, 2 edn. (2017)
13. Hu, Y.J., Kibler, D.: Generation of attributes for learning algorithms. In: AAAI/IAAI, Vol. 1. pp. 806–811 (1996)
14. Hutter, F., Kotthoff, L., Vanschoren, J. (eds.): *Automated Machine Learning: Methods, Systems, Challenges*. The Springer Series on Challenges in Machine Learning, Springer International Publishing, Cham (2019)
15. Khurana, U., Samulowitz, H.: Autonomous predictive modeling via reinforcement learning. In: Proc. 29th ACM International Conference on Information & Knowledge Management. pp. 3285–3288 (2020)
16. Khurana, U., Turaga, D., Samulowitz, H., Parthasarathy, S.: Cognito: Automated feature engineering for supervised learning. In: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW). pp. 1304–1307. IEEE (2016)
17. LeDell, E., Poirier, S.: H2o automl: Scalable automatic machine learning. In: Proc. AutoML Workshop at ICML. vol. 2020 (2020)
18. Liu, H., Motoda, H.: *Feature extraction, construction and selection: A data mining perspective*, vol. 453. Springer Science & Business Media (1998)
19. Ma, J., Gao, X.: A filter-based feature construction and feature selection approach for classification using genetic programming. *Knowledge-Based Systems* **196**, 105806 (2020)
20. Markovitch, S., Rosenstein, D.: Feature generation using general constructor functions. *Machine Learning* **49**, 59–98 (2004)
21. Matheus, C.J.: The need for constructive induction. In: *Machine Learning Proc. 1991*, pp. 173–177. Elsevier (1991)
22. McCarthy, J.: Generality in artificial intelligence. *Communications of the ACM* **30**(12), 1030–1035 (1987)
23. Miller, S., Hughes, D.: *The quant crunch: How the demand for data science skills is disrupting the job market*. Burning Glass Technologies (2017)
24. Olson, R.S., Moore, J.H.: Tpot: A tree-based pipeline optimization tool for automating machine learning. In: *Workshop on automatic machine learning*. pp. 66–74. PMLR (2016)
25. Salguero, A.G., Medina, J., Delatorre, P., Espinilla, M.: Methodology for improving classification accuracy using ontologies: application in the recognition of activities of daily living. *Journal of Ambient Intelligence and Humanized Computing* **10**(6), 2125–2142 (2019)

26. Wang, D., Weisz, J.D., Muller, M., Ram, P., Geyer, W., Dugan, C., Tausczik, Y., Samulowitz, H., Gray, A.: Human-ai collaboration in data science: Exploring data scientists' perceptions of automated ai. *Proc. ACM on Human-Computer Interaction* **3**(CSCW), 1–24 (2019)
27. Waring, J., Lindvall, C., Umeton, R.: Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artificial Intelligence in Medicine* **104**, 101822 (2020)
28. Xie, Y., Wang, Z., Li, Y., Ding, B., Gürel, N.M., Zhang, C., Huang, M., Lin, W., Zhou, J.: Fives: Feature interaction via edge search for large-scale tabular data. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. pp. 3795–3805 (2021)

Parametric Extension of q-Divergence-based Fuzzy Clustering for Spherical Data

Yuchi Kanzawa

Shibaura Institute of Technology, Japan,
kanzawa@shibaura-it.ac.jp

Abstract. In this study, a fuzzy clustering algorithm is proposed for spherical data, developed as an extension of a conventional algorithm, q-divergence-based fuzzy clustering for spherical data (QFCS), by increasing the number of fuzzification parameters from two to three. Through numerical experiments, the effects of the parameters of the proposed algorithm are clarified. In addition, two real datasets are used, to demonstrate that the proposed algorithm outperforms the conventional algorithm in terms of clustering accuracy.

Keywords: Fuzzy Clustering, Spherical Data, Parametric Extension

1 Introduction

Some clustering tasks, such as document clustering, can be reduced to the clustering of spherical data, where all of the objects are of unit length. In this situation, the cosine correlation is used to measure an object and a cluster prototype. The algorithm typically used for clustering spherical data is the spherical k-means algorithm [1], which is a hard clustering framework, in which each object belongs to a single cluster. Other algorithms, namely the Kullback-Leibler (KL) divergence-based fuzzy clustering for spherical data (KLFCS) [2], and Bezdek-type fuzzy clustering for spherical data (BFCS) [2], employ a fuzzy clustering framework, in which each object belongs to two or more clusters; the degree to which an object belongs to a given cluster is termed its membership in that cluster. Higashi et al. proposed the q-divergence-based fuzzy clustering for spherical data (QFCS) [3], and showed that QFCS outperforms the KLFCS and BFCS methods in terms of clustering accuracy. QFCS is a two-parameter model (that is, it has two fuzzification parameters), whereas the BFCS and KLFCS objective functions are one-parameter models. With the value of one parameter specified, QFCS reduces to KLFCS, and with the value of the other parameter specified, QFCS reduces to BFCS; thus, QFCS is a two-parameter extension of both KLFCS and BFCS. By controlling these two fuzzification parameters in QFCS, flexible clustering results are obtained, resulting in higher clustering accuracy. Extending QFCS further by increasing the number of fuzzification parameters has the potential to produce clustering results that are still more flexible, resulting in a further improvement in clustering accuracy.

In this study, we propose a fuzzy clustering algorithm for spherical data, developed as an extension of QFCS by increasing the number of fuzzification parameters. In the proposed method, which we call the revised QFCS (RQFCS) method, we distinguish the fuzzification parameter in the second term of the QFCS objective function from that in the first term. The proposed model is thus a three-parameter extension of QFCS and therefore, has the potential to produce clustering results that are more flexible than those produced by QFCS. Decoupling the fuzzification parameter in fuzzy clustering has been discussed in [4], where two different values are used for fuzzification parameter; one for computation of cluster centers, the other for computation of the final memberships, whereas in the proposed method the fuzzification parameter is decoupled at the optimization problem. Then, the proposed method uses both the two different fuzzification parameter values through all the steps. We conduct numerical experiments using an artificial dataset to clarify the effects of the fuzzification parameters in the proposed method. In addition, we conduct numerical experiments using two real datasets to compare the proposed algorithm with QFCS in terms of clustering accuracy.

The remainder of this paper is organized as follows. In Section 2, we briefly describe BFCS, KLFCS, and QFCS, which are the conventional algorithms for fuzzy clustering of spherical data. In Section 3, the proposed algorithm is described. In Section 4, we present the results of the numerical experiments conducted to clarify the effects of the fuzzification parameters in the proposed method, and to evaluate the performance of the proposed algorithm relative to QFCS in terms of clustering accuracy. In Section 5, we conclude the work.

2 Preliminaries

Let $X = \{x_k \in \mathbb{R}^M \mid \|x_k\|_2 = 1, k \in \{1, \dots, N\}\}$ be a set of $(M-1)$ -dimensional spherical data. We consider a partitioning of X into C clusters. We introduce a representative point for each cluster, called the cluster center. The set of such cluster centers is denoted by $V = \{v_i \in \mathbb{R}^M \mid \|v_i\|_2 = 1, i \in \{1, \dots, C\}\}$. Let $U = u_{i,k}$ ($i \in \{1, \dots, C\}, k \in \{1, \dots, N\}$) be the membership of x_k in the i -th cluster. The membership U has the constraint

$$\sum_{i=1}^C u_{i,k} = 1, \quad u_{i,k} \in [0, 1]. \quad (1)$$

Further, we introduce the cluster size controller, denoted by $A = \{\alpha_i \in (0, 1)\}_{i=1}^C$. The cluster size controller A has the constraint

$$\sum_{i=1}^C \alpha_i = 1. \quad (2)$$

The BFCS, KLFCS, and QFCS algorithms are obtained by solving the respective optimization problems

$$\underset{U,V,A}{\text{minimize}} \sum_{i=1}^C \sum_{k=1}^N (u_{i,k})^m (1 - x_k^\top v_i), \quad (3)$$

$$\underset{U,V,A}{\text{minimize}} \sum_{i=1}^C \sum_{k=1}^N u_{i,k} (1 - x_k^\top v_i) + \lambda^{-1} \sum_{i=1}^C \sum_{k=1}^N u_{i,k} \ln \left(\frac{u_{i,k}}{\alpha_i} \right), \quad (4)$$

and

$$\underset{U,V,A}{\text{minimize}} \sum_{i=1}^C \sum_{k=1}^N (\alpha_i)^{1-m} (u_{i,k})^m (1 - x_k^\top v_i) + \frac{\lambda^{-1}}{m-1} \sum_{i=1}^C \sum_{k=1}^N (\alpha_i)^{1-m} (u_{i,k})^m, \quad (5)$$

which are based on Eq. (1), Eq. (2), and

$$\|v_i\|_2 = 1 \text{ for } i \in \{1, \dots, C\}. \quad (6)$$

and in which $\lambda > 0$ and $m > 1$ are the fuzzification parameters. As it has previously been shown [3] that the QFCS algorithm outperforms KLFCS and BFCS in terms of clustering accuracy, this study considers only the QFCS algorithm for the comparison with the proposed algorithm; the KLFCS and BFCS optimization problems are compared with the proposed one in the later section. The QFCS algorithm is summarized as follows:

Algorithm 1 (QFCS)

STEP 1. Given the number of clusters C and the fuzzification parameters (m, λ) , where $m > 1$ and $\lambda > 0$, let the set of initial membership be u .

