UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE INFORMÁTICA

LEARNING PREDICTIVE MODELS FROM TEMPORAL THREE-WAY DATA USING TRICLUSTERING: APPLICATIONS IN CLINICAL DATA ANALYSIS

Diogo Filipe Marques Soares

Mestrado em Ciência de Dados

Dissertação orientada por:

Professora Doutora Sara Alexandra Cordeiro Madeira

2020

“We’re here to put a dent in the universe.
Otherwise, why else even be here?”

— Steve Jobs

Acknowledgements

First of all, I want to thank my Family for their support throughout my academic path, especially my Parents, Luísa and António, who always believed in me and without them, none of this would have been possible, my sister Mara, my uncle José and my grandparents who were also always by my side.

A huge thanks to my friends and partners with whom I spent the last five years, and without their help and support, I would never get to where I am today. Especially to my "CtrlAltDElite" group (Pedro, Daniel and Diogo), my LASIGE partners Caseirito and Avelãs and to all my Academic Family: Ana Leonor, Bernardo, Eduardo, Gabriel, Guilherme, Henrique, Hugo and Vasco. I also want to express gratitude to all of my friends that I did not mention here, but that contributed to my academic success.

With that said, I can only thank everyone who made this work possible. My advisor, Sara Madeira, who allowed me to integrate her projects and thus be able to deepen my desire and ambition to research in order to learn more and help develop science. Her support and availability were essential to achieve the success of this work. Thanks to Instituto de Medicina Molecular - João Lobo Antunes (IMM) team, Susana Pinto, Marta Gromicho and Doctor Mamede de Carvalho for their work on collecting ALS patients data and giving more details and explanations about our machine learning results. Additionally, thanks to Rui Henriques, from Instituto Superior Técnico, for his work and support about the research in triclustering. Final thank to Faculdade de Ciências da Universidade de Lisboa that was my second home during the last years.

Finally, a formal acknowledgment to Fundação para a Ciência e Tecnologia (FCT) funding through projects NEUROCLINOMICS2 (PTDC/EEI-SII/1937/2014), iCare4U (LISBOA-01-0145-FEDER-031474 + PTDC/EME-SIS/31474/2017) and the LASIGE Research Unit (UIDB/00408/2020 + UIDP/00408/2020) and to EU funding through project CIRCLES (H2020/818290).

Resumo

O conceito de *triclustering* estende o conceito de *biclustering* para um espaço tridimensional, cujo o objetivo é encontrar subespaços coerentes em dados tridimensionais. Considerando dados com dimensão temporal, a necessidade de aprender padrões temporais interessantes e usá-los para aprender modelos preditivos efetivos e interpretáveis, despoleta necessidade em investigar novas metodologias para análise de dados tridimensionais. Neste trabalho, propomos duas metodologias para esse efeito.

Na primeira metodologia, encontramos os melhores parâmetros a serem usados em *triclustering* para descobrir os melhores *triclusters* (conjuntos de objetos com um padrão coerente ao longo de um dado conjunto de pontos temporais) para que depois estes padrões sejam usados como *features* por um dos mais apropriados classificadores encontrados na literatura. Neste caso, propomos juntar o classificador com uma abordagem de *triclustering* temporal. Para isso, idealizámos um algoritmo de triclustering com uma restrição temporal, denominado TCtriCluster para desvendar *triclusters* temporalmente contínuos (constituídos por pontos temporais contínuos). Na segunda metodologia, adicionámos uma fase de *biclustering* para descobrir padrões nos dados estáticos (dados que não mudam ao longo do tempo) e juntá-los aos triclusters para melhorar o desempenho e a interpretabilidade dos modelos. Estas metodologias foram usadas para prever a necessidade de administração de ventilação não invasiva (VNI) em pacientes com Esclerose Lateral Amiotrófica (ELA). Neste caso de estudo, aprendemos modelos de prognóstico geral, para os dados de todos os pacientes, e modelos especializados, depois de feita uma estratificação dos pacientes em 3 grupos de progressão: Lentos, Neutros e Rápidos. Os resultados demonstram que, além de serem bastante equiparáveis e por vezes superiores quando comparados com os resultados obtidos por um classificador de alto desempenho (*Random Forests*), os nossos classificadores são capazes de refinar as previsões através das potencialidades da interpretabilidade do modelo. De facto, quando usados os *triclusters* (e *biclusters*) como previsores, estamos a promover o uso de padrões de progressão da doença altamente interpretáveis. Para além disso, quando usados para previsão de prognóstico em doentes com ELA, os nossos modelos preditivos interpretáveis desvendaram padrões clinicamente relevantes para um grupo específico de padrões de progressão da doença, ajudando os médicos a entender a elevada heterogeneidade da progressão da ELA. Os resultados mostram ainda que a restrição temporal tem impacto na melhoria da efetividade e preditividade dos modelos.

Palavras Chave: *Triclustering*; Dados Tridimensionais; Modelos Preditivos; Padrões de Progressão de Doença; Esclerose Lateral Amiotrófica

Abstract

Triclustering extends biclustering to the three-dimensional space, aiming to find coherent subspaces in three-way data (sets of objects described by subsets of features in a subset of contexts). When the context is time, the need to learn interesting temporal patterns and use them to learn effective and interpretable predictive models triggers the need for new research methodologies to be used in three-way data analysis. In this work, we propose two approaches to learn predictive models from three-way data: 1) a triclustering-based classifier (considering just temporal data) and 2) a mixture of biclustering (with static data) and triclustering (with temporal data). In the first approach, we find the best triclustering parameters to uncover the best triclusters (sets of objects with a coherent pattern along a set of time-points) and then use these patterns as features in a state-of-the-art classifier. In the case of temporal data, we propose to couple the classifier with a temporal triclustering approach. With this aim, we devised a temporally constrained triclustering algorithm, termed TCtriCluster algorithm to mine time-contiguous triclusters. In the second approach, we extended the triclustering-based classifier with a biclustering task, where biclusters are discovered in static data (not changed over the time) and integrated with triclusters to improve performance and model explainability. The proposed methodologies were used to predict the need for non-invasive ventilation (NIV) in patients with Amyotrophic Lateral Sclerosis (ALS). In this case study, we learnt a general prognostic model from all patients data and specialized models after patient stratification into Slow, Neutral and Fast progressors. Our results show that besides comparable and sometimes outperforming results, when compared to a high performing random forest classifier, our predictive models enhance prediction with the potentialities of model interpretability. Indeed, when using triclusters (and biclusters) as predictors, we promoting the use of highly interpretable disease progression patterns. Furthermore, when used for prognostic prediction in ALS, our interpretable predictive models unravelled clinically relevant and group-specific disease progression patterns, helping clinicians to understand the high heterogeneity of ALS disease progression. Results further show that the temporal restriction is effective in improving the effectiveness of the predictive models.

Keywords: Triclustering; Three-way Data; Predictive Models; Disease Progression Patterns; Amyotrophic Lateral Sclerosis

Resumo Alargado

Considerando que o conceito de *biclustering*, cujo seu objetivo é encontrar subespaços coerentes numa matriz (ou tabela) de dados (conjuntos de objetos descritos por subconjuntos de características), não é suficientemente eficaz se considerarmos um espaço tridimensional, em que deixamos de ter apenas uma matriz de dados e passamos a ter um cubo de dados, é necessário estender esta abordagem para que possamos explorar também subespaços dos dados tendo em conta a sua terceira dimensão (conjuntos de objetos descritos por subconjunto de características num dado conjunto de contextos), surgindo assim o conceito de *triclustering*. Quando consideramos o tempo como contexto (terceira dimensão para o conjunto de dados), a necessidade de aprender padrões temporais interessantes e usá-los para aprender modelos preditivos que sejam efetivos e interpretáveis despoleta uma necessidade de investigar novas metodologias para análise de dados tridimensionais.

Até à atualidade, várias aplicações promissoras de *triclustering* em domínios clínicos têm vindo a ser utilizadas, tais como a análise de dados multivariados de comportamento fisiológico, em que os *triclusters* obtidos foram capazes de capturar respostas fisiológicas coerentes para grupos de indivíduos; a análise de dados de imagens cerebrais (através de ressonâncias magnéticas) em que os *triclusters* obtidos conseguiram identificar funções hemodinâmicas de resposta e conectividade entre regiões do cérebro e a análise de registos de saúde eletrónicos em que os *triclusters* são usados para identificar grupos de pacientes com determinadas características clinicas correlacionadas ao longo do tempo.

Neste trabalho propomos duas abordagens para análise de dados de 3 dimensões. A primeira usando apenas dados temporais para aprender modelos preditivos baseado em *triclustering*. Na segunda, juntamos dados estáticos para complementar a análise anterior e aproveitar os padrões descobertos para melhorar a o desempenho dos modelos.

Inicialmente, propomos uma abordagem para aprender modelos preditivos a partir de dados tridimensionais através de um classificador baseado em *triclustering*. Esta metodologia apresenta três fases. Na primeira fase, pretendemos encontrar os melhores parâmetros a serem passados ao algoritmo de *triclustering* para de modo a que sejam descobertos os melhores *triclusters* (conjuntos de objetos com um padrão coerente ao longo de um dado conjunto de pontos temporais) e depois usá-los como *features* num dos mais apropriados classificadores recentes encontrado na literatura. Para isso, propomos que em conjunto com o classificador seja usada uma abordagem de *triclustering* temporal, na qual é usado um algoritmo de *triclustering* que tenha em conta uma restrição temporal (neste caso a contiguidade dos pontos temporais), denominado TCtriCluster, idealizado para desvendar *triclusters* temporalmente con-

tíguos (constituído apenas por pontos temporais contíguos). Na fase seguinte, o modelo final é aprendido tendo por base os melhores *triclusters* obtidos anteriormente, culminando num um modelo preditivo final baseado em *triclustering* capaz de, além de fazer previsões, traçar perfis de características de acordo com os padrões mais relevantes identificados pelo modelo. A fase final desta metodologia, consiste em usar os *triclusters* e o respetivo modelo para, dado um determinado novo objeto, realizar uma previsão da sua classe e avaliar as suas semelhanças com determinados padrões relevantes descobertos na fase anterior.

A segunda metodologia foi concebida partindo da primeira e tirando proveito do conjunto de dados estáticos (dados que não mudam ao longo do tempo) que tipicamente existe, em contexto clínico, em conjunto com os dados temporais. Esta abordagem introduz uma mistura de *biclustering* nos dados com *triclustering* nos dados temporais. Os *biclusters* e *triclusters* são depois usados pelo classificador para construir modelos preditivos de modo a fazer previsões e traçar os perfis das características baseadas nos dados aprendidos.

As metodologias propostas foram usadas para prever a necessidade de ventilação não invasiva (VNI) em paciente com Esclerose Lateral Amiotrófica (ELA). A ELA é uma doença neurodegenerativa caracterizada por uma fraqueza muscular de progressão rápida, que tipicamente provoca a morte dos pacientes por paragem respiratória em 3 a 5 anos depois de diagnosticada. Dada a sua heterogeneidade, muitos pacientes podem viver menos de um ano, enquantro outros sobrevivem por mais de 10 anos com esta patologia. Em Portugal, a ELA afeta 10 em cada 100 mil habitantes. Muitos pacientes desenvolvem hipoventilação associada a hipoxemia e hipercapnia, passando a necessitar da ajuda de ventilação não invasiva. Neste sentido, torna-se importante prever o início destas complicações em pacientes com ELA, para que se possa intervir atempadamente e assim providenciar melhor qualidade de vida aos doentes.

Neste caso de estudo, foram usados dados de pacientes com ELA, recolhidos no Hospital de Santa Maria, em Lisboa (Centro Hospitalar Universitário de Lisboa), para aprender modelos de prognóstico geral a partir de todos os dados dos pacientes, bem como modelos especializados depois de feita uma estratificação dos pacientes em 3 grupos de progressão: Lentos, Neutros e Rápidos.

Os nossos resultados demonstram que, além de serem bastante equiparáveis e por vezes superiores quando comparados com os resultados obtidos por um classificador (*Random Forests*) de alto desempenho, os nossos classificadores baseados em *triclustering* são capaz de refinar as previsões através das potencialidades da interpretabilidade do modelo. De facto, quando usados os *triclusters* (e *biclusters*) como previsores, estamos neste caso a usar padrões de progressão da doença altamente interpretáveis. Para além disso, quando usados para previsão de prognóstico em doentes com ELA, os nossos modelos preditivos interpretáveis desvendaram padrões clinicamente relevantes e para um grupo específico de padrões de progressão da doença, ajudando os médicos a entender a elevada heterogeneidade da progressão da ELA. Os resultados mostram ainda que a restrição temporal tem impacto na melhoria da efetividade e preditividade dos modelos.

Contents

List of Figures

List of Tables

Acronyms

3W	Three-way (Data)
2W	Two-way (Data)
ALS	Amyotrophic Lateral Sclerosis
ALSFRS-R	ALS Functional Rating Scale (Revised)
AUC	Area Under the (ROC) Curve
CS	Consecutive Snapshots
CV	Cross-Validation
DT	Decision Tree
EHR	Electronic Health Record
FN	False Negatives
FP	False Positives
FVC	Forced Vital Capacity
LOO	Leave-One-Out
MAR	Missing at Random
MCAR	Missing Completely at Random
MEP	Maximal Expiratory Pressure
MIP	Maximal Inspiratory Pressure
MNAR	Missing Not at Random
NIV	Non-invasive Ventilation
RF	Random Forests
ROC	Receiver Operating Characteristic
RU	Random Undersampler
SMOTE	Synthetic Minority Over-sampling Technique
TN	True Negatives
TNR	True Negative Rate

TP	True Positives
TPR	True Positive Rate

1 Introduction

1.1 Context and Motivation

Considering a (real-valued, symbolic or heterogenous) three-dimensional dataset (three-way data), triclustering aims to discover subsets of objects, features and contexts (triclusters), satisfying certain homogeneity and statistical significance criteria. Given the increasing prevalence of three-way data across biomedical and social domains, triclustering — the discovery of coherent subspaces within three-way data — became a key technique to enhance the understanding of complex biological, individual, and societal systems [24]. Clustering is limited in this context, since objects in three-way data domains are typically only meaningfully correlated on subspaces of the overall space. Although biclustering is enables subspaces of both objects and features, context is disregarded. This thesis targets the use of triclustering for clinical data analysis. [35].

In this context, promising triclustering applications in clinical domains are multivariate physiological signal data analysis, where triclusters can capture coherent physiological responses for a group of individuals; neuroimaging data analysis, where triclusters can capture hemodynamic response functions and connectivity between brain regions; and clinical records analysis, where triclusters identify groups of patients with correlated clinical features along time [2, 27, 24].

Having labelled observations, triclustering can be applied to differentiate classes and support real-world decisions [31]. With this in mind, we propose a triclustering-based classifier to learn predictive models from three-way data, taking advantage of the temporal dependence between the features, and enhancing model interpretability by learning from local temporal patterns. In order to incorporate static

data, we further propose a prognostic prediction approach combining biclustering (with static data) and triclustering (temporal data), where both biclusters and triclusters are used as features in a supervised learning approach.

As case study, we use the analysis of three-way data from clinical records (patient-feature-time data). We target prognostic prediction in Amyotrophic Lateral Sclerosis (ALS) using a large cohort of Portuguese patients, where the triclusters learnt from patients' follow-up data can be interpreted as disease progression patterns.

ALS is a highly heterogeneous neurodegenerative disease characterized by a rapidly progressive muscular weakness. In general, patients with ALS generally die from respiratory failure within 3 to 5 years. However, some patients can live for less than one year, while others can live more than 10 years [22].

Worldwide, ALS affects between 5.9 and 39 people per 100.000 inhabitants [10]. In Portugal, 10 in 100.000 inhabitants suffer from this disease [11]. Most patients develop hypoventilation with hypoxemia and hypercapnia, requiring non-invasive ventilation (NIV) support [22]. In this context, foreseeing the beginning of hypoventilation is key to anticipate opportune interventions, such us the start of NIV. NIV was demonstrated to be effective in prolonging life and improving quality of life in ALS, in particular in patients without major bulbar muscles weakness [3, 5].

1.2 Goals and Contributions

This dissertation follows the work developed by Carreiro et al. [8] proposing the first prognostic models based on clinically defined time windows to predict the need for NIV in ALS. Following this work, Pires et al. [39] stratified patients according to their state of disease progression, and proposed specialized learning models based on three ALS progression groups (slow, neutral and fast). They further used patient and clinical profiles with promising results [40]. Nevertheless, neither the approach using patient stratification or using clinical and patient profiles for prognostic prediction, took into account the temporal dependence between the features. Matos et al. [37] used biclustering-based classification. Biclustering was used to find groups of patients with coherent values in subsets of clinical features (biclusters), then used as features together with static demographic data. The results were interesting but no temporal data were used.

In this context, the first goal of this thesis was to design and implement triclustering-based approaches to learn predictive models from temporal three-way data, enabling the learn of class-discriminative temporal patterns, and further enhancing model interpretability .