STEP 2. Obtain v using

$$v_i = \frac{\sum_{k=1}^N (u_{i,k})^m x_k}{\|\sum_{k=1}^N (u_{i,k})^m x_k\|_2} \quad (7)$$

for $i \in \{1, \dots, C\}$.

STEP 3. Obtain $\{d_{i,k}\}_{i,k=(1,1)}^{(C,N)}$ using

$$d_{i,k} = 1 - x_k^\top v_i \quad (8)$$

for $i \in \{1, \dots, C\}$, $k \in \{1, \dots, N\}$.

STEP 4. Obtain α using

$$\alpha_i = \frac{\left(\sum_{k=1}^N (u_{i,k})^m (1 - \lambda(1 - m)d_{i,k})\right)^{1/m}}{\sum_{i'=1}^C \left(\sum_{k=1}^N (u_{i',k})^m (1 - \lambda(1 - m)d_{i',k})\right)^{1/m}} \quad (9)$$

for $i \in \{1, \dots, C\}$.

STEP 5. Obtain u using

$$u_{i,k} = \frac{\alpha_i (1 - \lambda(1 - m)d_{i,k})^{1/(1-m)}}{\sum_{i'=1}^C \alpha_{i'} (1 - \lambda(1 - m)d_{i',k})^{1/(1-m)}} \quad (10)$$

for $i \in \{1, \dots, C\}$, $k \in \{1, \dots, N\}$.

STEP 6. Check the stopping criterion for (u, v, α) . If the criterion is not satisfied, go to STEP 2.

3 Proposed Method

As mentioned in the previous section, the QFCS objective function is a two-parameter model, having the two fuzzification parameters m and λ . (By contrast, the BFCS and KLFCS objective functions are one-parameter models.) As QFCS with $m \searrow 1$ reduces to KLFCS, and as QFCS with $\lambda \rightarrow +\infty$ reduces to BFCS, QFCS is a two-parameter extension of both KLFCS and BFCS. Controlling these two fuzzification parameters of QFCS produces flexible clustering results, resulting in higher clustering accuracy.

Extending QFCS further by increasing the number of fuzzification parameters has the potential to produce clustering results that are still more flexible, resulting in a further improvement in clustering accuracy. In the proposed method, which we call the revised QFCS (RQFCS) method, we distinguish the fuzzification parameter m in the first term of the QFCS objective function from that in the second term, denoting them as m_1 and m_2 , respectively, as follows:

$$\begin{aligned} & \underset{U, V, A}{\text{minimize}} \sum_{i=1}^C \sum_{k=1}^N (\alpha_i)^{1-m_1} (u_{i,k})^{m_1} (1 - x_k^\top v_i) \\ & + \frac{\lambda^{-1}}{m_2 - 1} \sum_{i=1}^C \sum_{k=1}^N (\alpha_i)^{1-m_2} (u_{i,k})^{m_2}, \end{aligned} \quad (11)$$

subject to Eqs. (1), (2), and (6), where $m_1 > 1$, $m_2 > 1$, and $\lambda > 0$ are the fuzzification parameters. The proposed model is thus a three-parameter extension of QFCS. It reduces to the KLFCS method with $m_1 = m_2 \searrow 1$, to the BFCS method with $\lambda \rightarrow +\infty$ or $m_2 \searrow 1$, and to the QFCS method with $m_1 = m_2$. Thus, the proposed method, RQFCS, is a three-parameter extension of QFCS, KLFCS, and BFCS.

The Lagrange function of RQFCS, $L(U, V, A)$, is defined as

$$\begin{aligned}
L(U, V, A) &= \sum_{i=1}^C \sum_{k=1}^N (\alpha_i)^{1-m_1} (u_{i,k})^{m_1} (1 - x_k^\top v_i) \\
&\quad + \frac{\lambda^{-1}}{m_2 - 1} \sum_{i=1}^C \sum_{k=1}^N (\alpha_i)^{1-m_2} (u_{i,k})^{m_2} \\
&\quad + \sum_{i=1}^C \eta_i (1 - \|v_i\|_2^2) \\
&\quad + \theta \left(1 - \sum_{i=1}^C \alpha_i \right) \\
&\quad + \sum_{k=1}^N \zeta_k \left(1 - \sum_{i=1}^C u_{i,k} \right) \tag{12}
\end{aligned}$$

with Lagrange multipliers $(\zeta_1, \dots, \zeta_N, \theta, \eta_1, \dots, \eta_C)$. The conditions for optimality are described as

$$\frac{\partial L(U, V, A)}{\partial u_{i,k}} = 0 \quad (i \in \{1, \dots, C\}, k \in \{1, \dots, N\}), \tag{13}$$

$$\frac{\partial L(U, V, A)}{\partial v_i} = 0 \quad (i \in \{1, \dots, C\}), \tag{14}$$

$$\frac{\partial L(U, V, A)}{\partial \alpha_i} = 0 \quad (i \in \{1, \dots, C\}), \tag{15}$$

$$\frac{\partial L(U, V, A)}{\partial \zeta_k} = 0 \quad (k \in \{1, \dots, N\}), \tag{16}$$

$$\frac{\partial L(U, V, A)}{\partial \theta} = 0, \tag{17}$$

$$\frac{\partial L(U, V, A)}{\partial \eta_i} = 0 \quad (i \in \{1, \dots, C\}). \tag{18}$$

The set V of optimal cluster centers is obtained from Eqs. (14) and (18), and is described as

$$v_i = \frac{\sum_{k=1}^N (u_{i,k})^{m_1} x_k}{\left\| \sum_{k=1}^N (u_{i,k})^{m_1} x_k \right\|_2} \tag{19}$$

for $i \in \{1, \dots, C\}$. This formula is the same as those for BFCS and QFCS because the optimization problems for RQFCS, BFCS, and QFCS have the same form with respect to V .

The optimal membership condition, given in Eqs. (13) and (16), is equivalently written as

$$m_1 (\alpha_i)^{1-m_1} d_{i,k} (u_{i,k})^{m_1-1} + \frac{\lambda^{-1}}{m_2 - 1} m_2 (\alpha_i)^{1-m_2} (u_{i,k})^{m_2-1} = \zeta_k \tag{20}$$

for $i \in \{1, \dots, C\}$ and $k \in \{1, \dots, N\}$, along with Eq. (1). We fix $k \in \{1, \dots, N\}$ and consider obtaining the values $u_{1,k}, \dots, u_{C,k}, \zeta_k$ satisfying Eqs. (20) and (1). In general, this nonlinear equation is difficult to solve analytically; hence, we solve it numerically. Here, we define

$$f_i(\mu) = m_1(\alpha_i)^{1-m_1} d_{i,k}(\mu)^{m_1-1} + \frac{\lambda^{-1}}{m_2-1} m_2(\alpha_i)^{1-m_2} (\mu)^{m_2-1} \quad (21)$$

for $i \in \{1, \dots, C\}$. Since this function f is strictly increasing and since $u_{i,k}$ is bounded as $u_{i,k} \in [0, 1]$, we can obtain the optimal membership by the bisection method with a given ζ_k value as follows:

Algorithm 2

STEP 1. Set $(u_{i,k}^-, u_{i,k}^+) = (0, 1)$.

STEP 2. Set $\bar{u}_{i,k} = (u_{i,k}^- + u_{i,k}^+)/2$. If $|u_{i,k}^+ - u_{i,k}^-|$ is sufficiently small, then terminate the algorithm with the optimal $u_{i,k}$ being $\bar{u}_{i,k}$.

STEP 3. If $f_i(\bar{u}_{i,k}) < \zeta_k$, let $u_{i,k}^- = \bar{u}_{i,k}$; otherwise, let $u_{i,k}^+ = \bar{u}_{i,k}$. Go to STEP 2.