The second goal was to learn prognostic models using the proposed triclustering-based classifiers. As case study, we analyses three-way data from clinical records (patient-feature-time data) and target prognostic prediction in Amyotrophic Lateral Sclerosis (ALS) using a large cohort of Portuguese patients, where the triclusters learnt from patients' follow-up data can be interpreted as disease progression patterns. The goal is to predict whether an ALS patient will need NIV in the next 90 days (after a set of appointments). Furthermore, we aimed at using the model learned to obtain temporal disease progression patterns.

To achieve these goals, we designed and used TCtriCluster, a temporally constrained triclustering algorithm able to mine time-contiguous triclusters. TCtriCluster is an extension of TriCluster [47], a pioneer and highly cited triclustering algorithm, proposed by Zhao and Zaki to mine patterns in three-way gene expression data, extended to cope with three-way heterogeneous data and incorporate a temporal contiguity constraint.

In this scenario, the contributions of this thesis are the following:

- A preliminary study on the use of a triclustering-based classifier for prognostic prediction in ALS using triCluster. This work was published in proceedings of 14th International Conference on Practical Applications of Computational Biology and Bioinformatics (PACBB 2020) [43] (see Appendix A).
- TCtriCluster: an extension of triCluster adapted to mine time-contiguous triclusters and deal with possible missing values. We further applied TCtriCluster in prognostic prediction in ALS with improved performance and targeted model interpretability by analysing the disease progression patterns uncovered by triclustering (with and without patient stratification in Slow, Neutral and Fast Progressors [39]). This work originated a journal paper in Appendix B, currently under review in BMC Medical Informatics and Decision Making;
- BicTric: a new approach to learn predictive models combining biclustering and triclustering;
- Source Code in Python of triCluster and TCtriCluster [Git Repository].

1.3 The ALS Case Study

As mentioned above, worldwide, ALS affects between 5.9 and 39 people per 100.000 inhabitants [10]. In Portugal, 10 in 100.000 inhabitants suffer from this disease [11]. Most patients develop hypoventilation with hypoxemia and hypercapnia, requiring non-invasive ventilation (NIV) support [22]. In this context,

foreseeing the beginning of hypoventilation is key to anticipate opportune interventions, such us the start of NIV. NIV was demonstrated to be effective in prolonging life and improving quality of life in ALS, in particular in patients without major bulbar muscles weakness [3, 5].

With this in mind, in our case study we intend to build a prognostic prediction model to tackle the following questions:

- Given a set of consecutive patient appointments (T1, T2,..., T_k) can we predict if the patient will require NIV within a certain time window after evaluation T_k?
- Can we identify the most expressive combination of features that most describe the ALS patient disease progression profiles?

We use the Lisbon ALS clinic dataset containing Electronic Health Records from ALS Patients regularly followed at the local ALS clinic, since 1995 and last updated in March 2020. Its current version contains 1374 patients. Each patient has a set of static features (demographics, disease severity, co-morbidities, medication, genetic information, habits, trauma/surgery information and occupations) together with temporal features (collected repeatedly at follow-up), such as disease progression tests (ALSFRS-R scale, respiratory tests, etc) and clinical laboratory investigations.

Following previous work [39] and clinical feedback from the ALS experts we used the following features. From static data, we used Gender, Body Mass Index (BMI), MND familiar history, Age at onset, Disease duration, El Escorial reviewed criteria, UMN vs LMN, Onset form, C9orf72. From temporal data, we used 10 features per time point, the Functional Scores (ALSFRS-R), briefly described below, and respiratory tests: Forced Vital Capacity (FVC), Maximal Inspiratory Pressure (MIP) and Maximal Expiratory Pressure (MEP).

ALSFRS-R scores for disease progression rating are an aggregation of integers on a scale of 0 to 4 (where 0 is the worst and 4 is the best), providing different evaluations of the patient functional abilities at a given time point [14]. This functional evaluation is based on 13 questions, explained in Table 1.1. Different functional scores are then computed using subsets of scores, as shown in Table 1.2.

Table 1.1: ALSFRS-R Questions

Q1 - Speech
Q2 - Salivation
Q3 - Swallowing
Q4 - Handwriting
Q5 - Cutting food and Handling Utensils
Q6 - Dressing and Hygiene
Q7 - Turning bed ans adjusting bed clothes
Q8 - Walking
Q9 - Climbing Stairs
Q10 - Respiration
QR1 - Dyspnea
QR2 - Orthopnea
QR3 - Respiratory Insufficiency

1.4 Thesis Outline

This thesis is organized as follows:

Chapter 1 gives an introduction, explain the goals and contributions of this work and introduce the ALS case study.

Chapter 2 describes all the background concepts and definitions needed to better understand this dissertation.

Chapter 3 presents the proposed methodology to learn and use a triclustering-based classifier, together with the TCtriCluster algorithm.

Chapter 4 presents and discusses the results obtained in the ALS case study and shows the performed model interpretability analysis, pinpointing the patterns discovered and used by the predictive models.

Chapter 5 presents the proposed prognostic prediction approach combining biclustering and triclustering and discusses the results obtained in the same case study.

Chapter 6 presents the conclusions and future work.

Table 1.2: Functional Scores and Sub-scores according to ALSFRS-R.

Functional Score	**Description**
ALSFRS	sum of Q1 to Q10
ALSFRS-R	sum of Q1 to Q9 + QR1 + QR2 + QR3
ALSFRSb	Q1 + Q2 + Q3
ALSFRSsUL	Q4 + Q5 + Q6
ALSFRSsLL	Q7 + Q8 + Q9
ALSFRSr	Q10
R	QR1 + QR2 + QR3

2 Background and Related Work

This chapter discusses background and related work. We first introduce the necessary concepts on clinical and multiway data, data preprocessing and classification. We then provide key knowledge on clustering, biclustering and triclustering. Finally, we present related work on triclustering applications in biomedical data, (bi)clustering classification and other analyses using the ALS data used as case study.

2.1 Clinical Data and Multiway Data

In clinical domains data is in general composed of static and temporal data (as shown in Figure 5.1). When this is the case, static data corresponds to 2W dataset, while temporal data corresponds to a 3W dataset (composed of several 2W datasets). Two and three way datasets are formally defined as follows:

2W dataset A two-way dataset C (also refered as matrix, two-dimensional (2D), or 2W data in short) with n rows and m columns, is defined by n observations (rows) $X = \{x_1, \ldots, x_n\}$, m attributes (columns) $Y = \{y_1, \ldots, y_m\}$, and $n \times m$ elements (values) a_{ij}. Elements a_{ij} relate observation x_i and attribute y_j. 2W data can be real-valued ($a_{ij} \in \mathbb{R}$), symbolic ($a_{ij} \in \Sigma$, where Σ is a set of nominal or ordinal symbols) or integer ($a_{ij} \in \mathbb{Z}$) [35, 24].

3W dataset A three-way dataset A (also referred as tridiac data, cube data, three-dimensional (3D) or 3W data in short) is defined by n observations $X = \{x_1, \ldots, x_n\}$, m attributes $Y = \{y_1, \ldots, y_m\}$, and p contexts $Z = \{z_1, \ldots, z_p\}$. Elements a_{ijk} relate observation x_i , attribute y_j , and context z_k. 3W data can be real-valued ($a_{ijk} \in \mathbb{R}$), symbolic ($a_{ijk} \in \Sigma$, where Σ is a set of nominal or ordinal symbols) or

integer ($a_{ijk} \in \mathbb{Z}$). [24]

2.2 Data Preprocessing

Real-world data tend to be incomplete, noisy and inconsistent [20]. Since we intend to keep all data clean (or as clean as possible), it is necessary to adapt all data analysis processes to meet this requirement, providing them with the most appropriate techniques for that purpose. Dealing with missing and imbalanced data are important tasks in this dissertation context.

2.2.1 Missing Data

It is rare for a research investigation not to have missing data [38]. However, ignoring their existence will make the analysis more difficult and biased. Strategies for handling missing values are based on different assumptions and have different limitations. Key questions to consider when selecting a method for handling missing values include: 1) Why are data missing?; 2) How do records with missing and complete data differ?; and 3) Do the observed data help predict the missing values? [38].

We can consider three types of missing data: Missing Completely at Random (MCAR) , Missing at Random (MAR) and Missing Not at Random (MNAR) [32], described as follows:

- **MCAR:** when the probability of a missing value is independent of the feature itself and any external influences (e.g. lost information caused by system crash, human error);
- **MAR:** when the probability of a missing value is still independent of the feature itself but not of external influences, with the missing values having a predictable pattern (e.g. sensor fail);
- **MNAR:** when the probability of a missing value is dependent of the feature itself (e.g. if a sensor can not acquire information outside a certain range).

To deal with this issue, there are three main types of approaches that can be applied according to the type of missing value: Deletion, Imputation or Prediction [15], characterized as follows. In practice, we usually use a combination of these approaches.

Deletion methods consist in removing either instances or columns mostly composed by missing values

(according to a user-defined threshold). However, in datasets with low quantities of data or with many missing data this can lead to a considerable loss of information ;

Imputation methods can be divided into two groups: Single Imputation (SI) or Multiple Imputation (MI). Single imputation methods replace the missing value with plausible values by observing the characteristics of the population. The most common method is Mean Imputation which imputes missing values using the population mean for the variable. However, when performed in datasets with large quantities of missing data, it leads to a loss of feature variance and correlation distortion that can lead to a biased dataset [42]. Last Observation Carried Forward (LOCF) presents as an alternative to Mean Imputation, by assuming that the value does not change from the last observation [38]. This methodology is especially common in clinical datasets, where longitudinal data is available. [42];

Prediction methods are the most recently approaches and consists in using predictive models learned from the remaining data to predict missing values [12, 16].

According to the type of data analysis task and the used algorithms it can be useful to keep missing values, since their replacement can also introduce bias in data.

2.2.2 Data Imbalance

An imbalanced dataset is a dataset containing more labeled instances as from one of the classes values than the others, meaning that one of the classes is more represented within data. This contrast will have a lot of influence on many machine learning problems, more specifically on classification problems.

Since overcoming this challenge improves the performance and reliability of the results, there are two techniques that are being used widely to deal with it: Oversampling and Undersampling. The first one consists in adding "artificial" instances of the minority class(es) and the second in removing instances from the majority class, ensuring the balance of data [20].

There are several different approaches considered for the both techniques. Concerning Undersampling the most common is Random Undersampling, which consists in randomly remove instances from the majority class until obtain the same proportions of classes in the dataset [20]. This approach faces some problems, such as the loss of information by removing instances that would be informative [21]. On the other hand, with Undersampling techniques this problem does not appear, since we can just randomly add copies of instances of the minority class to the dataset until we get equal instance numbers for both classes. However this approach easily can lead to a biased dataset [21].

Synthetic Minority Over-sampling Technique (SMOTE) proposes an oversampling alternative, as it creates synthetic learning instances for the minority class by using k-Nearest Neighbors method to find similar k instances to one of the minority class examples, and use them to create a new instance [9]. Combining the two approaches, could be the solution to overcome some of the problems of both, when considering binary class dataset [28]. However, since with a multiclass dataset this problem is not easily overcome many authors tried to obtain an effective solution [20, 48, 41].

In practice when the imbalance is huge and enough data is available the best can be to first do under-sampling and then use SMOTE [37, 39, 6].

2.3 Classification

In Machine Learning, Classification, which can be defined as the process of finding a model that describes and distinguishes data classes or concepts, is a Supervised Learning task assuming the existence of labelled from which the classifier should be learnt. [20].

Classification Task is a two-step process. First, in the learning step, the classification model is built and latter the learned model is used to predict class label for a new given data, in classification step [20]. More concretely in the first step, a classification algorithm builds the model by examining a training set composed by observations and their associated class labels. An observation or tuple, X is a n-dimensional feature vector, $X = (x_1, x_2, ..., x_n)$, where it is assumed that X belongs to a predefined class. In the context of classification, tuples can be referred as samples, examples, instances, data points or objects [20].

2.3.1 Decision Trees

Decision Trees (DTs) are widely used models in Classification due to being easy to visualize, understand and interpret. Most algorithms used to learn a DT from training data (e.g. ID3, C4.5 and CART) employ a greedy top-down recursive divide-and-conquer strategy: the data is recursively divided into smaller partitions by selecting in each iteration the feature which separates best the data entries into their respective classes (the splitting attribute), by using an attribute selection measure (e.g. Information Gain, Gain Ratio or Gini Index) [20].

Figure 2.1 depicts an example of DT to predict whether a patient has flu. The top of the DT is the feature which better separates data, according to the class values of each instance. The next internal node

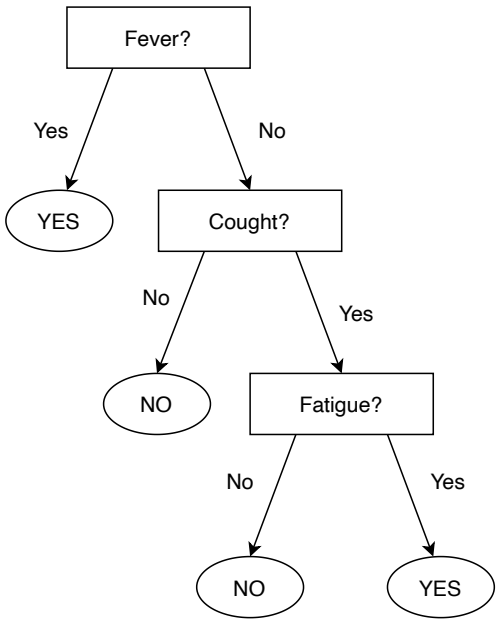


Figure 2.1: Example of Decision Tree for problem: ”Have the patient flu?”

for each branch will be the best describing feature for the data subset that follows each branch. This last step is repeated until either all features have been used, or all the final branches lead to a prediction. When we reach a feature in which each value leads to a single prediction, then there is no need to look further in the remaining features and the branches of that feature will lead to the leaves (final predictions) of the tree. In situations where even with all available features, there is no combination that allows reaching single predictions, we have often resort to majority voting. This consists of choosing the prediction value for each branch according to the most popular value [20].

2.3.2 Ensemble Classifiers and Random Forests

An ensemble method combines a set of k learned models (base classifiers), $M_1, M_2, ..., M_k$ with the aim of creating an improved composite classification model. A given dataset, D, is partitioning into k training sets, D_1, D_2, D_k, where D_i $(1 \leq i \leq k-1)$ is used to learn M_i. To classify a new tuple, the ensemble will return a class prediction based on the predicted classes (votes) by the base classifiers. Due to it's composition by several classifiers, ensemble tends to be more accurate than the separated classifiers [20].

Random Forests (RF) is an ensemble where the base classifiers are Decision Trees, composing a “forest”. The individual DTs are generated using a random selection of attributes at each node to determine the split. To make a prediction, each tree votes and the most popular class is the predicted class by RF.

Random Forests are more robust to errors and able to compensate for the overfitting individual DTs tend to suffer (as long as the number of trees is large) and are capable of returning internal estimates of Feature Importance [20].

2.3.3 Performance Evaluation

The building of a classification model requires a proper evaluation in order to measure if the classifier works like we expect. Considering Classification with Binary class problems (having a two class dataset) can be considered two types of tuples: Positive tuples, tuples of the main class of interest, and Negative tuples, the remaining tuples [20]. There are four additional terms useful to understand some metrics:

True positives (TP) represents the positive tuples correctly labeled by the classifier;

True negatives (TN) represents the negative tuples correctly labeled by the classifier;

False positives (FP) represents the negative tuples that were incorrectly labeled as positive by the classifier;

False negatives (FN) represents the positive tuples that were mislabeled as negative [20].

This four concepts can be summarized in a table called Confusion Matrix, represented in Figure 2.2.

With the concepts represented in the confusion matrix, it is possible to compute some metrics that are summarized in Table 2.1 . This metrics are used to evaluate the performance of classification.

Accuracy is the most frequently metric used to evaluate the classifier performance. Despite its good representative character it is necessary to take into account class imbalance problem. If the dataset distribution reflects a significant majority of one of the classes, accuracy tends to provide inconclusive evaluations, since the classifier could be correctly labeling only tuples that belong to one of the classes and misclassifying all the others. With this, it is important to couple accuracy with other metrics like Specificity and Sensitivity since this metrics can assess how well the classifier recognize positive and negative tuples [20].

As aforementioned, to evaluate a classifier using the metrics described above, need careful and take into account the proportions of each class value as to not be biased by the results. The ROC curve combines the Sensitivity and Specificity metrics in a graph displaying the trade-off between the rate at which a model can classify correctly Positive tuples (Sensitivity) versus the rate at which it misclassifies Negative tuples (Specificity) for different portions of the Test set.

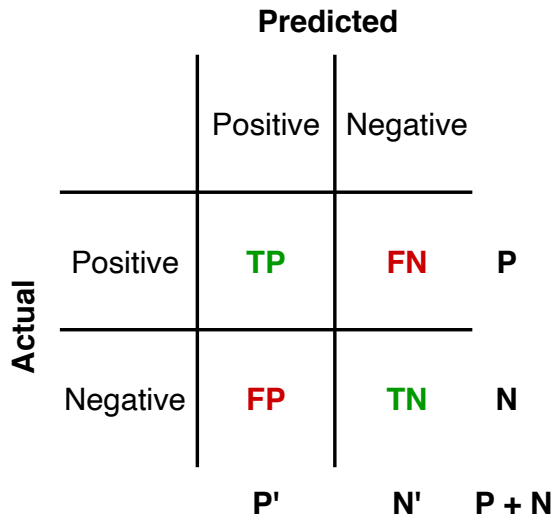


Figure 2.2: Confusion Matrix. TP, TN, FP, FN, P, N, P', N' refer to the number of true positives, true negatives, false positives, false negatives, actual positive, actual negative, predicted positive and predicted negative tuples, respectively.

The ROC is also used to compute one of the most popular metrics in performance evaluation, the Area Under the ROC Curve, also known as AUC. As it says in the name, this metric measures the area under the ROC curve. The AUC metric can be defined as either the representation of the classifier ability to separate the classes or the probability of an instance with a given class value being classified as such [20]. Figure 2.3 depicts an example of a ROC curve with AUC highlighted.

K-fold Cross-Validation (CV) is a popular method for performance evaluation in Classification. This method consists in split the data in a number of subsets in order to evaluate the performance of classifier using one subset as test set and the remaining as training set. Concretely, in k-fold CV the original dataset is randomly partitioned into k mutually exclusive subsets or "folds", each of approximately equal size. The process of training and testing is performed k times, once per iteration. In iteration i, partition D_i is used as the Test set, and the remaining partitions are collectively used to train the model. The final evaluation metrics are calculated by performing the mean of each metric between all folds [20]. There are some approaches to be considered according to the type of problem, such as:

Leave-One-Out (LOO) When k is set to the number of tuples in the original dataset, leaving only one tuple per iteration on the Test set;

Stratified Cross-Validation When the "folds" are stratified so that the class distribution of the tuples in each fold is approximately the same as that in the initial data

Table 2.1: Performance Evaluation Metrics.

Metric	Description	Formula
Accuracy	Represents the percentage of tuples correctly classified.	$\frac{TP+TN}{P+N}$
Sensitivity, Recall or True Positive Rate (TPR)	Shows the proportion of positive tuples that were correctly classified.	$\frac{TP}{P}$
Specificity or True Negative Rate (TNR)	Shows the proportion of negative tuples that were correctly classified.	$\frac{TN}{N}$
Precision	Represents the percentage of tuples correctly classified as positive.	$\frac{TP}{P'}$
F-measure	Harmonic mean of Precision and Recall.	$\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$

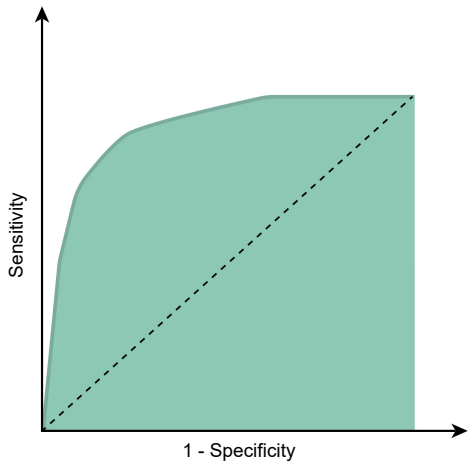


Figure 2.3: A ROC example

As described in [20] the most recommended approach is stratified 10-fold CV for estimating accuracy due to its relatively low bias and variance.

2.3.4 Feature Importance

In life sciences, interpretability of machine learning models is as important as their prediction accuracy [1]. With this in mind, becomes important to understand how the predictive features have an influence on training a classification model to predict some interesting target variable. It is then necessary to evaluate the importance that a given feature has in the predictive model in order to withdraw some assumptions of the reason for the influence. The more a feature is used to better split the data space, the higher its relative importance will be [1, 34]

2.4 Clustering and Biclustering

2.4.1 Clustering

Given a 2W dataset with n observations, $X = \{x_1, \ldots, x_n\}$, described by m features, Y, the clustering task aims to find subsets of observations (clusters), $\{I_1, \ldots, I_r\}$, where $I_i \subseteq X$ satisfies certain intracluster and intercluster criteria of (dis)similarity over the whole space [24].

Despite the relevance of the clustering task the (dis)similarity between observation becomes biased considering an high number of attributes per observation [24]. A way to tackle this problem is to perform clustering in data subspaces so that a group of observations needs only to be similar on a subset of attributes. Biclustering targets this problem [24].

2.4.2 Biclustering

Bicluster Given a 2W dataset (matrix), M, with n observations X and m features Y, a real-valued or symbolic matrix **bicluster** $B = (I, J)$ is a subspace given by a subset of observations, $I \subseteq X$ and subset of features, $J \subseteq Y$ [35, 24].

Biclustering Solution Given M, the biclustering task aims to find a set of biclusters $\{B_1, \ldots, B_q\}$, such that each bicluster B_i satisfies specific criteria of homogeneity and statistical significance [35, 24].

As we will see in what follows, by adding a third dimension to 2W data, similar assumptions can be done in order to get triclusters.

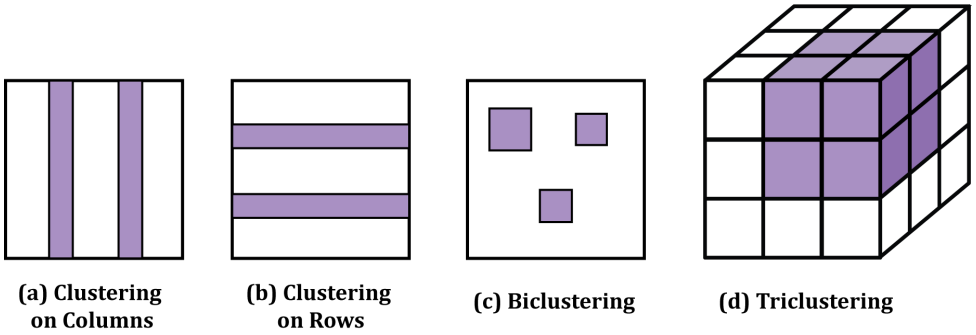


Figure 2.4: Comparison between Clustering, Biclustering and Triclustering

2.5 Triclustering

Tricluster Given a 3W dataset C with n observations X, m attributes Y, and p contexts Z, a **tricluster** $T = (I, J, K)$ is a subspace of the original space, where $I \subseteq X$, $J \subseteq Y$, and $K \subseteq Z$ are subsets of observations, features, and contexts, respectively [24].

Triclustering Solution Given C, the triclustering task aims to find a set of triclusters $\{T_1, \ldots, T_l\}$ such that each tricluster T_i satisfies specific criteria of homogeneity and statistical significance [24]. Figure 2.5 depicts the concepts used in 3W data analysis.

The **homogeneity** criterion determines the structure, coherence, and quality of a triclustering solution, where:

- the *structure* is described by the number, size, shape, and position of triclusters;
- the *coherence* of a tricluster is defined by the observed correlation of values (coherence assumption) and the allowed deviation from expectations (coherence strength); and
- the *quality* of a tricluster is defined by the type and amount of tolerated noise.

2.5.1 Tricluster Coherence

The **coherence assumption** defines the type of correlation between the values of a tricluster, and it can be:

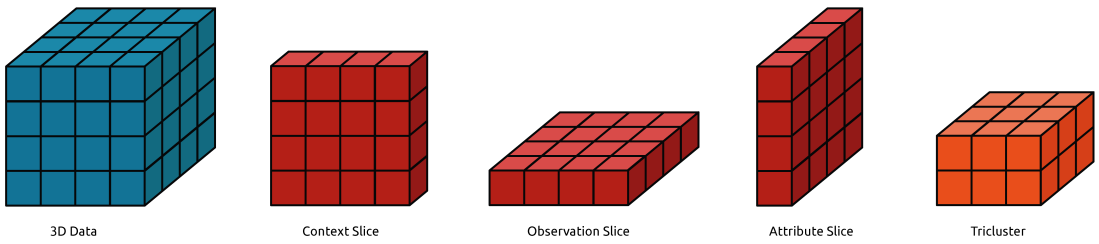


Figure 2.5: Concepts of 3W data analysis

1. **Cubic**, when established among all values in tricluster
2. **Intraplane**, if established for each slice of a tricluster
3. **Interplane** when established between the slices of a tricluster

Considering **Cubic coherence**, let $T = (I, J, K)$ be a tricluster with a_{ijk} values. If T has only categorical values and $a_{ijk} \in \Sigma$ we said that T corresponds to a tricluster with:

Constant symbol if $a_{ijk} = c$

Constant pattern if $a_{ijk} = c_j$ (or c_i or c_k)

Considering that T has only real values, $a_{ijk} \in \mathbb{R}$, whose values respecting $a_{ijk} = c + \alpha_i + \beta_j + \gamma_k + \eta_{ijk}$ (where $c \in \mathbb{R}$ and α_i, β_j and $\gamma_k \in \mathbb{R}$ are contributions from x_i observation, y_j attribute, and z_k context) it is said that B follows a:

Fully additive assumption when $\alpha_i \neq 0, \beta_j \neq 0$ and $\gamma_k \neq 0$; or

Partially additive assumption when $\alpha_i = 0, \beta_j = 0$ or $\gamma_k = 0$.

When the values of the tricluster are better described by $a_{ijk} = c \times \alpha_i \times \beta_j \times \gamma_k + \eta_{ijk}$ is said that it follows a:

Fully multiplicative assumption when $\alpha_i \neq 0, \beta_j \neq 0$ and $\gamma_k \neq 0$; or

Partially multiplicative assumption when $\alpha_i = 0, \beta_j = 0$ or $\gamma_k = 0$.

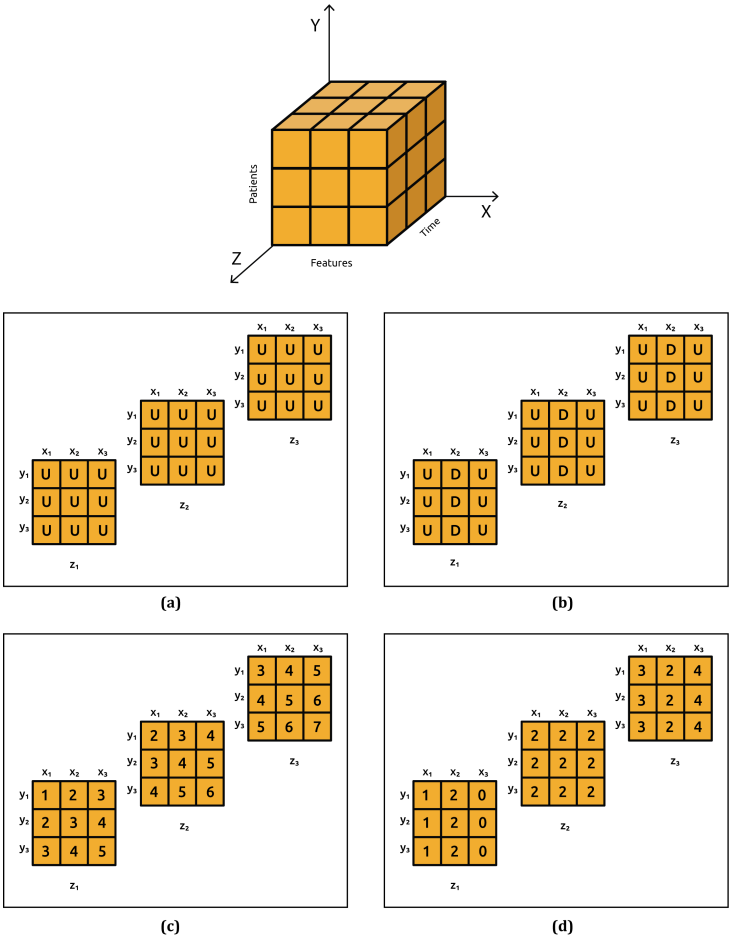


Figure 2.6: Tricluster Types according Cubic Coherence: (a) Constant symbol, (b) Constant pattern, (c) Fully additive, (d) Partially additive.

A tricluster considering the cumulative effects from other triclusters follows a **plaid** asumption and is defined as:

$$a_{ijk} = \mu_0 + \sum_{t=0}^{q} \theta_{ijkt} \rho_{it} \kappa_{jt} \tau_{kt} \tag{2.1}$$

where θ_{ijkt} defines the contribution from the tricluster $\mathbf{T}_t = (\mathbf{I}_t, \mathbf{J}_t, \mathbf{K}_t)$ to a_{ijk} when ρ_{it}, κ_{jt} and τ_{kt} are true, i.e., $\mathbf{x}_i \in \mathbf{I}_t,\ \mathbf{y}_j \in \mathbf{J}_t$ and $\mathbf{z}_k \in \mathbf{K}_t$.

Regarding the **intraplane coherence** the majority of models are adapted from Biclustering, since a slice of a Tricluster can be seen a matrix with 2 dimensions [24].

Figure 2.6 shows four examples of values for a tricluster with different types of cubic coherence.

2.5.2 Temporality of 3W Data

When contexts correspond to time points, we are in the presence of a **temporal 3W dataset** (also referred as three-way time series or temporal 3D dataset), where each observation is a multivariate time series with m order [24].

The context of 3W data is very extensive and plentiful. It is then necessary to take into account the various data-related specificities with impact on the triclustering task [24]. In the context of this work we will explore in a better way the temporality of data.

The vast majority of available 3W data comes from periodic observations on biological, individual and societal systems. Henriques and Madeira [24] identified some unique challenges of 3W temporal data, such as (1) place adequate homogeneity criteria to capture meaningful forms of temporal progression, (2) handle arbitrarily high temporal lags on observations, (3) place proper contiguity criteria, and (4) deal with the complex stochasticity inherent to temporal 3W data.

2.6 Related Work

This section presents related work on the concepts covered in this thesis. We first discuss relevant state-of-the-art triclustering algorithms and their applications. Finally, we present studies performed with the same data.

2.6.1 Triclustering Algorithms and Applications

The first steps in triclustering research were taken by Zhao and Zaki. They conceived triCluster algorithm to extract patterns in 3D gene expression data. This algorithm mines arbitrarily positioned and overlapping scaling and shifting patterns [46]. One year latter, Jiang et al. [25] proposed an extended and generalized version of the TriCluster algorithm, named gTriCluster that improved this proposal considering inter temporal coherence while generating triclusters. Many others researches have been done about Triclustering [18, 17, 19, 45, 33, 29, 4, 44, 30, 2, 26] . These will be briefly presented in Section 2.6.2.

In order to categorize the different algorithms according to the aspects of the triclustering tasks (diversity of data inputs, behavioral options, desirable homogeneity criteria), Henriques and Madeira [24] proposed to divide the algorithms relative to (1) whether their behaviour is based on iterative searches (**greedy**) or on distribution parameter identification (**stocastic**) and (2) whether they are able to offer guar-

antees of optimality (**exhaustive**) or not. Each approach can be further categorized according to whether the behaviour relies on biclustering algorithms, pattern mining procedures or evolutionary multiobjective optimization, among other possibilities.

Triclustering algorithms can be applied to different types of data, from biological and medical to social domains. Specifically this work will focus on biological and clinical data analysis. The main applications for biological data are: analysis of 3D omic, augmented, network, multispecious, chemical, drug and sample-by-sample data [24]. In clinical domains there are three major triclustering applications, according Henriques and Madeira [24]: (1) multivariate physiological signal (*individual-feature-time signal data*) analysis where triclusters can capture coherent physiological responses for a group of individuals; (2) neuroimaging data analysis, where triclusters can capture hemodynamic response functions and connectivity between brain regions; and (3) clinical records analysis, where triclusters correspond to groups of patients with correlated clinical features along time.

2.6.2 Triclustering Applications in Biomedical Data

This section presents the various applications and experiments made with the different types of 3D data, biological and medical, whose goals are typically:

1. Study response of patients to specific conditions/ treatment;
2. Evaluate disease progression;
3. Identify biological patterns/modules

Bhar et al. [4] proposed in 2013, a novel algorithm - δ-TRIMAX which was used to mine 3D gene expression datasets by introducing a 3D mean square residue (MSR). This new proposal extracts perfectly shifting triclusters with hub genes from breast cancer cells. Rubio-Escudero and Gutiérrez-Avilés [18] triclustered human genes (GDS 4472), using a multislope measure (MSL) to compute the similarity between triclusters by averaging differences on the angles of the plane slopes for all pairs of observations, attributes and contexts in a given subspace.

Amar et al. [2] proposed an algorithm for finding coherent and flexible modules in three-way data, based on hierarchical Bayesian data model and Gibbs sampling. This algorithm was applied to data from time series measurements of gene expression about humans septic shock response and brain functional magnetic resonance imaging (fmri) time series.