The optimal value for ζ_k can be obtained using the bisection method, as follows. First, we can determine the upper and lower bounds of ζ_k as

$$\begin{aligned} \zeta_k &= m_1(\alpha_i)^{1-m_1} d_{i,k}(u_{i,k})^{m_1-1} + \frac{\lambda^{-1}}{m_2-1} m_2(\alpha_i)^{1-m_2} (u_{i,k})^{m_2-1} \\ &\leq m_1(\alpha^-)^{1-m_1} d_k^+ + \frac{\lambda^{-1}}{m_2-1} m_2(\alpha^-)^{1-m_2}, \end{aligned} \quad (22)$$

$$\begin{aligned} \zeta_k &= m_1(\alpha_i)^{1-m_1} d_{i,k}(u_{i,k})^{m_1-1} + \frac{\lambda^{-1}}{m_2-1} m_2(\alpha_i)^{1-m_2} (u_{i,k})^{m_2-1} \\ &\geq 0, \end{aligned} \quad (23)$$

where $\alpha^- = \min_{1 \leq j \leq C} \{\alpha_j\}$, and $d_k^+ = \max_{1 \leq j \leq C} \{d_{j,k}\}$. Thus, the algorithm for obtaining the optimal ζ_k value can be consolidated as follows.

Algorithm 3

STEP 1. Set $(\zeta_k^-, \zeta_k^+) = (0, m_1(\alpha^-)^{1-m_1} d_k^+ + \frac{\lambda^{-1}}{m_2-1} m_2(\alpha^-)^{1-m_2})$.

STEP 2. Let $\bar{\zeta}_k$ be $(\zeta_k^- + \zeta_k^+)/2$. If $|\zeta_k^+ - \zeta_k^-|$ is sufficiently small, then terminate the algorithm with the optimal ζ_k being $\bar{\zeta}_k$.

STEP 3. Obtain $u_{i,k}$ using Algorithm 2.

STEP 4. If $\sum_{i=1}^C u_{i,k} < 1$, let $\zeta_k^- = \bar{\zeta}_k$; otherwise, let $\zeta_k^+ = \bar{\zeta}_k$. Go to STEP 2.

The optimal conditions for the cluster size controller, given in Eqs. (15) and (17), are equivalently written as

$$(m_1 - 1) \left\{ \sum_{k=1}^N (u_{i,k})^{m_1} d_{i,k} \right\} (\alpha_i)^{-m_1} + \lambda^{-1} \left\{ \sum_{k=1}^N (u_{i,k})^{m_2} \right\} (\alpha_i)^{-m_2} = \theta \quad (24)$$

for $i \in \{1, \dots, C\}$, along with Eq. (2). In general, this nonlinear equation is difficult to solve analytically; hence, we solve it numerically. Here, we define

$$g_i(a) = (m_1 - 1) \left\{ \sum_{k=1}^N (u_{i,k})^{m_1} d_{i,k} \right\} (a)^{-m_1} + \lambda^{-1} \left\{ \sum_{k=1}^N (u_{i,k})^{m_2} \right\} (a)^{-m_2} \quad (25)$$

for $i \in \{1, \dots, C\}$. Since this function g is strictly decreasing and since α_i is bounded as $\alpha_i \in (0, 1)$, we can obtain the optimal cluster size controller by the bisection method with a given θ value as follows:

Algorithm 4

- STEP 1. Set $(\alpha_i^-, \alpha_i^+) = (0, 1)$.
- STEP 2. Set $\bar{\alpha}_i = (\alpha_i^- + \alpha_i^+)/2$. If $|\alpha_i^+ - \alpha_i^-|$ is sufficiently small, then terminate the algorithm with the optimal α_i being $\bar{\alpha}_i$.
- STEP 3. If $g_i(\bar{\alpha}_i) < \theta$, let $\alpha_i^- = \bar{\alpha}_i$; otherwise, let $\alpha_i^+ = \bar{\alpha}_i$. Go to STEP 2.

The optimal value for θ can be obtained using the bisection method, as follows. First, we can determine the lower bound of θ using $u_{i,k} \geq 0$, $m_1 > 1$, $m_2 > 1$, and $\alpha_i \leq 1$, as well as the decreasing $(\alpha_i)^{-m_1}$ and $(\alpha_i)^{-m_2}$, as

$$\begin{aligned} \theta &= (m_1 - 1) \left\{ \sum_{k=1}^N (u_{i,k})^{m_1} d_{i,k} \right\} (\alpha_i)^{-m_1} + \lambda^{-1} \left\{ \sum_{k=1}^N (u_{i,k})^{m_2} \right\} (\alpha_i)^{-m_2} \\ &\geq (m_1 - 1) \hat{u}^- (\alpha_i)^{-m_1} + \lambda^{-1} \tilde{u}^- (\alpha_i)^{-m_2} \\ &\geq (m_1 - 1) \hat{u}^- + \lambda^{-1} \tilde{u}^-, \end{aligned} \quad (26)$$

where $\hat{u}^- = \min_{1 \leq j \leq C} \{ \sum_{k=1}^N (u_{j,k})^{m_1} d_{j,k} \}$ and $\tilde{u}^- = \min_{1 \leq j \leq C} \{ \sum_{k=1}^N (u_{j,k})^{m_2} \}$. The upper bound of θ , however, cannot be obtained analytically but can be obtained using the following algorithm.

Algorithm 5

- STEP 1. Set $\theta^- = (m_1 - 1) \hat{u}^- + \lambda^{-1} \tilde{u}^-$. Set the candidate for the upper bound of θ , $\theta^+ > \theta^-$.
- STEP 2. Obtain $\{\bar{\alpha}_i\}_{i=1}^C$ from Algorithm 4 with $\theta = \theta^+$. If $\sum_{i=1}^C \bar{\alpha}_i < 1$, then terminate this algorithm with the upper bound of θ being θ^+ . Otherwise, set $\theta^+ \leftarrow \kappa \theta^+$ with $\kappa > 1$, and return to the beginning of STEP 2.

Using the upper bound of θ obtained from Algorithm 5 and the lower bound of θ given by Eq. (26), the algorithm for obtaining the optimal θ value is consolidated as follows.

Algorithm 6

- STEP 1. Let the upper bound of θ , θ^+ , be that obtained from Algorithm 5. Let the lower bound of θ , θ^- , be that given by Eq. (26).

- STEP 2. Set $\bar{\theta} = (\theta^- + \theta^+)/2$. If $|\theta^+ - \theta^-|$ is sufficiently small, then terminate the algorithm with the optimal θ being $\bar{\theta}$.
- STEP 3. Calculate $\{\alpha_i\}_{i=1}^C$ using Algorithm 4.
- STEP 4. If $\sum_{i=1}^C \alpha_i < 1$, let $\theta^+ = \bar{\theta}$; otherwise, let $\theta^- = \bar{\theta}$. Go to STEP 2.

Based on the above discussion, we propose the following algorithm, as the RQFCS method:

Algorithm 7 (RQFCS)

- STEP 1. Fix the fuzzification parameter (m_1, m_2, λ) and the number of clusters C . Initialize the membership U .
- STEP 2. Obtain V using Eq. (19).
- STEP 3. Obtain θ using Algorithms 5 and 6, along with the cluster size controller A , using Algorithm 4.
- STEP 4. Obtain ζ_k using Algorithm 3, along with the membership U , using Algorithm 2.
- STEP 5. If the criterion for (U, V, A) is satisfied, then terminate the algorithm. Otherwise, go to STEP 2.

4 Numerical Experiment

In this section, we present some numerical examples to observe the fuzzification effect of the proposed method using an artificial dataset, and to compare the clustering accuracy of the proposed method with the QFCS method using two real datasets.

In the first experiment, we use an artificial dataset consisting of 150 points on the 2D unit sphere, each of which belongs to one of three clusters (50 points per cluster), as shown in Fig. 1. We observe that the proposed method produces appropriate clustering results with all combinations of the fuzzification parameter values. The fuzzy classification functions (FCFs) for the cluster #1 are shown in Figs. 2–5 with $(m_1, m_2, \lambda) = (1.05, 1.1, 10)$, $(m_1, m_2, \lambda) = (10.0, 1.1, 10)$, $(m_1, m_2, \lambda) = (1.5, 1.001, 10)$, and $(m_1, m_2, \lambda) = (1.5, 2.3, 10)$, respectively. Figs. 2 and 3 show that the larger the fuzzification parameter value m_1 , the fuzzier is the FCF. Figs. 4 and 5 show that the larger the fuzzification parameter value m_2 , the fuzzier is the FCF. We note that the fuzzification effects of m_1 and of m_2 differ as follows. We observe that in Fig. 3 the FCF as a whole is blue except the region neighboring v_1 , which is red, and the regions v_2 and v_3 , which are black. This implies that the effect of m_1 is to fuzzify the FCF as a whole except in the regions neighboring cluster centers, where the FCF remains crisp. We observe in Fig. 4 that the FCF values around the area of G_1 are red, and those around the area of G_2 and G_3 are blue. This implies that the effect of m_2 is to fuzzify the FCF around each cluster uniformly. Furthermore, the fuzzification by m_2 also brings the positions of the cluster centers closer together. The mechanism of this difference and its effect on clustering accuracy will be investigated theoretically in future work. Fig. 6 shows the FCFs for the first

cluster obtained using the proposed method with $(m_1, m_2, \lambda) = (1.2, 1.2, 10)$, and Fig. 7 shows the FCFs for the first cluster obtained using the QFCS method with $(m, \lambda) = (1.2, 10)$. A comparison of these two figures confirms that the proposed method with $m_1 = m_2$ produces the same result as the QFCS method.

For the second experiment, we used two real datasets, called ‘‘Cora’’ and ‘‘CiteSeer,’’ obtained from [5]. In the Cora dataset, 2708 documents are classified into seven topics. Each document is represented by a 1432-dimensional binary-valued vector, each element of which corresponds to a unique word and indicates whether or not that word appears. In the CiteSeer dataset, 3312 documents are classified into six topics. Each document is represented by a 3703-dimensional binary-valued vector, each element of which corresponds to a unique word and indicates whether or not that word appears. In the experiment, these datasets were clustered using the RQFCS and QFCS methods. The number of clusters, C , was set to the correct number of topics for each dataset. For each algorithm, the fuzzification parameter λ was set from $\lambda \in \{10^{0 \times 5 + 1}, 10^{1 \times 5 + 1}, \dots, 10^{3 \times 5 + 1}\}$. The fuzzification parameters m_1 and m_2 for the RQFCS and m for QFCS were set from $m, m_1, m_2 \in \{1 + 10^{-1}, 1 + 10^{-2}, \dots, 1 + 10^{-4}\}$. The initial object memberships were set according to the actual class labels. Clustering accuracy was measured using the adjusted Rand index (ARI) [6]. ARI values lies within $[-0.5, 1]$, and higher values are preferred. The highest ARI value for each method and the parameter values at which this was achieved are shown in Tables 1 and 2; the higher of the ARI values given by the two methods is underlined. As shown in these tables, RQFCS outperformed QFCS in terms of accuracy with both datasets; therefore, it can be concluded that separating the fuzzification parameter m of QFCS into the two parameters m_1 and m_2 of RQFCS leads to more accurate clustering.

5 Conclusion

In this paper, we have proposed the RQFCS algorithm, extending the two-parameter QFCS method to create a three-parameter model. The effects of the fuzzification parameters were investigated by numerical experiments using an artificial dataset. In addition, through numerical experiments using two real datasets, it was demonstrated that the proposed method produced results having greater clustering accuracy than those produced by the QFCS method.

In our future research,

- The two fuzzification parameters m_1 and m_2 will be further investigated theoretically to characterize their fuzzification mechanism and their differing effects on clustering accuracy.
- The clustering accuracy of the proposed algorithm will be compared with that of conventional methods using large numbers of real datasets.

References

1. Dhillon, I. S. and Modha, D. S.: ‘‘Concept Decompositions for Large Sparse Text Data Using Clustering,’’ *Machine Learning*, Vol. 42, pp. 143–175, (2001).

2. Miyamoto, S. and Mizutani, Y.: “Fuzzy Multiset Model and Methods of Nonlinear Document Clustering for Information Retrieval,” Lecture Notes in Computer Science (LNCS), Vol. 3131, pp. 273–283, (2004).
3. Higashi, M., Kondo, T., and Kanzawa, Y.: “Fuzzy Clustering Method for Spherical Data Based on q-Divergence,” JACIII, Vol. 23, No. 3, pp. 561–570, (2019).
4. Torra, V.: “Fuzzy microaggregation for the transparency principle,” Journal of Applied Logic, Vol. 23, pp. 70–80, (2017).
5. Lise’s Inquisitive Students, Machine Learning Research Group @UMD.: available from: <http://www.cs.umd.edu/~sen/lbc-proj/LBC.html>
6. Hubert, L., and Arabie, P.: “Comparing Partitions,” Journal of Classification, Vol. 2, pp. 193–218, (1985).

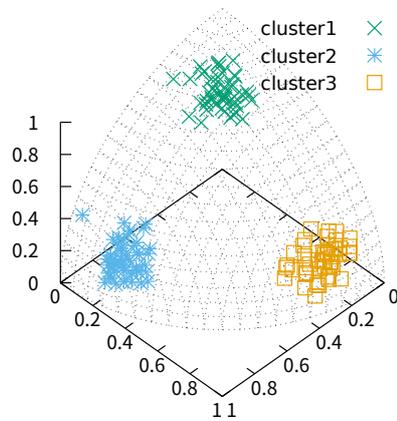


Fig. 1: Artificial dataset.

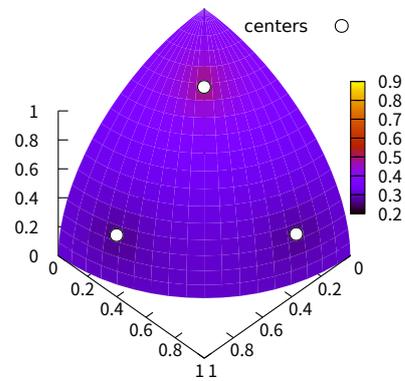


Fig. 3: FCF of the RQFCS method: $(m_1, m_2, \lambda) = (10.0, 1.1, 10)$.

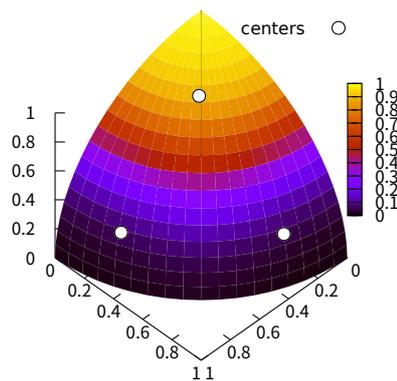


Fig. 2: FCF of the RQFCS method: $(m_1, m_2, \lambda) = (1.05, 1.1, 10)$.

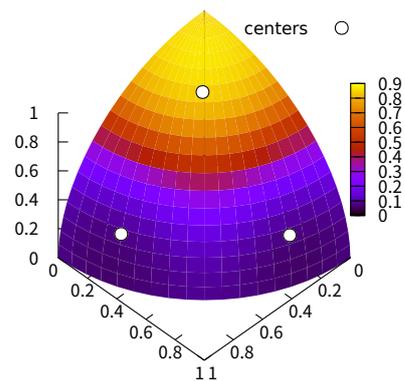


Fig. 4: FCF of the RQFCS method: $(m_1, m_2, \lambda) = (1.5, 1.001, 10)$.

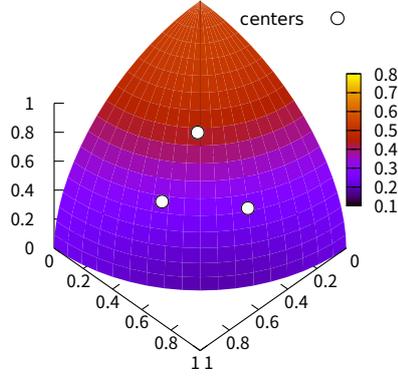


Fig. 5: FCF of the RQFCS method:
 $(m_1, m_2, \lambda) = (1.5, 2.3, 10)$.

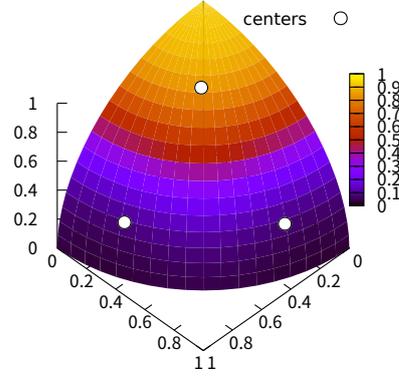


Fig. 7: FCF of the QFCS method:
 $(m, \lambda) = (1.2, 10)$.

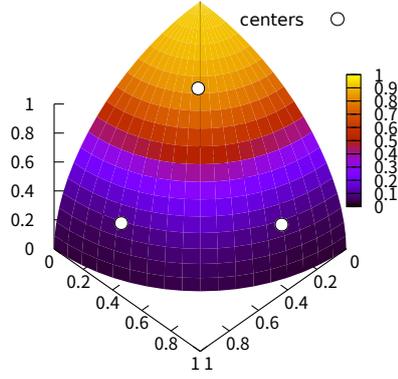


Fig. 6: FCF of the RQFCS method:
 $(m_1, m_2, \lambda) = (1.2, 1.2, 10)$.

Table 1: Highest ARI value for each method and the corresponding parameter value; for the “Cora” dataset.

Method	ARI	Parameter Value
QFCS	0.175945	$(m, \lambda) = (1 + 10^{-5}, 1000)$
RQFCS	0.176728	$(m_1, m_2, \lambda) = (1 + 10^{-5}, 1 + 10^{-1}, 1000)$

Table 2: Highest ARI value for each method and the corresponding parameter value; for the “CiteSeer” dataset.