Li et al. [29] used TRI-Clustering with Yeast cell cycle dataset to explore regulated expression values

and mine time-delayed gene expression patterns from microarray data and conclude that this model can be extended to 3D gene × sample × time datasets to identify 3D td-clusters. MultiFacTV algorithm was proposed to extract modules that are composed of some genes, conditions and time-points based on tensor factorization objective [30].

TriGen, implemented by Gutiérrez-Avilés and Rubio-Escudero, is a triclustering-genetic algorithm based on an evolutionary heuristic, genetic algorithms, which finds patterns of similarity for genes on a three dimensional space, thus taking into account the gene, conditions and time factors. They experimented this algorithm together with 2 evaluation measures: LSL [17] and MSL [18].

More recently, Kakati et al. [26] developed an algorithm with the intent of solving the underlying problem of most of triclustering algorithms, they are not able to handle co-occurring shifting-and-scaling patterns. THD-Tricluster, identify triclusters over the Gene × Sample × Time (GST) domain and was applied on HIV-1 progression data to identifies disease-specific genes.

Table 2.2 summarize these algorithms and gives a brief explanation of their algorithmic approaches and datasets used for validation.

2.6.3 (Bi)clustering-based Classification

Considering clustering-based classification, we can consider a set of found clusters by any clustering algorithm, as a class-discriminative features, taking advantage of the fact that the objects are grouped and that there is a clear similarity between them to evidence the separation of the data. If we consider Biclustering instead of just Clustering given the relationship between features within the biclusters, we can obtain better results, taking advantage of the subset of features' space of the pattern from each Bicluster [6, 7].

The most simple approach to use Biclustering-based classification is to build a matrix (observations × biclusters) considering biclusters as features and assign the observations to biclusters using binary labels (1 if the observation is contained in bicluster; 0 otherwise). This matrix will be used next by the classifier to make predictions based on bicluster composition patterns. Many variants of this simple approach can be found in [6, 7].

Matos et al. [37] used biclustering-based classification, using the pattern-based biclustering algorithm BICPAM [23] and Random Forests, to extract discriminative meta-features. These so called meta-features corresponded to the patterns of discriminative biclusters.

Table 2.2: State-of-the-art triclustering algorithms for biomedical data analysis. The goals are: (1) Study response of patients to specific conditions/treatment; (2) Evaluate disease progression and (3) Identify biological patterns/modules.

Algorithm		**Year**	**Data set**	**Algorithmic Approach**	**Goals**
triCluster	[46]	2005	Yeast cell cycle	Biclustering based (quasi-exhaustive) with graph consensus	(3)
gTRICLUSTER	[25]	2006	Yeast cell cycle regulated	Biclustering based (quasi-exhaustive) with graph consensus	(3)
MOGA3C	[33]	2007	Yeast cell cycle regulated	Multiobjective optimization (quasi-exhaustive)	(3)
TRI-Clustering	[29]	2009	Yeast genome	Greedy (divide-and-conquer)	(3)
TD-Clustering	[45]	2010	Yeast gene expression	Pattern based (quasi-exhaustive)	(3)
δ - TRIMAX	[4]	2012	Human estrogen induced breast cancer cell	Greedy (Divide-and-conquer)	(3)
OPTricluster	[44]	2012	Mice, *Arabidopsis thaliana*, *Brassica napu*	Biclustering-based (greedy)	(3)
MultiFacTV	[30]	2013	*Arabidopsis thaliana* and Yeast cell cycle	Stochastic (tensor factorization objective)	(3)
LSL-TriGen	[17]	2014	Yeast cell cycle and human inflammation and host response to injury	Multiobjective optimization (quasi-exhaustive)	(3)
MSL-TriGen	[18]	2015	Yeast cell cycle, Mice (GDS4510) and Humans (GDS4472)	Multiobjective optimization (quasi-exhaustive)	(3)
TWIGS	[2]	2015	Microarray dataset for transcriptional response of patients to sepsis and fMRI data	Stochastic (hierarchical Bayesian model)	(1), (3)
THD - Tricluster	[26]	2018	Yeast cell cycle and HIV-1 disease progression	Biclustering based	(2)

2.6.4 Previous Work with similar ALS Patients Data

Carreiro et al. [8] proposed the first prognostic models based on clinically defined time windows to predict the need for NIV in ALS. Following this work, Pires et al. [39] stratified patients according to their state of disease progression, and proposed specialized learning models based on three ALS progression groups (slow, neutral and fast). They further used patient and clinical profiles with promising results [40]. Nevertheless, neither the approach using patient stratification or using clinical and patient profiles for prognostic prediction, took into account the temporal dependence between the features. Matos et al. [37] used biclustering-based classification. Biclustering was used to find groups of patients with coherent values in subsets of clinical features (biclusters), then used as features together with static demographic data. The results were interesting but no temporal data were used.

3 Triclustering-based Classification

Considering a temporal and heterogeneous three-way dataset, composed by a set of observations, features and time-points as shown in Figure 3.1, we aim to learn a classification model based on triclustering not only to effectively classify pre-labelled observations but also to unravel the most important temporal patterns used for classification in order to promote model explainability.

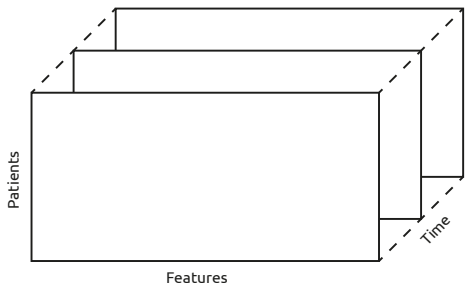


Figure 3.1: Example of temporal and heterogeneous three-way data: Electronic Health Records with Patients, Features and Time as dimensions

In this context, we propose a new triclustering-based classification, where triclusters are first discovered and then used as features to construct a predictive model and identify the most important temporal patterns to be analyzed. Figure 3.2 depicts the overall workflow.

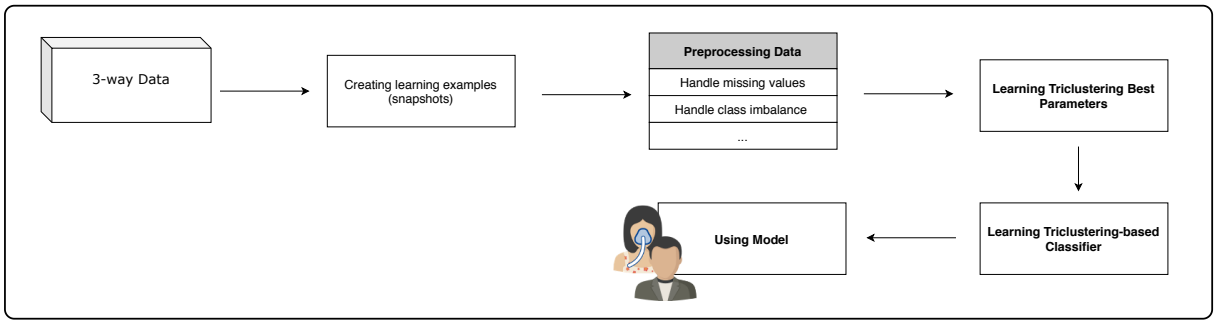


Figure 3.2: Proposed Workflow to Learn a Triclustering-based Classifier

This chapter describes the proposed methodology to learn a triclustering-based classifier from three-way data, from preprocessing (including creating learning examples) to classifier performance evaluation, exploring several approaches in each step. We further describe TCtriCluster, the proposed triclustering algorithm to mine temporally constrained triclusters.

In what follows, consider that a three-way dataset, D, is defined by n objects $X = \{x_1, ..., x_n\}$, m features $Y = \{y_1, ..., y_m\}$, and p contexts $Z = \{z_1, ..., z_p\}$, where the elements d_{ijk} relate object x_i, feature y_j, and context z_k. Consider also that, a bicluster $B = (I, J)$ is a subspace given by a subset of objects, $I \subseteq X$, and a subset of features, $J \subseteq Y$ [35]. Similarly, a tricluster $\mathcal{T} = (I, J, Z)$, contains $I \subseteq X$ objects, $J \subseteq Y$ features and $K \subseteq Z$ contexts, and t_{ijk} denote the elements of $\mathcal{T}$, where $1 \leq i \leq I$, $1 \leq j \leq J$ and $1 \leq k \leq K$ [24]. In this context, each tricluster $\mathcal{T}$ can be represented as a set of K biclusters $\mathcal{T} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_K\}$:

$$\mathcal{B}_1 = \begin{bmatrix} t_{111} & t_{121} & \cdots & t_{1J1} \\ t_{211} & t_{221} & \cdots & t_{2J1} \\ \vdots & \vdots & \ddots & \vdots \\ t_{I11} & t_{I21} & \cdots & t_{IJ1} \end{bmatrix}$$

$$\mathcal{B}_2 = \begin{bmatrix} t_{112} & t_{122} & \cdots & t_{1J2} \\ t_{212} & t_{222} & \cdots & t_{2J2} \\ \vdots & \vdots & \ddots & \vdots \\ t_{I12} & t_{I22} & \cdots & t_{IJ2} \end{bmatrix}$$

$$\vdots$$

$$\mathcal{B}_K = \begin{bmatrix} t_{11K} & t_{12K} & \cdots & t_{1JK} \\ t_{21K} & t_{12K} & \cdots & t_{1JK} \\ \vdots & \vdots & \ddots & \vdots \\ t_{I1K} & t_{I2K} & \cdots & t_{IJK} \end{bmatrix}$$

3.1 Creating Learning Examples and Preprocessing Data

Three-way data can have different formats. As such, and in order to apply the proposed approach, data needs to be preprocessed in order to create learning examples (snapshots) composed of sets of features from consecutive time-points. Our approach follows the work of Carreiro et al. [6] and Pires et al. [39]. Depending on the dataset, dealing with missing values and class imbalance, might also be needed.

3.2 TCtriCluster: A new Temporal Triclustering Algorithm

For these methodology, we considered TriCluster [47], the pioneer and highly cited triclustering approach, proposed and implemented by Zhao and Zaki in 2005. It is a quasi-exhaustive approach, able to mine arbitrarily positioned and overlapping triclusters with constant, scaling, and shifting patterns from three-way data. Given that TriCluster was proposed to mine coherent triclusters in three-way gene expression data (gene-sample-time), at this point it is important to understand that clinical data can be preprocessed in order to have a similar structure, in which gene-sample-time data becomes patient-feature-time data, for instance. TriCluster has 3 main steps: 1) construct a multigraph with similar value ranges between all pairs of samples; 2) mine maximal biclusters from the multigraph formed for each time point (slices of the 3D dataset); and 3) extract triclusters by merging similar biclusters from different time-points. Optionally, it can delete or merge triclusters, according to the overlapping criteria used.

However, since our goal is to mine temporal three-way data, meaning the Z dimension (contexts) is time, we borrowed the idea in CCC-Biclustering [36], a state of the art and highly efficient temporal biclustering algorithm, and introduced a temporal constraint in triclustering. The goal thus became to

mine Time-Contiguous Triclusters (TCTriclusters), triclusters with consecutive time-points. In this context, we re-implemented TriCluster in Python and extended it to cope with a time constrain. The new TCtriCluster algorithm implements this time constrain on its 3rd phase, as shown in Algorithm 1 (line 9).

Allowing missing values directly in triclustering, thus preventing previous imputation, can improve the interpretability of obtained triclusters and reduce the noise imposed by imputation with other artificial values. With this in mind, we decided to introduce a step in the algorithm (2nd step - biCluster) to handle missing values. This step evaluates the missing values with respect to each observation and compares the similarity between these observations (with missing values) with the observations composing the obtained biclusters. If the similarity value is within the limits dictated by the ratio threshold, the observation is added to the bicluster if the proportion between the number of features with missing values and the number of features in bicluster is less than missing threshold.

TCtriCluster allows different combinations of input parameters, that should be explored in order to discover the best parameters, with which the final classifier should be learnt. The input parameters are: $\varepsilon, mx, my, mz, \delta^x, \delta^y, \delta^z, \eta$ and γ, corresponding to maximum ratio value, minimum size of tricluster dimensions x, y and z, maximum range threshold along dimensions x, y and z, overlapping and merging threshold, respectively.

3.3 Learning Triclustering Best Parameters

In this step, the goal is to compute the best parameters to be used as input by the triclustering algorithm (we used both triCluster and the proposed temporal triclustering algorithm TCtriCluster) in order to obtain the best classification performance. The workflow, depicted in Figure 3.3, starts by performing triclustering on the preprocessed data to obtain triclusters. Next, the virtual pattern 3D is computed for each tricluster. The proposed virtual pattern 3D, extended from the 2D version defined in [13], is computed as follows, using the virtual pattern of each bicluster composing the tricluster.

Definition 1. (Virtual Pattern 3D). Given a tricluster $\mathcal{T}$, its virtual pattern $\mathcal{P}$ is defined as a set of elements $\mathcal{P} = \{\rho_1, \rho_2, ..., \rho_I\}$, where $\rho_i, 1 \leq i \leq I$ is defined as the mean (or the mode, in case of categorical features) of values in the i^{th} row for each context:

$$\rho_i = \frac{1}{J \cdot Z} \sum_{z=1}^{Z} \sum_{j=1}^{J} b_{ijz} \tag{3.1}$$

To assess how well a specific object (patient) follows the general tendency of a given tricluster $\mathcal{T}$, we proposed three approaches: (1) Binary - consider that if an observation p_i is contained in $\mathcal{T}$ it follows

Algorithm 1: TCtriCluster: Extension of triCluster able to mine TCTriclusters

Input: $\varepsilon, mx, my, mz, \delta^x, \delta^y, \delta^z$, bicluster sets $\{\mathcal{C}^t\}$ of all contexts (time-points), set of objects X, features Y and contexts (time-points) Z

Output: cluster set $\mathcal{C}$

Initialisation: $\mathcal{C} = \varnothing$, call TCtriCluster($\mathcal{T} = X \times Y \times \varnothing$, Z)
TCtriCluster ($\mathcal{T} = I \times J \times K$, U)
if $\mathcal{T}$ *satisfies* $\delta^x, \delta^y, \delta^z$ **then**
    if $|\mathcal{T}.K| \geq mz$ **then**
        if $\mathcal{T} \not\subset \mathcal{T}' \in \mathcal{C}$ **then**
            Delete any $\mathcal{T}'' \in \mathcal{C}$, if $\mathcal{T}'' \subset \mathcal{T}$
            Add $\mathcal{T}$ to $\mathcal{C}$
foreach $t_i \in U$ **do**
    if $(\mathcal{T}^{new} = \varnothing) \vee (\mathcal{T}.K_{-1} + 1 = t_i)$ **then**
        $\mathcal{T}^{new}.K \longleftarrow \mathcal{T}.K + t_i$
        Remove t_i from U
        forall $t_k \in \mathcal{T}.K$ *and each bicluster* $b_i^{t_k} \in \mathcal{C}^{t_k}$, *such that* $|b_i^{t_k}.I \cap \mathcal{T}.I| \geq mx$ *and* $|b_i^{t_k}.J \cap \mathcal{T}.J| \geq my$ **do**
            $\mathcal{T}^{new}.I \longleftarrow b_i^{t_k}.I \cap \mathcal{T}.I$
            $\mathcal{T}^{new}.J \longleftarrow b_i^{t_k}.J \cap \mathcal{T}.J$
            if $|\mathcal{T}^{new}.I| \geq mx$ *and* $|\mathcal{T}^{new}.J| \geq my$ *and the ratios at time* t_i, t_k *are coherent* **then**
                TCTriCluster ($\mathcal{T}^{new}, P$)