Method	ARI	Parameter Value
QFCS	0.332952	$(m, \lambda) = (1 + 10^{-5}, 1000)$
RQFCS	0.333434	$(m_1, m_2, \lambda) = (1 + 10^{-5}, 1 + 10^{-4}, 1000)$

Aggregation for Hesitant Fuzzy sets

Yasuo Narukawa¹ and Vicenç Torra²

¹ Department of Management Science, Tamagawa University, 6-1-1 Tamagawagakuen,
Machida, Tokyo, 194-8610, Japan

`nrkwy@eng.takagawa.ac.jp`

² Dept. of Computing Sciences, Umeå University, Umeå, Sweden

`vtorra@ieee.org`

Abstract. Hesitant fuzzy sets were introduced as an extension of fuzzy sets. Its motivation is that membership degrees are not just a value in $[0,1]$ but a set of values in this interval. This is to account for the uncertainty in assigning these numbers.

Membership functions for hesitant fuzzy sets make that the elements in the reference set are not fully comparable with respect to membership. That is, membership functions define only partial order. Score functions permit to build total orders from membership functions.

In this paper we study aggregation for hesitant fuzzy sets, and we discuss their role for score functions.

1 Introduction

Among the existing extensions [6] for fuzzy sets [33] we find hesitant fuzzy sets, which we introduced in 2009 [25, 26]. While in fuzzy sets membership values are in $[0,1]$, hesitant fuzzy sets have membership defined as subsets of $[0,1]$. Typically, these sets are finite. These are typical hesitant fuzzy sets.

We have showed in previous papers their relationship with other types of extensions, including Interval-valued and Atanassov's intuitionistic fuzzy sets [3, 4], and type-n fuzzy sets [12] for details. Note that there is a discussion on the terminology related to intuitionistic fuzzy sets (see e.g. Dubois et al. [11]).

In this paper we study aggregation for hesitant fuzzy sets, and how they can be used for defining score functions for hesitant fuzzy sets. We also present some results about weak orders for these sets. Some of our results are applications of fuzzy integrals and some of their generalizations (e.g., the generalized t-conorm integral).

The structure of this paper is as follows. After providing some definitions in Sections 2, 3, and 4 that we need later, we introduce results related to score functions for hesitant fuzzy sets, and, more particularly, about orders induced by the score functions. These results are in Sections 5 and 6. Then, we propose (Section 7) the use of a generalized t-conorm integral, providing some additional results for this type of integral. The paper finishes with a conclusions section.

2 Hesitant Fuzzy Sets

Let us introduce in this section hesitant fuzzy sets on a reference set X .

Definition 1. [25] Let X be a reference set, then a hesitant fuzzy set on X is defined in terms of a function h that for each $x \in X$ returns a subset of $[0, 1]$.

The function $h(x)$ is called a hesitant fuzzy element by Z.S. Xu [32]. Note that for typical hesitant fuzzy sets, $h(x)$ is a finite set of values (see e.g. [1, 5]). For examples and relationships to other types of fuzzy sets see e.g. [25, 22, 21].

The following basic operations were proposed for hesitant fuzzy sets.

Definition 2. [25] Let h, h_1 , and h_2 be hesitant fuzzy sets. Then, the following operations are defined:

– Lower bound:

$$h^-(x) = \min h(x)$$

– Upper bound:

$$h^+(x) = \max h(x)$$

– α -upper bound:

$$h_\alpha^+(x) = \{h \in h(x) | h \geq \alpha\}$$

– α -lower bound:

$$h_\alpha^-(x) = \{h \in h(x) | h \leq \alpha\}$$

– Complement:

$$h^c(x) = \cup_{\gamma \in h(x)} \{1 - \gamma\}$$

– Union:

$$(h_1 \cup h_2)(x) = \{h \in (h_1(x) \cup h_2(x)) | h \geq \max(h_1^-, h_2^-)\},$$

or, equivalently

$$(h_1 \cup h_2)(x) = (h_1(x) \cup h_2(x))_\alpha^+$$

for $\alpha = \max(h_1^-, h_2^-)$.

– Intersection:

$$(h_1 \cap h_2)(x) = \{h \in (h_1(x) \cup h_2(x)) | h \leq \min(h_1^+, h_2^+)\},$$

or, equivalently,

$$(h_1 \cap h_2)(x) = (h_1(x) \cup h_2(x))_\alpha^-$$

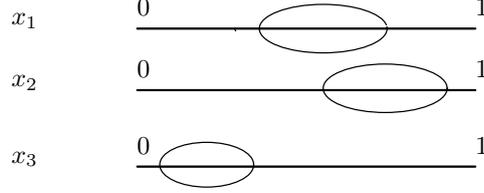
for $\alpha = \min(h_1^+, h_2^+)$.

The definition of the union of two hesitant fuzzy sets is based on the following rationale: if a hesitant fuzzy set is understood as a set of possible alternatives, it is clear that, for a given x , the lower bound of the $h_1 \cup h_2$ is the largest of the two h_1^-, h_2^- . The definition of the intersection follows a similar approach.

Example 1. Let us consider the problem of evaluating “good” papers, expressed by hesitant fuzzy sets h_1, h_2 .

Let us consider the set of attributes $X := \{x_1, x_2, x_3\}$. E.g. x_1 : Originality, x_2 : Significance, x_3 : Presentation.

Then, let us consider that one referee evaluates a paper h_1 using the lines below (for the three attributes above).



This information can be represented using hesitant fuzzy sets as follows:

- $h_1(x_1) = [0.4, 0.7]$
- $h_1(x_2) = [0.7, 0.8]$
- $h_1(x_3) = [0.2, 0.4]$

3 Binary operations for hesitant fuzzy sets

Let us review some binary operations. We define first t-norm and t-conorm. We denote t-norms and t-conorms, respectively, by \top and \perp .

Definition 3. A t-conorm is a binary function $[0, 1] \times [0, 1] \rightarrow [0, 1]$ such that

- (i) $\perp(a, 0) = a$,
- (ii) for $b \leq c$ then $\perp(a, b) \leq \perp(a, c)$,
- (iii) $\perp(a, b) = \perp(b, a)$,
- (iv) $\perp(a, \perp(b, c)) = \perp(\perp(a, b), c)$.

A t-norm is a binary function $[0, 1] \times [0, 1] \rightarrow [0, 1]$ such that

- (i) $\top(a, 1) = a$.
- (ii) for $b \leq c$ then $\top(a, b) \leq \top(a, c)$,
- (iii) $\top(a, b) = \top(b, a)$,
- (iv) $\top(a, \top(b, c)) = \top(\top(a, b), c)$.

We say that a t-conorm \perp is Archimedean if and only if $x \perp x > x$ for all $x \in (0, 1)$.

For details see e.g. [2, 16].

Examples of t-conorms include the maximum and the bounded sum. We denote the former by $\perp = \vee$ and the latter by $\perp(a, b) = \hat{\perp}(a, b) = \min(1, a + b)$. We will also consider Sugeno family of t-conorms. They are defined by the following expression $\perp_\lambda(a, b) = 1 \wedge (a + b + \lambda ab)$ for $(-1 < \lambda < \infty)$. We will denote these t-conorms by $a +_\lambda b$. Sugeno t-conorms are Archimedean t-conorms.

Next we define the binary operation on a hesitant fuzzy set.

Definition 4. Let X be a reference set, \perp a t -conorm, $x_1, x_2 \in X$ and h be a hesitant fuzzy set on X .

Define a binary operation \oplus on $h(X)$ by

$$h(x_1) \oplus h(x_2) = \{a_1 \perp a_2 \mid a_1 \in h(x_1), a_2 \in h(x_2)\}.$$

The next proposition is immediately from the definition.

Proposition 1. Let X be a reference set, $x_1, x_2, x_3 \in X$ and h be a hesitant fuzzy set on X .

Then we have $(h(x_1) \oplus h(x_2)) \oplus h(x_3) = h(x_1) \oplus (h(x_2) \oplus h(x_3))$

It follows from the proposition above that we can write

$$h(x_1) \oplus h(x_2) \oplus h(x_3) = (h(x_1) \oplus h(x_2)) \oplus h(x_3).$$

Therefore we define

$$\bigoplus_{k=1}^n h(x_k) = h(x_1) \oplus h(x_2) \oplus \cdots \oplus h(x_n)$$

for $x_1, x_2, \dots, x_n \in X$.

Next we define a scalar multiplication.

Definition 5. Let X be a reference set, $x \in X$, \boxtimes a t -norm, and h be a hesitant fuzzy set on X and let $0 \leq \alpha \leq 1$.

Define $\alpha \boxtimes h(x)$ by

$$\alpha \boxtimes h(x) = h(x) \boxtimes \alpha = \{\alpha \boxtimes a \mid a \in h(x)\}.$$

.

4 Aggregation functions and fuzzy integrals

This section introduces some concepts related to fuzzy integrals that are needed later on in this paper. We begin reviewing the concept of aggregation function. We understand them as functions that combine n values in the $[0,1]$ interval and return another value in the same interval. We will denote them by \mathbb{C} . A function $\mathbb{C} : [0,1]^n \rightarrow [0,1]$ is an aggregation function if it is monotonic, and satisfies unanimity. Some authors (e.g., [27]) require unanimity for all $a \in [0,1]$, while others require unanimity only for 0 and 1.

Following the notation above, we use X to denote the reference set. Then, let \mathcal{B} denote a subset of the power set of X (i.e., 2^X) such that $\emptyset \in \mathcal{B}$. An element of \mathcal{B} is said to be a *fuzzy measurable set* and, then, (X, \mathcal{B}) is a *fuzzy measurable space*. In addition, we say that a function $f : X \rightarrow R$ is *measurable* when

$$\{x \mid f(x) > r\} \in \mathcal{B}$$

for all $r \in R$. We denote the class of measurable functions by \mathcal{M} . In addition, we denote by \mathcal{M}^+ the class of non-negative measurable functions.

Definition 6. A fuzzy measure (also known as non-additive measure and capacity) μ is a real valued set function $\mu : \mathcal{B} \rightarrow [0, 1]$ that satisfies the following properties:

1. $\mu(\emptyset) = 0$ (boundary condition)
2. $\mu(X) = 1$ (boundary condition)
3. $A \subset B$ for $A, B \in \mathcal{B}$ implies $\mu(A) \leq \mu(B)$ (monotonicity condition)

Among the different families of fuzzy measures, some are of interest in this work. We define them below.

Definition 7. Let X be a set, then we consider fuzzy measures on (X, \mathcal{B}) .

1. *Probability measures.* A measure P is a probability measure if it satisfies the additivity axiom. That is, for all $A \cap B = \emptyset$ we have that

$$P(A \cup B) = P(A) + P(B),$$

and, in addition,

$$P(X) = 1.$$

2. *Possibility measures.* A measure Pos is a possibility measure if it satisfies the following axiom

$$Pos(A \cup B) = \max(Pos(A), Pos(B))$$

for all A, B . These measures were introduced by Zadeh [34] in the context of fuzzy sets.

3. *Necessity measure.* A measure Nec is a necessity measure if it satisfies

$$Nec(A \cap B) = \min(Nec(A), Nec(B))$$

for all A, B . These measures were also introduced by Zadeh [34] and they can be defined as conjugate of possibility measures.

4. *The 0-1 possibility measure Pos_A focused on a set $A \subseteq X$.* Given a set A we define the measure as follows.

$$Pos_A(B) = \begin{cases} 1 & \text{if } A \cap B \neq \emptyset \\ 0 & \text{if } A \cap B = \emptyset \end{cases}$$

5. *The 0-1 necessity measure Nec_A focused on a set $A \subseteq X$.* This measure is defined as follows, and corresponds to the unanimity game [15].

$$Nec_A(B) = \begin{cases} 1 & \text{if } A \subseteq B \\ 0 & \text{if } A \not\subseteq B \end{cases}$$

6. *Distorted probabilities.* A fuzzy measure μ is a distorted probability if it can be described in terms of a function Q and a probability distribution P as $\mu(A) = Q(P(A))$ for all A , and Q is a non-decreasing function Q such that $Q(0) = 0$ and $Q(1) = 1$. When P is just the Lebesgue measure, then we call the measure μ a distorted Lebesgue measure. We will denote the Lebesgue measure by λ as usual.

Let us now introduce fuzzy integrals. They permit to integrate a function with respect to a fuzzy measure. We will consider functions $f : X \rightarrow R$. Then, a fuzzy integral of f with respect to a fuzzy measure μ will provide a value in R that combines the values $f(x)$ for $x \in X$ taking into account μ . When $\mu(X) = 1$, these integrals can be seen as aggregation functions, where the values being aggregated are the values $f(x)$.

Definition 8. [9] Let X be a set, let f be a function on X as above, and let μ be a fuzzy measure on (X, \mathcal{B}) . Then, the Choquet integral of $f \in \mathcal{M}^+$ with respect to μ is defined by

$$(C) \int f d\mu = \int_0^\infty \mu_f(r) dr,$$

where $\mu_f(r) = \mu(\{x | f(x) \geq r\})$.

We also need to consider the restriction of the integral on a set. Let $A \in \mathcal{B}$ be such set. Then, the Choquet integral of f with restricted domain A is defined as follows:

$$(C) \int_A f d\mu = \int_0^\infty \mu(A \cap \{x | f(x) \geq r\}) dr.$$

From the above definitions, it is obvious the next theorem.

Theorem 1. Let (X, \mathcal{B}) be a measurable space, let f be a nonnegative measurable function on X and $A \in \mathcal{B}$. Then,

1. $(C) \int f d\text{Pos}_A = \sup_{x \in A} f(x)$ where Pos_A is the 0-1 possibility measure focused on A .
2. $(C) \int f d\text{Nec}_A = \inf_{x \in A} f(x)$ where Nec_A is the 0-1 necessity measure focused on A .

Sugeno integral is another integral that also permits to integrate a function with respect to a fuzzy measure.

Definition 9. [23] Let X be a set, let f be a function on X as above, and let μ be a fuzzy measure on (X, \mathcal{B}) . Then, the Sugeno integral of a function $f \in \mathcal{M}^+$ with respect to μ is defined by

$$S_\mu(f) := \sup_{r \in [0,1]} [r \wedge \mu_f(r)],$$

where $\mu_f(r) := \mu(\{x | f(x) > r\})$.

These two definitions lead to different outcomes for the same pair f and μ . The generalized t -conorm integral [24, 17] generalizes both integrals exploiting the similarities between the two definitions.

Let us begin the definitions with the one of a t -system, and then the one of generalized t -conorm integral.

Definition 10. [17] A *t*-conorm system for integration (or *t*-system for short) is a quadruplet (F, M, I, \square) where the first objects are continuous *t*-conorms $F := ([0, 1], \perp_1)$, $M = ([0, 1], \perp_2)$, $I = ([0, 1], \perp_3)$, where \perp_i $i = 1, 2, 3$ which are \vee or Archimedean, and the fourth object is a non decreasing operator $\square : F \times M \rightarrow I$, and where these operators satisfy the following conditions:

- (M1) \square is left continuous on $(0, 1]$.
- (M2) $a \square x = 0$ if and only if $a = 0$ or $x = 0$.
- (M3) if $x \perp_2 y < 1$ then $a \square (x \perp_2 y) = (a \square x) \perp_3 (a \square y)$.
- (M4) if $a \perp_1 b < 1$ then $(a \perp_1 b) \square x = (a \square x) \perp_3 (b \square x)$.

When there are generators of the *t*-conorms \perp_i for $i = 1, 2, 3$, then we use g_i for $i = 1, 2, 3$ to denote them.

For the sake of brevity, we will also use $(\perp_1, \perp_2, \perp_3, \square)$ to denote the *t*-system. For example, we denote by $(+\lambda, \vee, \hat{+}, \cdot)$ the system with $\perp_1 = +\lambda$, $\perp_2 = \vee$, $\perp_3 = \hat{+}$, $\square = \cdot$. Here, \cdot represents the ordinary multiplication. Similarly, $(\vee, \vee, \vee, \wedge)$ denotes $\perp_1 = \perp_2 = \perp_3 = \vee$, and $\square = \wedge$.

As the definition requires the *t*-conorms to be either Archimedean *t*-conorms or the maximum operator \vee , the following four types of *t*-systems are considered:

1. \perp_i for $i = 1, 2, 3$ are Archimedean,
2. $\perp_1 = \perp_2 = \perp_3 = \vee$,
3. \perp_3 is Archimedean and at least one of the other is \vee ,
4. $\perp_3 = \vee$ and at least one of the other is Archimedean.

Sugeno and Murofushi [24, 17] (see also [27] for details) discuss and prove that among these four types of *t*-systems only types (1) and (2) have a rich structure. Taking this into account, here we only consider Archimedean *t*-systems (i.e., type (1) above) and *t*-systems of \vee type (i.e., type (2) above).

Let us now introduce another operation related to a given *t*-conorm \perp . It is defined as follows $a -_{\perp} b = \inf\{c | a \perp b \geq c\}$.

A fuzzy measure \perp is decomposable if $\mu(A \cup B) = \perp(\mu(A), \mu(B))$ for disjoint A, B . Then, a \perp -decomposable fuzzy measure μ with generator g is called normal when $\perp = \vee$ or $g \circ \mu$ is an infinite additive measure or $g \circ \mu$ is a finite additive measure.

Now we introduce the integrals. We consider first the integrals for simple functions. A function f is a simple function if $f = \perp_{i=1}^n a_i 1_{D_i}$, where $D_i \cap D_j \neq \emptyset$ for $i \neq j$.