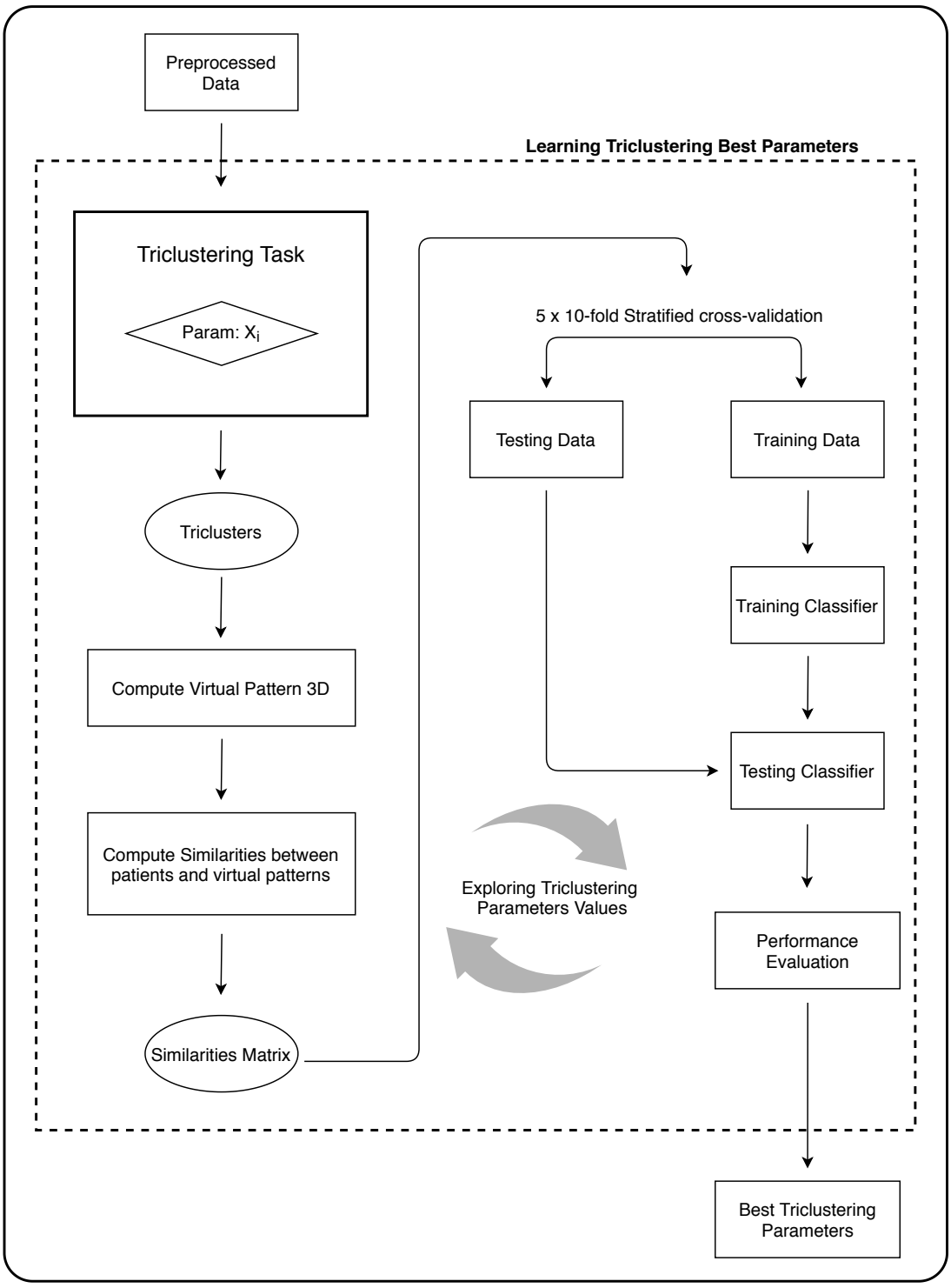


Figure 3.3: Learning Triclustering Best Parameters: Workflow

the trend of the tricluster totally (1) and nothing otherwise (0); (2) compute the Euclidean distance; or (3) compute Pearson correlation between the 3D virtual pattern $\mathcal{P}$ and the equivalent pattern (same features and contexts) of p_i.

We denote last two assessments as Virtual Distance 3D and Virtual Correlation 3D, and define them as follows:

Definition 2. (Virtual Distance 3D). The virtual distance between an observation p_i and a tricluster $\mathcal{T}$ is defined as

$$\texttt{VD}_{\texttt{3D}}(p_i, \mathcal{T}) = E(p_i, \rho) = \sqrt{\sum_{e=1}^{I}(p_{i_e} - \rho_e)^2} \tag{3.2}$$

Definition 3. (Virtual Correlation 3D). The virtual correlation between an object p_i and a tricluster $\mathcal{T}$ is defined as

$$\texttt{VC}_{\texttt{3D}}(p_i, \mathcal{T}) = r(p_i, \rho) = \frac{\sum_{e=1}^{I}(p_{i_e} - \bar{p_i})(\rho_e - \bar{\rho})}{\sqrt{\sum_{e=1}^{I}(p_{i_e} - \bar{p_i})^2 \sum_{e=1}^{I}(\rho_e - \bar{\rho})^2}} \tag{3.3}$$

Considering as an example a Tricluster $T(I, J, K)$, mined from three-way data $S(X, Y, Z)$, composed by 3 objects, 3 features and 3 contexts, such that $I = \{X_1, X_3, X_7\},\ J = \{Y_1, Y_3, Y_7\},\ K = \{Z_2, Z_3, Z_4\}$. Y_1 and Y_3 contains only categorical values. For simplicity, consider $T = \{B_2, B_3, B_4\}$:

$$B_2 = \begin{bmatrix} 1 & 3.1 & 5 \\ 1 & 2.8 & 3 \\ 3 & 2.1 & 10 \end{bmatrix}; B_3 = \begin{bmatrix} 2 & 3.0 & 3 \\ 3 & 2.8 & 3 \\ 3 & 2.9 & 9 \end{bmatrix}; B_4 = \begin{bmatrix} 3 & 2.9 & 3 \\ 2 & 2.9 & 3 \\ 3 & 2.4 & 8 \end{bmatrix}$$

and an object (patient) $P(X_p, I, K)$ defined as $P = \{C_2, C_3, C_4\} : C_2 = \begin{bmatrix} 1 & 2.22 & 5 \end{bmatrix};\ C_3 = \begin{bmatrix} 1 & 2.26 & 7 \end{bmatrix};\ C_4 = \begin{bmatrix} 2 & 2.35 & 8 \end{bmatrix}$

The Virtual Patterns are: $\rho(B_2) = \begin{bmatrix} 1 & 2.6667 & 5 \end{bmatrix}; \rho(B_3) = \begin{bmatrix} 3 & 2.9 & 3 \end{bmatrix}; \rho(B_4) = \begin{bmatrix} 3 & 2.7333 & 3 \end{bmatrix};$ and $\rho(T) = \begin{bmatrix} 3 & 2.7667 & 3 \end{bmatrix}$.

After computing similarity matrices according to the proposed approaches, a 5×10-fold Stratified Cross-Validation is performed using the Triclustering-based Classifier in order to find the best tricluster-

ing parameters, using classification performance as a metric. The best parameters then fed to the next step.

3.4 Learning Triclustering-based Classifier

Figure 3.4 depicts the steps involved in learning the final model. With the best parameters output in the previous step, a final iteration is performed in order to obtain the final Triclustering-based predictive model and the final triclusters, then used to make predictions in the next step.

3.5 Using the Triclustering-based Predictive Model

After learning the final triclustering-based predictive model, it can be used to classify new three-way objects. To do this, it is necessary to first calculate the array of similarities between the new object and the triclusters obtained in the previous steps. This array will be fed to the classifier that will in turn return the classification for the new object with a percentage of accuracy. Figure 3.5 depicts an example using clinical three-way data (case study described in the section 1.3).

3.6 Final Remarks

This Chapter presented a promising approach to learn predictive models based on temporal triclustering. We proposed a new triclustering algorithm able to identify temporal patterns and cope with missing values used in a triclustering-based classifier. In clinical domains, temporal data analysis is an important task since the assessment of patients evolution during their follow-up is a critical task for clinicians. As such, being able to identify progression patterns and use them to predict how will the patient evolve is key. With this in mind, we applied the proposed methodology to ALS patients' data. Results are presented in the next chapter.

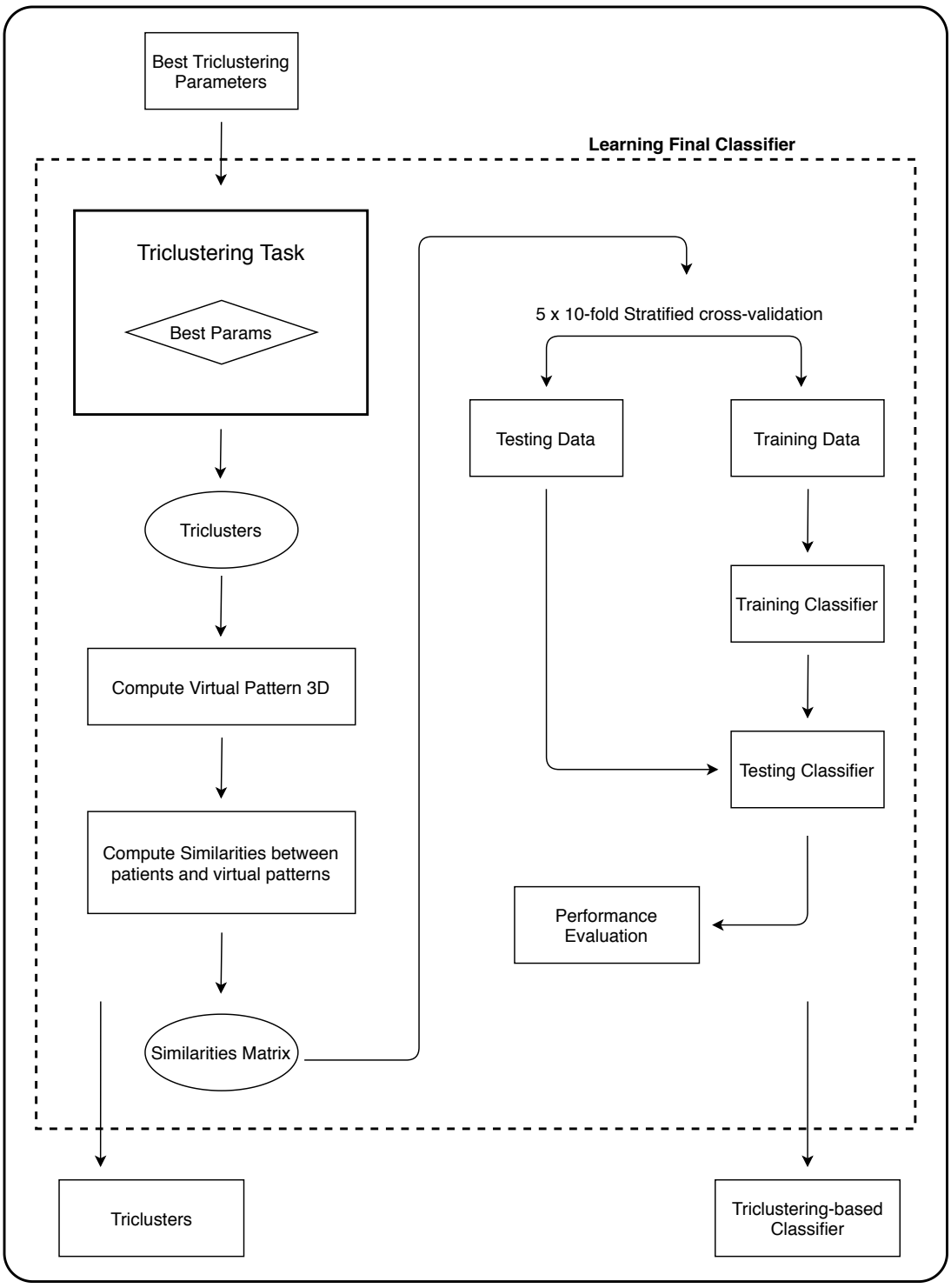


Figure 3.4: Learning Final Triclustering-based Model: Workflow.

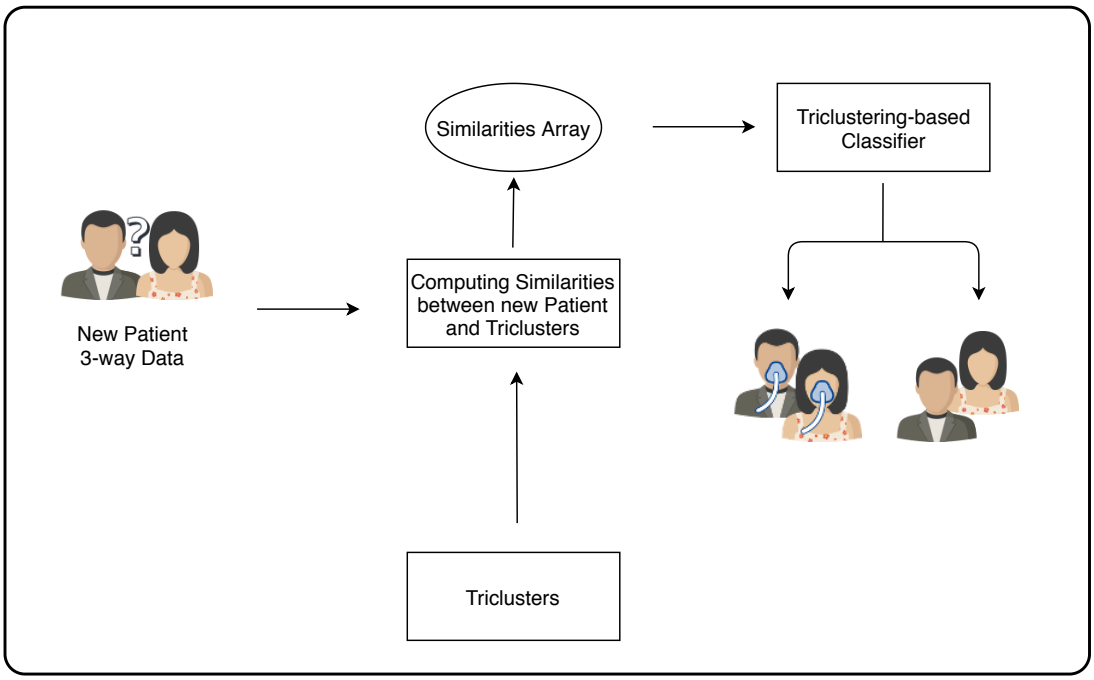


Figure 3.5: Using the Model: Workflow with three-way Clinical Data

Learning Predictive Models Using a Triclustering-based Classifier: A Case Study in ALS

4

In this chapter we use the triclustering-based classification approach proposed in Chapter 3 to learn predictive models for the ALS case study. The goal is to predict whether a patient will require NIV within 90 days after last evaluation, using 3W data corresponding to patient follow-up.

We first discuss ALS data preprocessing, then present and discuss results for the prognostic models, and finally tackle model interpretability.

4.1 Preprocessing ALS Data

The Lisbon ALS clinic dataset described in Section 1.3 was preprocessed as described by Carreiro et al. [8] and Pires et al. [39] to obtain patient snapshots and then compute the Evolution class for each snapshot using NIV administration date: a patient is labelled `Y`, if 90 days after the snapshot he/she was administrated NIV, and `N` otherwise.

We performed experiments using training examples composed by: 3, 4 and 5 consecutive snapshots (CS) for each patient (corresponding to clinical evaluations at 3, 4 and 5 consecutive appointments, respectively) were used as features, and the NIV evolution value of the last snapshot was used as class. For each remaining patient, we performed missing value imputation by using values in previous appointments to input latter missing values (Last Observation Carried Forward), when possible, and mean/mode of all patients values, otherwise.

After tackling missing values, we had to deal with class imbalance. In our case, and due to the time

window of 90 days (next appointment) used as case study, the number of patients labeled as `N`, non-evolutions, largely outnumbered those labelled as `Y`, identifying the patients requiring NIV within 90 days, key for the learning task. To deal with this issue, we first used a Random Undersampler (RU) to reduce `N` examples until obtaining a class proportion of 2/3 - 1/3 and then used SMOTE [9] to balance datasets to 50%/50% class proportion. Values for the class proportions obtained after RU and SMOTE are depicted in Table 4.1.

Table 4.1: Number and Class Distribution of Learning Examples.

	Total		**RU**		**SMOTE**	
	N	Y	N	Y	N	Y
3 CS	1721	227	457	227	454	454
4 CS	1335	166	332	166	332	332
5 CS	1038	121	242	121	242	242

4.2 Learning Prognostic Models

This section provides results and discussion for the following experiments (with and without missing values imputation):

1. Baseline: Random Forests classification with original features
2. Triclustering-based Classification with all data
3. Triclustering-based Classification with patient stratification

4.2.1 Baseline Results: Random Forests with original features

To obtain baseline results, to whom compare the predictive models learned with triclustering-based classifiers, we trained a Random Forest using the original features (respiratory tests and ALSFRS-R scores) treated as independent features (number of CS $\times$ 10 features). Table 4.2 shows the baseline results. We can observe that baseline classifier achieved classification accuracies around 0.80 in CV with low standard deviation. Sensitivity and specificity show approximately the same values, meaning all classifiers perform well when predicting both classes.

Table 4.2: Baseline Results using Random Forests and Original Features (with missing values imputation)

CS	AUC	Accuracy	Sensitivity	Specificity
3	0.87 ± 0.0326	0.79 ± 0.0371	0.83 ± 0.0437	0.74 ± 0.0702
4	0.89 ± 0.0370	0.80 ± 0.0509	0.84 ± 0.0659	0.75 ± 0.0899
5	0.89 ± 0.0462	0.80 ± 0.0577	0.78 ± 0.0818	0.75 ± 0.0879

Random Forest itself can deal with missing values, so we decide to verify the similar baseline results disregarding imputation step on data preprocessing, in order to verify if results are skewed by the artificial values planted in those stage. Table 4.3 shows the results, and as we can see, with this case study data missing values imputation does not affect the classifier performance, since results are quite the same.

Table 4.3: Baseline Results using Random Forests and Original Features (allowing missing values)

CS	AUC	Accuracy	Sensitivity	Specificity
3	0.88 ± 0.0314	0.80 ± 0.0408	0.84 ± 0.0582	0.76 ± 0.0663
4	0.90 ± 0.0371	0.80 ± 0.0473	0.85 ± 0.0614	0.76 ± 0.0831
5	0.89 ± 0.0506	0.80 ± 0.0581	0.85 ± 0.0670	0.75 ± 0.0994

4.2.2 Triclustering-based Classification Results

To test the proposed triclustering-based classifier on the ALS case study, allowing fair comparisons with the baseline, we also use Random Forests as classifier.

Since TriCluster allows different parameterizations, potentially discovering triclusters with different types of coherence, we run the algorithm using three different settings: Unconstrained, to capture all coherent triclusters across the three dimensions (x-patient, y-feature and z-time); $\delta^x = \delta^y = \delta^z = 0$, to capture triclusters with constant values across the three dimensions; and $\delta^x = 0$, to force constant coherence on patient dimension while relaxing the others two. Together with these parameterizations, we varied deletion (η) and merging (γ) thresholds from 0.45 to 0.95 (and none). We set minimum number of features and time-points in each tricluster as 1 and 2 respectively, and minimum number of patients as 25, 20 and 15 for 3, 4 and 5 consecutive snapshots (CS). We performed the experiments with data preprocessed with missing values imputation and without.

To compute the similarities matrix, to be used by each triclustering-based classifier, we tried the three proposed approaches: binary (B), distance (D) and correlation (C). Regarding the last two approaches,

was computed the similarities between the patients and the different biclusters that compose the triclusters, since according to the case study and for interpretability concerns, this will make the features more informative, than when considering similarities with the general trend of the complete tricluster.

Table 4.4 shows the best triclustering parameters obtained with the different CS experiments. As we can see, Unconstrained was chosen as the best parameter in the most of the experiments.

Table 4.4: Learned Triclustering Best Parameters

AP	CS	With MV imputation	Without MV imputation
B	3	$\delta^x = \delta^y = \delta^z = 0$	Unconstrained
	4	Unconstrained	Unconstrained
	5	Unconstrained	Unconstrained
D	3	Unconstrained; $\eta = 0.95$	Unconstrained
	4	Unconstrained	Unconstrained
	5	Unconstrained	Unconstrained
C	3	Unconstrained; $\gamma = 0.50$	Unconstrained
	4	Unconstrained	$\delta^x = 0$
	5	Unconstrained; $\gamma = 0.95$	Unconstrained

Table 4.5: Performance Evaluation Results of Triclustering-based Classifier using original TriCluster Algorithm

AP	CS	AUC	Accuracy	Sensitivity	Specificity
B	3	0.74 ± 0.0011	0.69 ± 0.0037	0.76 ± 0.0023	0.66 ± 0.0028
	4	0.74 ± 0.0016	0.65 ± 0.0016	0.66 ± 0.0165	0.62 ± 0.0093
	5	0.75 ± 0.0087	0.68 ± 0.0028	0.66 ± 0.0135	0.70 ± 0.0022
D	3	0.79 ± 0.0384	0.72 ± 0.0364	0.72 ± 0.0561	0.72 ± 0.0631
	4	0.79 ± 0.0021	0.70 ± 0.0071	0.71 ± 0.0096	0.72 ± 0.0045
	5	0.82 ± 0.0016	0.76 ± 0.0048	0.76 ± 0.0088	0.75 ± 0.0044
C	3	0.79 ± 0.0010	0.71 ± 0.0064	0.71 ± 0.0089	0.71 ± 0.0050
	4	0.78 ± 0.0023	0.70 ± 0.0066	0.69 ± 0.0045	0.71 ± 0.0096
	5	0.85 ± 0.0019	0.75 ± 0.0086	0.74 ± 0.0146	0.75 ± 0.0049

Table 4.5 shows the results obtained by Triclustering-based classifier using the original version of triCluster [47] and Table 4.6 depicts the results obtained with the extended version TCtriCluster (see Algorithm 1). As we can see although these results still do not exceed the same results obtained by the

baseline, they generally exceed the results of Table 4.5, which proves the effectiveness of the temporal constraint, allowing to improve group coherence and consequently classifier performance.

Table 4.6: Performance Evaluation Results of Triclustering-based Classifier learned from ALS Lisbon Clinic Data using TCtriCluster Algorithm (with missing values imputation)

AP	CS	AUC	Accuracy	Sensitivity	Specificity
B	3	0.78 ± 0.0453	0.72 ± 0.0444	0.77 ± 0.0748	0.67 ± 0.0675
	4	0.81 ± 0.0408	0.74 ± 0.0444	0.73 ± 0.0763	0.74 ± 0.0719
	5	0.85 ± 0.0544	0.78 ± 0.0630	0.81 ± 0.0871	0.75 ± 0.1004
D	3	0.84 ± 0.0384	0.76 ± 0.0364	0.78 ± 0.0561	**0.74 ± 0.0631**
	4	0.86 ± 0.0389	0.78 ± 0.0432	0.80 ± 0.0706	0.76 ± 0.0830
	5	0.85 ± 0.0615	0.76 ± 0.0669	0.78 ± 0.0967	0.75 ± 0.0979
C	3	0.84 ± 0.0380	0.75 ± 0.0379	0.78 ± 0.0560	0.72 ± 0.0649
	4	0.85 ± 0.0428	0.75 ± 0.0456	0.76 ± 0.0780	0.74 ± 0.0767
	5	0.85 ± 0.0546	0.75 ± 0.0598	**0.80 ± 0.0856**	0.71 ± 0.0967

Analyzing the results in Table 4.6, we can verify that the highest precision values were obtained by the classifiers using the distance approach to compute the similarity matrix, and that these are comparable to those obtained by the baseline. We note that despite not outperforming the baseline, these results improve the interpretability of the model, since the different temporal patterns used can uncover disease progression patterns and help clinicians in their predictive task.

Observing the results obtained with preprocessing data without missing values imputation, depicted in Table 4.7, we can verify that the results are quietly the same as those obtained with missing values imputation, meaning that using data with missing values does not affect the results obtained by the classifier.

4.2.3 Results with Patient Stratification

Using the same data, we stratified the patients in three groups according with their disease progression and following the approach used by Pires et al. [39]. The patients thus were stratified in Slow, Neutral and Fast progressors according to a Progression Rate (PR) value, computed by:

$$\mathtt{PR} = \frac{48 - \text{ALSFRS-R}_{\text{1st Visit}}}{\Delta t_{\text{1st Symptoms; 1st Visit}}}, \tag{4.1}$$

Table 4.7: Performance Evaluation Results of Triclustering-based Classifier learned from ALS Lisbon Clinic Data using TCtriCluster Algorithm (without missing values imputation)

AP	CS	AUC	Accuracy	Sensitivity	Specificity
	3	0.79 ± 0.0415	0.73 ± 0.0452	0.78 ± 0.0536	0.68 ± 0.0678
B	4	0.80 ± 0.0458	0.74 ± 0.0563	0.72 ± 0.0615	0.72 ± 0.0636
	5	0.85 ± 0.0498	0.78 ± 0.0630	0.79 ± 0.0745	0.76 ± 0.0958
	3	0.84 ± 0.0416	0.75 ± 0.0484	0.78 ± 0.0624	0.73 ± 0.0840
D	4	0.87 ± 0.0402	0.79 ± 0.0486	0.81 ± 0.0703	0.77 ± 0.0845
	5	0.85 ± 0.0538	0.75 ± 0.0586	0.79 ± 0.0922	0.72 ± 0.0933
	3	0.83 ± 0.0401	0.74 ± 0.0393	0.73 ± 0.0574	0.74 ± 0.0539
C	4	0.85 ± 0.0429	0.77 ± 0.0519	0.78 ± 0.0689	0.76 ± 0.0912
	5	0.85 ± 0.0538	0.75 ± 0.0554	0.77 ± 0.0890	0.73 ± 0.0963

where 48 is the maximum score for ALSFRS-R feature, $\text{ALSFRS-R}_{\text{1st Visit}}$ is the ALSFRS-R score in the first appointment (diagnosis) and $\Delta t_{\text{1st Symptoms; 1st Visit}}$ is the time in months between the dates of first symptoms and the first appointment [39]. Progression rates are computed for each patient, and then based on these values they are divided into three groups based the distribution of PR values, and as suggested by the clinicians: 25% of patients with lower and higher values are stratified as Slow and Fast progressors, respectively. The remaining 50% are grouped and considered Neutral progressors. Table 4.8 shows the class distribution after patient stratification.

After stratifying the patients in the three groups of patients, we apply our triclustering-based methodology to learn a specialized predictive model for each group of patients. Table 4.9 shows the results, obtained in the same way as those for the model learned with all data, as baseline. As expected, the results after patient stratification outperform the results obtained when learning from all patients, since patients are now more homogeneous. The small number patients with 4 and 5 CS in the Fast group prevented us from obtaining reliable results in these cases.

Table 4.10 shows the results obtained by the specialized models for each disease progression group, according to the number of consecutive snapshots considered. We can see that the results outperform the baseline for 3 and 4 CS in the Neutral group, using the distance approach for similarities, confirming the evidence that this approach is the most appropriate to this type of data. Comparing these results with those obtained by the general classifier we can see that there is a noticeable increase in classification performance, resulting from being able to discover and use group specific disease progression patterns. Since the general model learns from heterogeneous patients, with a wide spectrum of disease progression patterns, where the Neutral group is dominant, it potentially misses significant patterns from patients who

Table 4.8: Distribution of Classes with Patient Stratification

		TOTAL		**RU**		**SMOTE**	
		N	Y	N	Y	N	Y
3CS	Slow	921	58	116	58	116	116
	Neutral	600	134	268	134	268	268
	Fast	120	22	44	22	44	44
4CS	Slow	767	50	100	50	100	100
	Neutral	426	96	192	96	192	192
	Fast	80	11	22	11	22	22
5CS	Slow	635	43	86	43	86	86
	Neutral	299	67	134	67	134	134
	Fast	55	5	10	5	10	10

not follow a common disease progression trend and whose specific disease progression patterns are not discovered. Despite the good results in the Neutral group, the results concerning the Slow and Fast groups slightly decreased, since the class imbalance in these groups is even more accentuated than in complete dataset or in the neutral group. This imbalance has hampered the computation of some metrics given the bias.

We decided to verify again if the results are skewed by missing values imputation and we performed same experiments without missing values imputation. The results can be seen in Table 4.12, and it is possible to verify that results are slightly the same with a little penalization for those obtained without missing values imputation for some disease progression groups.

In order to assess whether the general model would also be better for Slow and Faster progressors in the same way that the specialized model would be better for the Neutral group, we perform experiments with the general model to evaluate its performance with patients from the different three groups. Table 4.13 shows the results obtained, and we can see that even for Slow and Fast progressors the general model performs well, but specialized models achieved more balanced results in the prediction of both classes (values of sensitivity and specificity are closer). As already mentioned the general model by the specialized model in the Neutral group.

We note however, that these results could be improved with a larger collection of patients data from the Slow and Fast groups, as shown by the slightly higher values of standard deviation in these cases.

Table 4.9: Baseline Results: Random Forests with Original Features per Disease Progression Group

CS	Group	AUC	Accuracy	Sensitivity	Specificity
3	Slow	0.89 ± 0.0713	0.80 ± 0.0841	0.84 ± 0.1083	0.76 ± 0.1201
	Neutral	0.87 ± 0.0407	0.79 ± 0.0474	0.80 ± 0.0677	0.78 ± 0.0787
	Fast	0.81 ± 0.1668	0.74 ± 0.1454	0.74 ± 0.2627	0.74 ± 0.1998
4	Slow	0.93 ± 0.0986	0.86 ± 0.0895	0.87 ± 0.0996	0.85 ± 0.1236
	Neutral	0.85 ± 0.0683	0.771 ± 0.0645	0.81 ± 0.0797	0.73 ± 0.1112
	Fast				
5	Slow	0.92 ± 0.0562	0.839 ± 0.0730	0.84 ± 0.1166	0.84 ± 0.1178
	Neutral	0.87 ± 0.0729	0.79 ± 0.0788	0.79 ± 0.1021	0.79 ± 0.1235
	Fast				

4.3 Model Interpretability

The relevance of a triclustering-based classifier methodology should be evaluated not only by analysing its performance regarding classification results, but also by its potential concerning model interpretability. This means analysing the temporal patterns uncovered through triclustering, regarding their domain relevance, clinical in this case, and their importance for the predictive model, by computing the pattern/feature importance according to the learnt classifier. To this aim, we chose to analyse the patterns discovered when triclustering the 3 CS dataset. The goal is to understand what are the most relevant features, what features appear together, and whether the patterns found to be relevant in the general model, putative patterns of the average patient, differ from those relevant to the specialized models, that should be group-specific, highlighting disease progression patterns of Slow, Neutral and Fast progressors.

Table 4.18 shows the characterization of triclusters obtained by the 4 learnt models (General, Slow, Neutral and Fast), where $|I|$, $|J|$ and $|K|$ represent the number of patients, features and time-points composing each tricluster. The patterns are represented by feature values across the time-points. Tables 4.14, 4.15, 4.16 and 4.17 depict the most important as used patterns by the classifier, ranked by their feature importance.

An overall analysis of the most important patterns discovered shows that the majority of the patterns refer to the last snapshot/time-point of the tricluster. This makes sense, since this is the snapshot closer to the target. However, other patterns not corresponding to the last snapshot that remain important, are also relevant for further clinical analysis, since these features can be relevant in identifying disease progression pattern leading to the need of NIV in a given time-window.

Table 4.10: Performance Evaluation Results obtained with Triclustering-based Classifier (Specialized for each Disease Progression Group, with missing values imputation)

AP	CS	Group	AUC	Accuracy	Sensitivity	Specificity
D	3	Slow	0.84 ± 0.0933	0.77 ± 0.0924	0.81 ± 0.1386	0.73 ± 0.1377
		Neutral	0.86 ± 0.0415	**0.794 ± 0.0393**	0.79 ± 0.0583	0.80 ± 0.0694
		Fast	0.70 ± 0.1798	0.66 ± 0.1582	0.81 ± 0.2211	0.51 ± 0.2397
	4	Slow	0.87 ± 0.0797	0.81 ± 0.0835	0.85 ± 0.1063	0.77 ± 0.1221
		Neutral	0.84 ± 0.0635	**0.774 ± 0.0731**	0.77 ± 0.0826	0.78 ± 0.1059
		Fast	0.70 ± 0.2666	0.74 ± 0.2094	0.80 ± 0.2442	0.68 ± 0.3177
	5	Slow	0.89 ± 0.0739	**0.841 ± 0.0908**	0.88 ± 0.1221	0.81 ± 0.1178
		Neutral	0.81 ± 0.0808	0.72 ± 0.0839	0.71 ± 0.1118	0.74 ± 0.1204
		Fast				
C	3	Slow	0.85 ± 0.0827	0.75 ± 0.0874	0.76 ± 0.1394	0.75 ± 0.1470
		Neutral	0.86 ± 0.0404	0.78 ± 0.0471	0.76 ± 0.0769	0.79 ± 0.0746
		Fast	0.66 ± 0.1768	0.66 ± 0.1586	0.80 ± 0.2201	0.51 ± 0.2447
	4	Slow	0.85 ± 0.0762	0.79 ± 0.0933	0.81 ± 0.1237	0.78 ± 0.1436
		Neutral	0.83 ± 0.0655	0.77 ± 0.0660	0.78 ± 0.0819	0.76 ± 0.0968
		Fast	0.63 ± 0.2414	0.70 ± 0.1854	0.77 ± 0.2424	0.62 ± 0.2978
	5	Slow	0.88 ± 0.0822	0.83 ± 0.0858	0.88 ± 0.1021	0.78 ± 0.1343
		Neutral	0.81 ± 0.0780	0.73 ± 0.0800	0.71 ± 0.1159	0.75 ± 0.1194
		Fast				

Table 4.11: Learned Triclustering Best Parameters: TCtricluster without missing values imputation for each disease progression group

AP	CS	Group	Best Parameters
D	3	Slow	Unconstrained
		Neutral	Unconstrained
		Fast	$\delta^x = \delta^y = \delta^z = 0$
	4	Slow	Unconstrained; $\gamma = 0.90$
		Neutral	Unconstrained
		Fast	Unconstrained; $\eta = 0.75$
	5	Slow	Unconstrained
		Neutral	Unconstrained
C	3	Slow	Unconstrained
		Neutral	Unconstrained
		Fast	Unconstrained; $\eta = 0.95$
	4	Slow	Unconstrained; $\gamma = 0.95$
		Neutral	Unconstrained
		Fast	Unconstrained; $\eta = 0.75$
	5	Slow	Unconstrained; $\gamma = 0.95$
		Neutral	Unconstrained; $\eta = 0.50$

As expected, most of the important set of features used by the General model are the same as those discovered for the Neutral group, corresponding to the common tendency within the set of all patients, corresponding to the average patient (Neutral progressor). However, the highest values observed for the features in Neutral model expose the influence that the other groups (Slow and Fast progressors) created in the General model. Comparing patterns of Slow progressors with those of Neutral progressors, we can confirm that, as expected the values for the same set of features are smaller in the first.