Definition 11. [17] Let $(X, 2^X, m)$, (\perp_i, \square) for $i = 1, 2, 3$ as above. Let m be a normal \perp_2 -decomposable fuzzy measure. Then, for a simple function $f : X \rightarrow [0, 1]$, the *t*-conorm integral is defined as follows:

$$(T) \int f \square dm := \perp_{i=1}^n a_i \square m(D_i).$$

Suppose that $f = \perp_{i=1}^n (a_i \text{--} \perp_1 a_{i-1}) 1_{A_i}$ where $A_i = \cup_{j=i}^n D_j$, we can define another integral.

Definition 12. [17] Let $(X, 2^X, m)$, (\perp_i, \boxminus) for $i = 1, 2, 3$ as above. Then, for a function $f : X \rightarrow [0, 1]$ the generalized t-conorm integral is as follows.

$$(GT) \int f \boxminus dm := \perp_{i=1}^n (a_i \text{--} \perp_1 a_{i-1}) \boxminus m(A_i).$$

We will define another generalized t-conorm integral.

Definition 13. Let $(X, 2^X, m)$, (\perp_i, \boxminus) for $i = 1, 2, 3$ as above. Then, for a function $f : X \rightarrow [0, 1]$ the generalized t-conorm integral (GT2) is as follows. If f is a simple function

$$(GT2) \int f \boxminus dm := \perp_{i=1}^n a_i \boxminus (m(A_i) \text{--} \perp_2 m(A_{i-1})).$$

We provide now definitions constrained to a set A .

Definition 14. [17] Let $(X, 2^X, m)$, (\perp_i, \boxminus) for $i = 1, 2, 3$, and f as above. Then, for $A \subset X$, the generalized t-conorm integral on A is defined as follows:

$$(GT) \int_A f \boxminus dm := \perp_{i=1}^n (a_i \text{--} \perp_1 a_{i-1}) \boxminus m(A_i \cap A).$$

Definition 15. Let $(X, 2^X, m)$, (\perp_i, \boxminus) for $i = 1, 2, 3$, and f as above. Then, for $A \subset X$, the generalized t-conorm integral GT2 on A is defined as follows:

$$(GT2) \int_A f \boxminus dm := \perp_{i=1}^n a_i \boxminus (m(A_i \cap A) \text{--} \perp_2 m(A_{i-1} \cap A)).$$

The following lemma links the generalized t-conorm integral with Choquet and Sugeno integrals.

- Lemma 1.**
1. For $\perp_1 = \perp_2 = \perp_3 = \hat{\dagger}$ and $\boxminus = \cdot$, the generalized t-conorm integral is a Choquet integral (i.e., Def. 8).
 2. For $\perp_1 = \perp_2 = \perp_3 = \vee$ and $\boxminus = \wedge$, the generalized t-conorm integral is a Sugeno integral (i.e., Def. 9).

5 Score function for a hesitant fuzzy set

We now define two score functions. The first one is based on the Choquet integral and the second one is based on the generalized t-conorm integral. These score functions compute a value for a given element x of the reference set X by means of considering the membership values in $h(x)$. The definition requires a fuzzy measure.

Definition 16. Let \mathcal{H}_X be the set of all hesitant fuzzy sets on X and μ be a fuzzy measure on $([0, 1], \mathcal{B})$ where \mathcal{B} is a class of Borel sets.

The CI-score function $s_\mu^{CI}(h_x)$ of $h \in \mathcal{H}_X$ for $x \in X$ with respect to a fuzzy measure μ is defined by

$$s_\mu^{CI}(h_x) := (C) \int_{h_x} r d\mu|_{h_x}(r).$$

where $\mu|_{h_x}(A) = \frac{\mu(A \cap h_x)}{\mu(h_x)}$.

Similarly, the GI-score function is defined by

$$s_\mu^{GI}(h_x) := (GT) \int_{h_x} r \square d\mu|_{h_x}(r)$$

where $\mu|_{h_x}(A) = \frac{\mu(A \cap h_x)}{\mu(h_x)}$.

Let us consider that the range of a hesitant fuzzy set is the closed interval $[a, b] \subset [0, 1]$. In this case, we will study the score function $s_{\mu_n}^{CI}(h_x)$ of $h(x)$ with respect to a distorted Lebesgue measure λ^α [20]. Observe that λ^α means that $\lambda^\alpha(A) = (\lambda(A))^\alpha$. We will consider $\alpha \in (0, \infty)$.

$$\begin{aligned} (C) \int_{[a,b]} r d\lambda^\alpha(r) &= \int \lambda^\alpha([a, b] \cap \{x | 1 \geq x \geq r\}) dr \\ &= \frac{(b-a)^\alpha (\alpha a + b)}{(1+\alpha)}, \end{aligned}$$

the following proposition holds.

Proposition 2. Let X be a reference set, let $h \in \mathcal{H}_X$ be a hesitant fuzzy set, and assume that $h(x)$ is a closed interval $[a_x, b_x]$ for all $x \in X$. Then,

$$s_{\lambda^\alpha}^{CI}(h_x) := \frac{\alpha a_x + b_x}{(1+\alpha)}.$$

From the last equation it is easy to see that for $\alpha = 1$,

$$s_\lambda^{CI}(h_x) := \frac{(a_x + b_x)}{2}.$$

That is, for $\alpha = 1$, the score function of h_x is the center of the closed interval (or the centroid). Using the equation above, we can also prove the following result.

Proposition 3. Let X be a reference set, let $h \in \mathcal{H}_X$ be a hesitant fuzzy set, and assume that $h(x)$ is the closed interval $A \subset [0, 1]$. Then we have

- $s_{PosA}^{CI}(h_x) := \lim_{\alpha \rightarrow 0} s_{\lambda^\alpha}^{CI}(h_x)$
- $s_{NegA}^{CI}(h_x) := \lim_{\alpha \rightarrow \infty} s_{\lambda^\alpha}^{CI}(h_x)$.

6 On weak orders in HFS

We define now the order induced by the score function $s_\mu(h_x)$.

Definition 17. Let h be a hesitant fuzzy set on X and $x_1, x_2 \in X$, and let μ be a fuzzy measure on $([0, 1], \mathcal{B})$ where \mathcal{B} is a class of Borel sets.

Then, the order \prec_{s_μ} induced by the score function $s_\mu(h(x))$ is defined by

$$h(x_1) \prec_{s_\mu} h(x_2)$$

if and only if

$$s_\mu(h_{x_1}) < s_\mu(h_{x_2}).$$

If $\mu = \lambda^\alpha$, we denote $h(x_1) \prec_{s_\mu} h(x_2)$ by $h(x_1) \prec_\alpha h(x_2)$.

In particular, $h(x_1) \prec_0 h(x_2)$ means $h(x_1) \prec_{Pos} h(x_2)$, and $h(x_1) \prec_\infty h(x_2)$ means $h(x_1) \prec_{Nec} h(x_2)$.

Next we define the incomparable relation by

$$h(x_1) \sim_{s_\mu} h(x_2)$$

if and only if

$$s_\mu(h_{x_1}) = s_\mu(h_{x_2}).$$

\sim_α, \sim_0 and \sim_∞ are similiary defined.

In the following, we suppose that $h(x)$ is a closed interval $A \subset [0, 1]$, that is, $A = [a_x, b_x]$.

Let $x_1, x_2 \in X$ and $\alpha > 0$. If $h(x_1) \sim_\alpha h(x_2)$, generally $h(x_1) \neq h(x_2)$.

Suppose $h(x_1) \sim_\alpha h(x_2)$ and $h(x_1) \sim_0 h(x_2)$ or $h(x_1) \sim_\infty h(x_2)$ then $h(x_1) = h(x_2)$.

Therefore we can define the total order $\prec_{t\alpha}$.

Definition 18. Let h be a hesitant fuzzy set on X , $x_1, x_2 \in X$ and $\alpha > 0$.

If $h(x_1) \prec_\alpha h(x_2)$, then $h(x_1) \prec_{t\alpha} h(x_2)$.

If $h(x_1) \sim_\alpha h(x_2)$ and $h(x_1) \prec_0 h(x_2)$, then $h(x_1) \prec_{t\alpha} h(x_2)$.

If $h(x_1) \sim_\alpha h(x_2)$ and $h(x_1) \sim_0 h(x_2)$, then $h(x_1) \sim_{t\alpha} h(x_2)$.

The following proposition can be proven for $\sim_{t\alpha}$.

Proposition 4. Let h be a hesitant fuzzy set on X , $x_1, x_2 \in X$ and $\alpha > 0$.

If $h(x_1) \sim_{t\alpha} h(x_2)$, then $h(x_1) = h(x_2)$.

Definition 19. Let h be a hesitant fuzzy set on X , $x_1, x_2 \in X$ and $\alpha > 0$.