When analysing the most important patterns used by the model learnt for Fast progressors, it is interesting to witness that these patterns are typically very different from those of the other models. This confirms the fact that Fast progressors are very different patients and can be useful to help understanding their unique progression patterns. Furthermore, a quick identification of Fast progressors, whose clinical condition degrades very quickly, through their progression patterns, can promote timely intervention and prolong survival.

Table 4.12: Performance Evaluation Results obtained with Triclustering-based Classifier (Specialized for each Disease Progression Group, without missing values imputation)

AP	CS	Group	AUC	Accuracy	Sensitivity	Specificity
D	3	Slow	0.87 ± 0.0686	0.79 ± 0.0825	0.81 ± 0.1264	0.76 ± 0.1135
		Neutral	0.85 ± 0.0511	0.78 ± 0.0615	0.77 ± 0.0935	0.80 ± 0.0797
		Fast	0.68 ± 0.1674	0.63 ± 0.1543	0.80 ± 0.1975	0.45 ± 0.2476
	4	Slow	0.87 ± 0.0846	0.82 ± 0.0851	0.87 ± 0.1157	0.76 ± 0.1368
		Neutral	0.82 ± 0.0686	0.74 ± 0.0665	0.73 ± 0.0934	0.76 ± 0.1045
		Fast	0.70 ± 0.2272	0.68 ± 0.2138	0.73 ± 0.2843	0.63 ± 0.3029
	5	Slow	0.89 ± 0.0707	0.81 ± 0.0882	0.83 ± 0.1214	0.79 ± 0.1160
		Neutral	0.78 ± 0.0791	0.72 ± 0.0765	0.71 ± 0.1206	0.74 ± 0.1140
		Fast				
C	3	Slow	0.87 ± 0.0635	0.77 ± 0.0813	0.79 ± 0.1296	0.75 ± 0.1088
		Neutral	0.85 ± 0.0527	0.77 ± 0.0618	0.74 ± 0.0893	0.79 ± 0.0874
		Fast	0.60 ± 0.1772	0.62 ± 0.1457	0.81 ± 0.2069	0.45 ± 0.2247
	4	Slow	0.88 ± 0.0791	0.81 ± 0.0845	0.83 ± 0.1207	0.78 ± 0.1433
		Neutral	0.83 ± 0.0587	0.75 ± 0.0552	0.73 ± 0.0805	0.77 ± 0.0930
		Fast	0.70 ± 0.2590	0.71 ± 0.1877	0.77 ± 0.2807	0.64 ± 0.2736
	5	Slow	0.86 ± 0.0725	0.79 ± 0.0872	0.80 ± 0.1174	0.77 ± 0.1168
		Neutral	0.77 ± 0.0806	0.72 ± 0.0801	0.69 ± 0.1074	0.75 ± 0.1243
		Fast				

Table 4.13: Performance Evaluation Results obtained with Triclustering-based Classifier (General) per Disease Progression Group

CS	Group	AUC	Accuracy	Sensitivity	Specificity
3	Slow	0.85 ± 0.1094	0.81 ± 0.0754	0.67 ± 0.3053	0.84 ± 0.0765
	Neutral	0.82 ± 0.0536	0.74 ± 0.0924	0.78 ± 0.0745	0.69 ± 0.0879
	Fast	0.78 ± 0.0965	0.70 ± 0.0956	0.76 ± 0.1082	0.59 ± 0.1629
4	Slow	0.89 ± 0.0986	0.82 ± 0.0895	0.70 ± 0.2680	0.86 ± 0.0923
	Neutral	0.84 ± 0.0683	0.77 ± 0.0776	0.80 ± 0.0897	0.74 ± 0.1420
	Fast	0.81 ± 0.1100	0.75 ± 0.0937	0.85 ± 0.1086	0.57 ± 0.2116
5	Slow	0.86 ± 0.0474	0.80 ± 0.0844	0.65 ± 0.2256	0.85 ± 0.1192
	Neutral	0.82 ± 0.0760	0.72 ± 0.0508	0.78 ± 0.0784	0.63 ± 0.1192
	Fast	0.75 ± 0.1198	0.74 ± 0.0890	0.85 ± 0.0794	0.52 ± 0.1532

4.4 Final Remarks

This chapter presented promising results on ALS prognostic prediction with temporal using the triclustering-based approach proposed in Chapter 3. We show that model interpretability is a key advantage for better understanding patients' evolution and pinpoint relevant patterns for clinical analysis. Besides the relevance of these results obtained with temporal data, commonly clinical data is composed not only by temporal data but also by static data, which we discarded in the presented approach. Considering static data, will provide more information on patients that can improve prognostic prediction models and interpretability. With this in mind, we developed a new classification approach combining biclustering in static data with triclustering in temporal data, presented in the next chapter.

Table 4.14: Top 20 patterns discovered in the general model trained with all data (with 3 CS)

Rank	#Tricluster	TP	Feat. Import	Pattern
1	10	2	0.031960	ALS-FRS = 38
2	18	2	0.029058	ALS-FRS-R = 46
3	54	0	0.027954	MIP = 53.5705; MEP = 65.0705
4	53	0	0.023513	FVC = 88.2423; MEP = 66.6423
5	12	2	0.022829	ALS-FRS = 31
6	54	1	0.022564	MIP = 53.2824; MEP = 66.1471
7	35	2	0.018606	ALS-FRSsUL = 9; R = 12
8	13	2	0.017853	ALS-FRS = 30
9	53	1	0.017221	FVC = 89.15; MEP = 67.5577
10	55	1	0.012898	ALS-FRS = 24; ALS-FRS-R = 32
11	25	2	0.012194	ALS-FRS-R = 33
12	66	1	0.011938	ALS-FRS = 38
13	7	1	0.011871	ALS-FRS = 26; ALS-FRS-R = 34
14	15	2	0.011854	ALS-FRS = 27
15	69	1	0.011695	ALS-FRS = 27; R = 12
16	0	1	0.011637	ALS-FRS = 24
17	87	1	0.011274	ALS-FRSsUL = 12; ALS-FRSsLL = 12
18	29	2	0.011250	ALS-FRSb = 12; R = 12
19	22	2	0.010727	ALS-FRS-R = 35
20	83	1	0.010561	ALS-FRS-R = 46

Table 4.15: 20 Best discovered patterns in the specialized model for Slow group (with 3 CS)

Rank	#Tricluster	TP	Feat. Import	Pattern
1	1	2	0.059431	ALS-FRS = 37; ALS-FRS-R = 45
2	2	2	0.043924	ALS-FRS-R = 45
3	120	0	0.027597	ALS-FRS = 37; ALS-FRS-R = 45
4	120	1	0.027366	ALS-FRS = 37; ALS-FRS-R = 45
5	46	2	0.026735	ALS-FRSb = 12; ALS-FRSr = 4; R = 12
6	71	2	0.026688	ALS-FRSsLL = 12; ALS-FRSr = 4
7	47	2	0.025992	ALS-FRSsLL = 12; R = 12
8	3	0	0.025549	ALS-FRS = 37
9	0	1	0.024103	ALS-FRS = 37; ALS-FRS-R = 45
10	4	0	0.023486	ALS-FRS-R = 45
11	62	1	0.023258	ALS-FRSsUL = 9; ALS-FRSr = 4; R = 12
12	31	2	0.023019	ALS-FRS-R = 12
13	2	0	0.022919	ALS-FRS-R = 45
14	1	1	0.022609	ALS-FRS = 37; ALS-FRS-R = 45
15	2	1	0.022548	ALS-FRS-R = 45
16	3	1	0.021562	ALS-FRS = 37
17	4	1	0.020758	ALS-FRS-R = 45
18	71	1	0.019888	ALS-FRSsLL = 12; ALS-FRSr = 4
19	60	1	0.019324	ALS-FRSsUL = 12; R = 12
20	55	1	0.019007	ALS-FRSsUL = 12; ALS-FRSr = 4

Table 4.16: 20 Best discovered patterns in the specialized model for Neutral group (with 3 CS)

Rank	#Tricluster	TP	Feat. Import	Pattern
1	4	2	0.018618	ALS-FRS = 38; ALS-FRS-R = 46
2	260	2	0.016051	ALS-FRSsUL = 12; ALS-FRSsLL = 12
3	238	2	0.014534	ALS-FRSsUL = 12; ALS-FRSr = 4
4	5	2	0.013115	ALS-FRS = 35
5	291	2	0.013027	ALS-FRSsUL = 12; ALS-FRSr = 4; R = 12
6	551	1	0.012610	MIP = 58.355; MEP = 66.5
7	261	2	0.012074	ALS-FRSsUL = 12
8	1	2	0.011431	ALS-FRS = 29; ALS-FRS-R = 37
9	11	2	0.011142	ALS-FRS-R = 43
10	551	0	0.011140	MIP = 54.2; MEP = 63.8
11	3	2	0.011110	ALS-FRS = 33; ALS-FRS-R = 41
12	190	2	0.010652	ALS-FRSb = 12; ALS-FRSr = 4
13	550	0	0.010239	FVC = 87.3571; MEP = 65.7071
14	550	1	0.010007	FVC = 88.7429; MEP = 66.9928
15	2	2	0.009997	ALS-FRS = 31; ALS-FRS-R = 39
16	13	2	0.009706	ALS-FRS-R = 41
17	10	2	0.009301	ALS-FRS-R = 39
18	226	2	0.008809	ALS-FRSb = 12; ALS-FRSr = 4; R = 12
19	244	2	0.007572	ALS-FRSb = 12; R = 12
20	342	2	0.007409	ALS-FRSsLL = 12; R = 12

Table 4.17: Discovered patterns in the specialized model for Fast group (with 3 CS)

Rank	#Tricluster	TP	Feat. Import	Pattern
1	1	2	0.251100	R = 12
2	2	2	0.198099	ALS-FRSr = 3
3	2	1	0.100364	ALS-FRSr = 4
4	0	0	0.100323	R = 12
5	1	0	0.093527	R = 12
6	2	0	0.093015	ALS-FRSr = 4
7	1	1	0.091529	R = 12
8	0	1	0.072043	R = 12

Table 4.18: Characterization of Triclusters obtained with different learned models (with 3 CS)

General Model				
#Tricluster	$\|I\|$	$\|J\|$	$\|K\|$	Patterns
0	24	1	2	[ALS-FRS=28], [ALS-FRS=24]
7	25	2	2	[ALS-FRS=27; ALS-FRS-R=35], [ALS-FRS=26; ALS-FRS-R=34]
10	59	1	3	[ALS-FRS=38], [ALS-FRS=38], [ALS-FRS=38]
12	54	1	3	[ALS-FRS=35], [ALS-FRS=35], [ALS-FRS=31]
13	42	1	3	[ALS-FRS=36], [ALS-FRS=36], [ALS-FRS=30]
15	41	1	3	[ALS-FRS=33], [ALS-FRS=33], [ALS-FRS=27]
18	56	1	3	[ALS-FRS-R=46], [ALS-FRS-R=46], [ALS-FRS-R=46]
22	38	1	3	[ALS-FRS-R=41], [ALS-FRS-R=41], [ALS-FRS-R=35]
25	28	1	3	[ALS-FRS-R=40], [ALS-FRS-R=36], [ALS-FRS-R=33]
29	151	2	3	[ALS-FRSb=12; R=12], [ALS-FRSb=12; R=12], [ALS-FRSb=12; R=12]
35	20	2	3	[ALS-FRSsUL=9; R=12], [ALS-FRSsUL=9; R=12], [ALS-FRSsUL=9; R=12]
53	26	2	2	[FVC=88.2423; MEP=66.6423], [FVC=89.15; MEP=67.5577]
54	34	2	2	[MIP=53.5706; MEP=65.0706], [MIP=57.2824; MEP=66.1471]
55	22	2	2	[ALS-FRS=27; ALS-FRS-R=35], [ALS-FRS=24; ALS-FRS-R=32]
66	31	1	2	[ALS-FRS=38], [ALS-FRS=38]
69	23	2	2	[ALS-FRS=37; R=12], [ALS-FRS=37; R=12]
83	29	1	2	[ALS-FRS-R=46], [ALS-FRS-R=46]
87	44	2	2	[ALS-FRSsUL=12; ALS-FRSsLL=12], [ALS-FRSsUL=12; ALS-FRSsLL=12]
Specialized Model for Slow Progressors				
#Tricluster	$\|I\|$	$\|J\|$	$\|K\|$	Patterns
0	29	2	2	[ALS-FRS=38.0; ALS-FRS-R=46.0], [ALS-FRS=37.0; ALS-FRS-R=45.0]
1	26	2	3	[ALS-FRS=38.0; ALS-FRS-R=46.0], [ALS-FRS=37.0; ALS-FRS-R=45.0], [ALS-FRS=37.0; ALS-FRS-R=45.0]
2	13	1	3	[ALS-FRS-R=45.0], [ALS-FRS-R=45.0], [ALS-FRS-R=45.0]
3	13	1	2	[ALS-FRS=37.0], [ALS-FRS=37.0]
4	14	1	2	[ALS-FRS-R=45.0], [ALS-FRS-R=45.0]
31	58	2	3	[ALS-FRSb=12.0; ALS-FRSr=4.0], [ALS-FRSb=12.0; ALS-FRSr=4.0], [ALS-FRSb=12.0; ALS-FRSr=4.0]
46	51	3	3	[ALS-FRSb=12.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSb=12.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSb=12.0; ALS-FRSr=4.0; R=12.0]
47	51	2	3	[ALS-FRSb=12.0; R=12.0], [ALS-FRSb=12.0; R=12.0], [ALS-FRSb=12.0; R=12.0]
55	17	2	2	[ALS-FRSsUL=12.0; ALS-FRSr=4.0], [ALS-FRSsUL=12.0; ALS-FRSr=4.0]
60	14	2	2	[ALS-FRSsUL=12.0; R=12.0], [ALS-FRSsUL=12.0; R=12.0]
62	15	3	2	[ALS-FRSsUL=9.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSsUL=9.0; ALS-FRSr=4.0; R=12.0]
71	14	2	3	[ALS-FRSsLL=12.0; ALS-FRSr=4.0], [ALS-FRSsLL=12.0; ALS-FRSr=4.0], [ALS-FRSsLL=12.0; ALS-FRSr=4.0]
120	28	2	2	[ALS-FRS=37.0; ALS-FRS-R=45.0], [ALS-FRS=37.0; ALS-FRS-R=45.0]
Specialized Model for Neutral Progressors				
#Tricluster	$\|I\|$	$\|J\|$	$\|K\|$	Patterns
1	59	2	3	[ALS-FRS=33.0; ALS-FRS-R=41.0], [ALS-FRS=31.0; ALS-FRS-R=38.0], [ALS-FRS=29.0; ALS-FRS-R=37.0]
2	48	2	3	[ALS-FRS=33.0; ALS-FRS-R=41.0], [ALS-FRS=33.0; ALS-FRS-R=41.0], [ALS-FRS=31.0; ALS-FRS-R=39.0]
3	53	2	3	[ALS-FRS=35.0; ALS-FRS-R=43.0], [ALS-FRS=35.0; ALS-FRS-R=43.0], [ALS-FRS=33.0; ALS-FRS-R=41.0]
4	12	2	3	[ALS-FRS=38.0; ALS-FRS-R=46.0], [ALS-FRS=36.0; ALS-FRS-R=44.0], [ALS-FRS=37.0; ALS-FRS-R=45.0]
5	17	1	3	[ALS-FRS=37.0], [ALS-FRS=35.0], [ALS-FRS=35.0]
10	18	1	3	[ALS-FRS=37.0], [ALS-FRS=35.0], [ALS-FRS=35.0]
11	14	1	3	[ALS-FRS-R=45.0], [ALS-FRS-R=43.0], [ALS-FRS-R=43.0]
13	19	1	3	[ALS-FRS-R=46.0], [ALS-FRS-R=41.0], [ALS-FRS-R=41.0]
190	18	2	3	[ALS-FRSb=12.0; ALS-FRSr=4.0], [ALS-FRSb=12.0; ALS-FRSr=4.0], [ALS-FRSb=12.0; ALS-FRSr=4.0]
226	64	3	3	[ALS-FRSb=12.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSb=12.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSb=12.0; ALS-FRSr=4.0; R=12.0]
238	19	1	3	[ALS-FRSb=12.0], [ALS-FRSb=12.0], [ALS-FRSb=12.0]
244	67	2	3	[ALS-FRSb=12.0; R=12.0], [ALS-FRSb=12.0; R=12.0], [ALS-FRSb=12.0; R=12.0]
260	20	2	3	[ALS-FRSsUL=12.0; ALS-FRSsLL=12.0], [ALS-FRSsUL=12.0; ALS-FRSsLL=12.0], [ALS-FRSsUL=12.0; ALS-FRSsLL=12.0]
261	19	1	3	[ALS-FRSsUL=12.0], [ALS-FRSsUL=12.0], [ALS-FRSsUL=12.0]
291	19	3	3	[ALS-FRSsUL=12.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSsUL=12.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSsUL=12.0; ALS-FRSr=4.0; R=12.0]
342	20	2	3	[ALS-FRSsLL=12.0; R=12.0], [ALS-FRSsLL=12.0; R=12.0], [ALS-FRSsLL=12.0; R=12.0]
550	14	2	2	[FVC=87.3571; MEP=65.7071], [FVC=88.7429; MEP=66.9929]
551	20	2	2	[MIP=54.2; MEP=63.8], [MIP=58.355; MEP=66.5]
Specialized Model for Fast Progressors				
#Tricluster	$\|I\|$	$\|J\|$	$\|K\|$	Patterns
0	10	1	2	[R=12.0], [R=12.0]
1	8	1	3	[R=12.0], [R=12.0], [R=12.0]
2	8	1	3	[ALS-FRSr=4.0], [ALS-FRSr=4.0], [ALS-FRSr=3.0]

Learning Predictive Models Using a Mixture of Biclustering and Triclustering

5

In Chapter 3 we proposed a methodology based on triclustering for temporal and heterogeneous data analysis. However, clinical data is in general composed not only by temporal features but also by static features (demographics, medication, genetic information, habits, trauma/surgery information, etc), that remain unchanged over the different time points (Figure 5.1).