$h(x_1) \preceq_{t\alpha} h(x_2)$ denotes $h(x_1) \prec_{t\alpha} h(x_2)$ or $h(x_1) \sim_{t\alpha} h(x_2)$.

Using $\prec_{t\alpha}$, we can define a weak order for hesitant fuzzy sets.

Definition 20. Let h_1, h_2 be hesitant fuzzy sets on X and $\alpha > 0$.

$h_1 \prec_{t\alpha} h_2$ denotes $h_1(x) \prec_{t\alpha} h_2(x)$ for all $x \in X$.

7 Aggregation for a hesitant fuzzy set using a generalized t-conorm integral

In this section we consider a generalized t-conorm integral. We begin with its definition.

Definition 21. Let $X = \{x_1, x_2, \dots, x_n\}$ and ν be a fuzzy measure on $(X, 2^X)$.

Let $h \in \mathcal{H}_X$ be a hesitant fuzzy set and $\alpha \geq 0$.

We will define the aggregation of h with respect to the fuzzy measure ν using a generalized t-conorm integral by

$$(GT2) \int_{\alpha} h \boxplus d\nu = \oplus_{k=1}^n h(x_k) \boxplus (\nu(\{x_k, x_{k+1}, \dots, x_n\}) \perp_{\alpha} \nu(\{x_{k+1}, \dots, x_n\}))$$

where $X = \{x_1, x_2, \dots, x_n\}$ is rearranged so that $h(x_1) \preceq_{t\alpha} h(x_2) \preceq_{t\alpha} \dots \preceq_{t\alpha} h(x_n)$.

We illustrate this definition with an example.

Example 2. Let $\perp_1 = \perp_2 = \hat{+}$ and $\boxplus = \cdot$, then $(GT2) \int_{\alpha} h \boxplus d\nu$ is the extended Choquet integral to a closed interval, that is:

$$(C) \int_{\alpha} h \cdot d\nu = \oplus_{k=1}^n h(x_k) (\nu(\{x_k, x_{k+1}, \dots, x_n\}) - \nu(\{x_{k+1}, \dots, x_n\}))$$

where $X = \{x_1, x_2, \dots, x_n\}$ is rearranged so that $h(x_1) \preceq_{t\alpha} h(x_2) \preceq_{t\alpha} \dots \preceq_{t\alpha} h(x_n)$.

The following proposition follows from the properties of the generalized t-conorm integral.

Proposition 5. Let $X = \{x_1, x_2, \dots, x_n\}$ and ν be a fuzzy measure on $(X, 2^X)$.

Let $h, h_1, h_2 \in \mathcal{H}_X$ be a hesitant fuzzy set and $\alpha \geq 0$.

1. $h_1 \prec_{t\alpha} h_2$ implies

$$(GT2) \int_{\alpha} h_1 \cdot d\nu \leq (GT2) \int_{\alpha} h_2 \cdot d\nu$$

2. If $h(x) = [a, b]$ for all $x \in X$, then

$$(GT2) \int_{\alpha} h \cdot d\nu = [a, b].$$

8 Conclusion

In this paper we have studied score functions for hesitant fuzzy sets. We have underlined the role of aggregation functions for this type of problem, and shown how fuzzy integrals can be used to define score functions. We have provided some mathematical results about orders inferred from the score functions.

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation

References

1. Alcantud, J. C. R., Torra, V. (2018) Decomposition theorems and extension principles for hesitant fuzzy sets, *Inf. Fusion* 41 48-56.
2. Alsina, C., Frank, M. J., Schweizer, B. (2006) *Associative Functions: Triangular Norms and Copulas*, World Scientific.
3. Atanassov, K. (1986) Intuitionistic fuzzy sets, *Fuzzy Sets and Systems* 20 87-96.
4. Atanassov, K. T. (1999) *Intuitionistic Fuzzy Sets*, Physica-Verlag, Heidelberg.
5. Bedregal, B., Reiser, R., Bustince, H., López-Molina, C., Torra, V. (2014) Aggregating functions for typical hesitant fuzzy elements and the action of automorphisms, *Information Sciences* 256 82-97.
6. Bustince, H., Barrenechea, E., Pagola, M., Fernández, J., Xu, Z., Bedregal, B., Montero, J., Herrera, F., Hagrass, H., De Baets, B. (2015) A historical account of types of fuzzy sets and their relationship, *IEEE Trans. on Fuzzy Systems*.
7. Calvo, T., Mayor, G. (1999) Remarks on two types of extended aggregation functions, *Tatra mountains* 16 235-253.
8. Calvo, T., Mayor, G., Torrens, J., Suñer, J., Mas, M., Carbonell, M. (2000) Generation of Weighting Triangles Associated with Aggregation Functions, *Int. J. of Unc., Fuzz. and Knowledge-Based Systems* 8:4 417-452.
9. Choquet, G. (1953/54) Theory of capacities, *Ann. Inst. Fourier* 5 131-295.
10. Dimuro, G. P., Fernandez, J., Bedregal, B., Mesiar, R., Sanz, J. A., Lucca, G., Bustince, H. (2020) The state-of-art of the generalizations of the Choquet integral: From aggregation and pre-aggregation to ordered directionally monotone functions, *Information Fusion* 57 27-43.
11. Dubois, D., Gottwald, S., Hajek, P., Kacprzyk, J., Prade, H. (2005) Terminological difficulties in fuzzy set theory - The case of Intuitionistic Fuzzy Sets, *Fuzzy Sets and Systems* 156:3 485-491
12. Dubois, D., Prade, H. (1980) *Fuzzy Sets and Systems*, Academic Press.
13. Dujmović, J. J., Torra, V. (2021) Properties and comparison of andness-characterized aggregators, *Int. J. of Intel. Syst.* 36:3 1366-1385.
14. Farhadinia, B. (2014) A series of score functions for hesitant fuzzy sets, *Information Sciences* 277 102-110.
15. Grabisch, M. (2016) *Set Functions, Games and Capacities in Decision Making*, Springer.
16. Klir, G. J., Yuan, B. (1995) *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, UK
17. Murofushi, T., Sugeno, M., (1991) Fuzzy t-conorm integral with respect to fuzzy measures: generalization of Sugeno integral and Choquet integral, *Fuzzy Sets and Systems*, 42, 57-71.
18. Narukawa, Y., Torra, V. (2021) An score index for hesitant fuzzy sets based on the Choquet integral, *Proc. FUZZ-IEEE 2021*: 1-5
19. Narukawa, Y., Torra, V. (2021) Score indices for hesitant fuzzy sets: generalized fuzzy integrals, *Proc. INFUS 2021*.

20. Narukawa, Y., Torra, V., Sugeno, M. (2016) Choquet integral with respect to a symmetric fuzzy measure of a function on the real line, *Ann Oper Res* 244, 571-581.
21. Rodríguez, R. M., Martínez, L., Herrera, F., Torra, V. (2016) A review of hesitant fuzzy sets: quantitative and qualitative extensions, in C. Kahraman, U. Kaymak, A. Yazici (eds) *Fuzzy Logic in its 50th Year*, Springer, 109-128.
22. Rodríguez, R. M., Martínez, L., Torra, V., Xu, Z. S., Herrera, F. (2014) Hesitant fuzzy sets: state of the art and future directions, *Int. J. of Intel. Systems* 29 495-524.
23. Sugeno, M., *Theory of fuzzy integrals and its applications*, Doctoral Thesis, Tokyo Institute of Technology, 1974.
24. Sugeno, M., Murofushi, T. (1987) Pseudo additive measures and integrals, *J. Math. Anal. Appl.* 122 197-222.
25. Torra, V., Hesitant fuzzy sets. *Int. J. Intell. Syst.* 2010 25:6 529-539.
26. Torra V, Narukawa Y., On hesitant fuzzy sets and decision. In *The 18th IEEE Int Conf on Fuzzy Systems*, 2009. pp. 1378-1382.
27. Torra, V., Narukawa, Y. (2007) *Modeling decisions: information fusion and aggregation operators*, Springer.
28. Torra, V., Narukawa, Y. (2008) The h-index and the number of citations: two fuzzy integrals, *IEEE Trans. on Fuzzy Systems* 16:3 795-797.
29. Torra, V., Narukawa, Y., Sugeno, M. (eds.) (2013) *Non-additive measures: theory and applications*, Springer.
30. Weber, S. (1984) \perp -decomposable measures and integrals for archimedean t -conorms \perp , *J. of Math. Analysis and Applications* 101 114-138.
31. Xia, MM, Xu, ZS. (2011) Hesitant fuzzy information aggregation in decision making. *Int. J. Approx. Reason* 52 395-407.
32. Xu, Z. (2014) *Hesitant Fuzzy Sets Theory*, Springer.
33. Zadeh, L. A. (1965) Fuzzy sets, *Inform. Control.* 8 338-353.
34. Zadeh, L. A. (1978) Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems* 1 3-28.