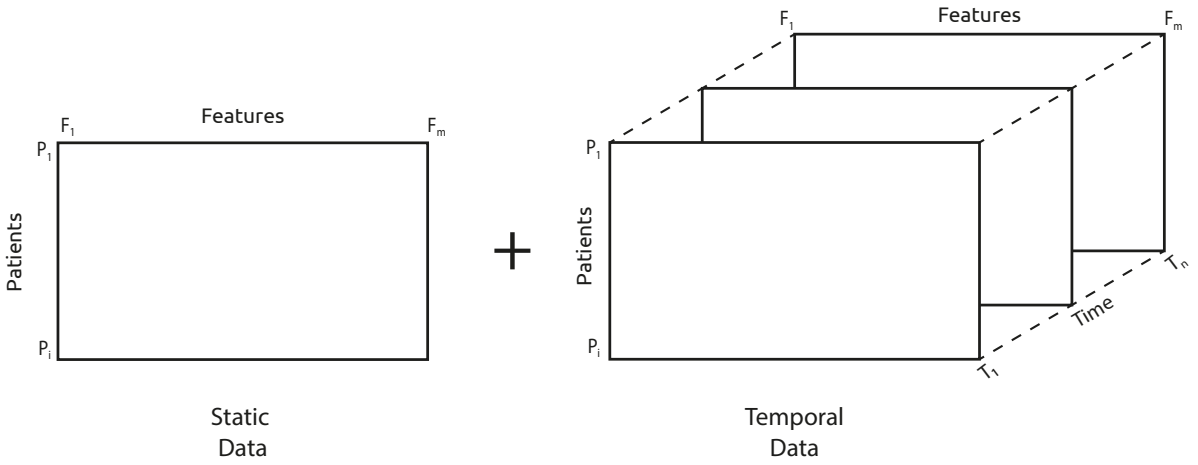


Figure 5.1: Example of heterogeneous dataset composed by static and temporal data.

In this context, this chapter proposes a new classification approach combining the previous triclustering-based approach with biclustering on static data. The idea is improve the performance of predictive model previously learnt with triclustering results, by enabling the use of patterns obtained when biclustering static data. The explainability of the model should also be improved by using static patterns (learnt with biclustering) together with the temporal patterns (learnt with triclustering).

5.1 Learning BicTric Classifier

Following the triclustering-based classifier in Chapter 3, we now propose BicTric to learn predictive models using a mixture of biclustering and triclustering, enabling to take advantage not only of temporal patterns but also of static patterns. We thus analyse static data using biclustering to explore subspaces in static features and use these biclusters to complement the learning data to be used by the classifier.

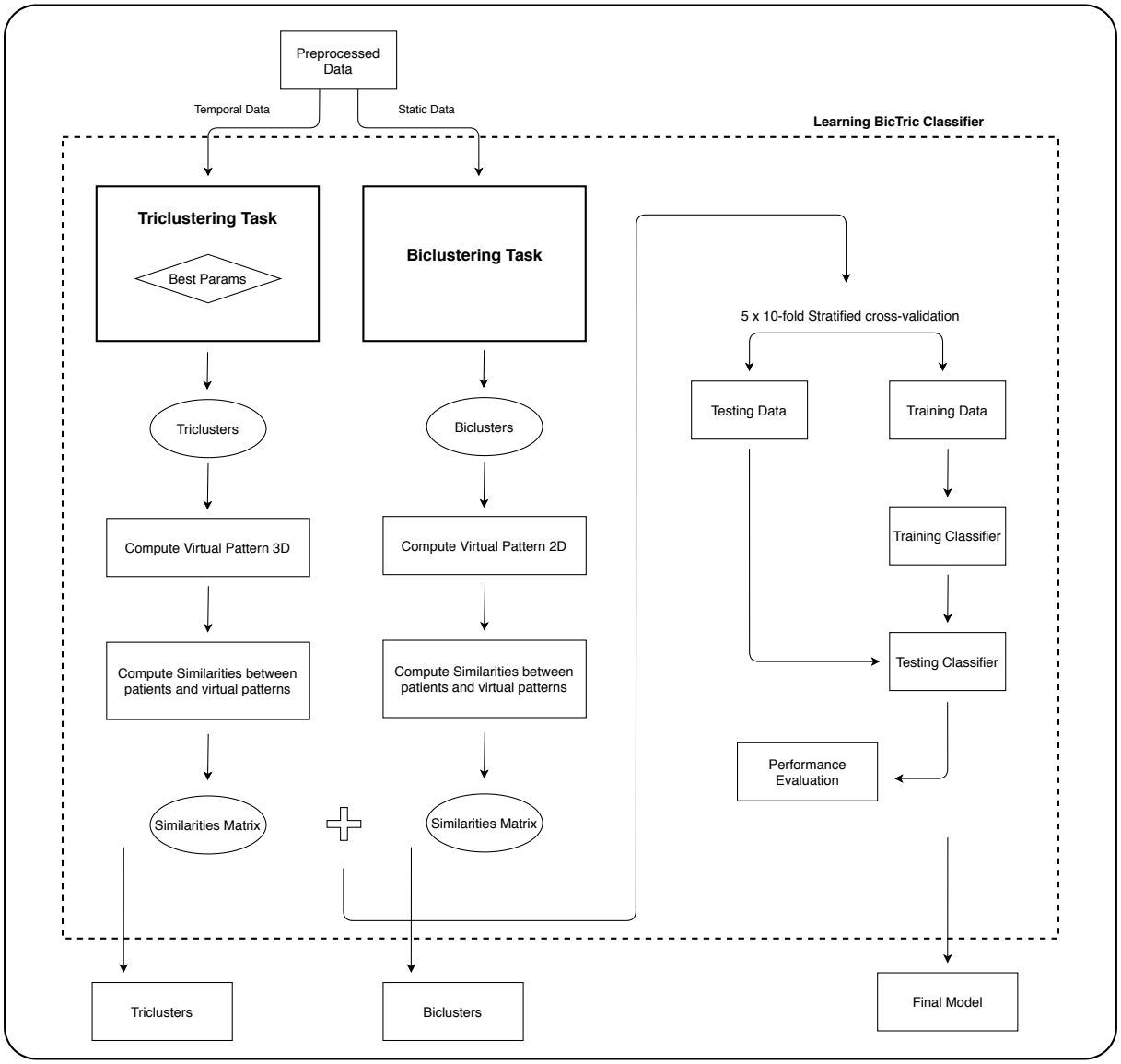


Figure 5.2: BicTric: Workflow.

The workflow of the new proposed methodology depicted in Figure 5.2 differs from the one presented in Figure 3.2 in the second step, where we couple a triclustering task with a biclustering task. We now find triclusters on temporal data and biclusters on static data, then used as features in the classifier.

In the biclustering task, we can use any biclustering algorithm able to deal with heterogeneous data. Since we used an extension of triCluster for triclustering, and taking advantage that this algorithm is biclustering-based, we used biCluster from triCluster to mine biclusters (see Chapter 3). We adapted biCluster to handle missing values. This enables to use data with missing values not only in biclustering but also in triclustering.

To build the similarity matrix to be used by classifier, the virtual pattern 2D is computed for each bicluster (similarly to what was performed for each tricluster). The virtual pattern 2D, defined in [13], is computed as follows:

Definition 4. (Virtual Pattern 2D). Given a bicluster $\mathcal{B}$, its virtual pattern $\mathcal{P}$ is defined as a set of elements $\mathcal{P} = \{\rho_1, \rho_2, ..., \rho_I\}$, where $\rho_i, 1 \leq i \leq I$ is defined as the mean (or the mode, in case of categorical features) of values in the i^{th}:

$$\rho_i = \frac{1}{J} \sum_{j=1}^{J} b_{ij}. \tag{5.1}$$

To assess how well a specific object (patient) follows the general tendency of a given bicluster $\mathcal{B}$, we proposed two approaches: (2) compute the Euclidean distance; or (3) compute Pearson correlation between the virtual pattern 2D $\mathcal{P}$ and the equivalent pattern (same features) of p_i.

We denoted (1) and (2) as Virtual Distance 2D and Virtual Correlation 2D, respectively, and defined them as follows:

Definition 5. (Virtual Distance 2D). The virtual distance between an observation p_i and a bicluster $\mathcal{B}$ is

$$\texttt{VD}_{\texttt{2D}}(p_i, \mathcal{B}) = E(p_i, \rho) = \sqrt{\sum_{e=1}^{I} (p_{i_e} - \rho_e)^2}. \tag{5.2}$$

Definition 6. (Virtual Correlation 2D). The virtual correlation between an object p_i and a bicluster $\mathcal{B}$ is

$$\texttt{VC}_{\texttt{2D}}(p_i, \mathcal{B}) = r(p_i, \rho) = \frac{\sum_{e=1}^{I} (p_{i_e} - \bar{p_i})(\rho_e - \bar{\rho})}{\sqrt{\sum_{e=1}^{I} (p_{i_e} - \bar{p_i})^2 \sum_{e=1}^{I} (\rho_e - \bar{\rho})^2}}. \tag{5.3}$$

The triclustering task is performed as explained in Chapter 3, and the BicTric classifier is then learnt from learning examples using a mixture of biclusters and triclusters as features.

5.2 Learning Predictive Models using BicTric in the ALS Case Study

We used again the Lisbon ALS dataset containing Electronic Health Records from ALS Patients regularly followed at the local ALS clinic, since 1995 and last updated in March 2020. Its current version contains 1374 patients. For these experiments with BicTric we use not only the temporal features but also the following static features: Gender, Body Mass Index (BMI), MND familiar history, Age at onset, Disease duration, El Escorial reviewed criteria, UMN vs LMN, Onset form, C9orf72. Data was preprocessed as explained in Chapter 4. Since both biclustering and triclustering tasks can deal with missing values we did not perform any imputations. In this context, following what we did in Chapter 4, we performed the following experiments:

1. Random Forests with original features (static and temporal),
2. Random Forests with original static features and triclusters,
3. BicTric classification,
4. BicTric classification with patient stratification.

5.2.1 Baseline Results: Random Forests with original features (static and temporal)

The baseline results, to be compared with the predictive models learned with BicTric classifiers, were obtained by training a Random Forest with the original features (static and temporal) treated as independent features (number of CS $\times$ 10 temporal features + 8 static features). Table 5.1 shows the baseline results, which are slightly worst than those obtained using just the temporal features (Table 4.2). Apparently, static data (individually) do not introduce relevant performance improvements when the model is trained with original features. Moreover, we can observe that Sensitivity and Specificity have more variance than when only temporal features were considered, meaning that static data are introducing an unbalance in both classes prediction.

Table 5.1: Baseline Results using Random Forests and Original Features (static and temporal)

CS	AUC	Accuracy	Sensitivity	Specificity
3	0.79 ± 0.0044	0.72 ± 0.0035	0.77 ± 0.0025	0.61 ± 0.0067
4	0.81 ± 0.0033	0.75 ± 0.0070	0.78 ± 0.0092	0.67 ± 0.0104
5	0.79 ± 0.0074	0.71 ± 0.0065	0.76 ± 0.0080	0.58 ± 0.0106

5.2.2 Random Forests with Original Static Features and Triclusters

To further assess the relevance of static features in the performance of classifiers in this case study, we added the original static features to the similarity matrices used in the triclustering-based models presented in Chapter 4. We observed (Table 5.2) that in fact this experiment outperforms the baseline with original static and temporal features. Furthermore, comparing with the results presented in Section 4.2.2 we can see a slight performance improvement, pinpointing that static features are contributing to the model. This could be confirmed by inspecting the feature ranking used by random forests, which shows static features in the top 10 of the best-used features. Despite this performance improvement when using the original static features together triclusters, we show next that the use of biclustering in the static data can add further improvements not only in the performance of the models but also in the interpretability of the patterns of features that can be found in the static data.

Table 5.2: Performance Evaluation Results of Random Forests with original static features and triclusters.

AP	CS	AUC	Accuracy	Sensitivity	Specificity
D	3	0.84 ± 0.0421	0.75 ± 0.0435	0.77 ± 0.0637	0.73 ± 0.0590
	4	0.88 ± 0.0443	0.79 ± 0.0484	0.81 ± 0.0697	0.78 ± 0.0805
	5	0.85 ± 0.0535	0.75 ± 0.0545	0.80 ± 0.0845	0.72 ± 0.0999
C	3	0.84 ± 0.0391	0.74 ± 0.0369	0.74 ± 0.0590	0.74 ± 0.0592
	4	0.87 ± 0.0443	0.73 ± 0.0576	0.74 ± 0.0779	0.75 ± 0.0957
	5	0.85 ± 0.0524	0.75 ± 0.0572	0.77 ± 0.0903	0.72 ± 0.0988

5.2.3 BicTric Classification

To test the proposed BicTric classifier on the ALS case study, allowing fair comparisons with the baseline, we also used Random Forests as classifier.

For these experiments, we followed the same steps as explained in Chapter 4, using the same triclusters obtained with the best parameters described in Table 4.4. We performed biclustering in static features, using biCluster algorithm from triCluster [47], and computed the similarities between patients and biclusters using two approaches: distance (D) and correlation (C).

Table 5.3 shows the performance results obtained by BicTric classifier. We can observe that, overall, the results exceed the baseline, meaning biclustering and triclustering features improve the performance results of Random Forest learnt with original features. Moreover, we can see that Sensitivity and Specificity are now more balanced and thus the classifier performs well on predicting both classes, contrary to what happened in the baseline.

Although these results exceeded the baseline results, when compared with the results obtained by triclustering-based classifier (Table 4.7), the results are similar. However, there is improvement in Sensitivity and Specificity balancing.

Table 5.3: Performance Evaluation Results of BicTric Classifier.

AP	CS	AUC	Accuracy	Sensitivity	Specificity
D	3	0.85 ± 0.0348	**0.76 ± 0.0407**	0.78 ± 0.0588	0.74 ± 0.0658
	4	0.88 ± 0.0367	0.78 ± 0.0403	0.79 ± 0.0704	0.76 ± 0.0747
	5	0.82 ± 0.0016	**0.76 ± 0.0048**	0.76 ± 0.0088	0.75 ± 0.0044
C	3	0.84 ± 0.0404	0.74 ± 0.0468	0.73 ± 0.0696	0.76 ± 0.0592
	4	0.86 ± 0.0450	**0.75 ± 0.0493**	0.75 ± 0.0912	0.76 ± 0.0855
	5	0.85 ± 0.0019	**0.75 ± 0.0086**	0.74 ± 0.0146	0.75 ± 0.0049

5.2.4 Results with Patient Stratification

As in Chapter 4, we also used data from patients stratified in Slow, Neutral and Fast progressors and used BicTric to learn a specialized predictive model for each group of patients. Table 5.4 shows the results, obtained in the same way as those for the model learned with all data, as baseline. Similar to what happened when considering only the temporal features, the results after patient stratification outperform the results obtained when learning from all patients, since patients are now more homogeneous. The small number of patients with 5 CS in the Fast group prevented us from obtaining reliable results in this case.

Table 5.5 presents the performance results obtained with BicTric specialized models for each progression group according to the considered number of consecutive snapshots considered. We can see that,

Table 5.4: Baseline Results: Random Forests with Original Features (temporal + static) per Disease Progression Group.

CS	Group	AUC	Accuracy	Sensitivity	Specificity
3	Slow	0.82 ± 0.0069	0.75 ± 0.0187	0.79 ± 0.0139	0.64 ± 0.0319
	Neutral	0.76 ± 0.0043	0.71 ± 0.0018	0.75 ± 0.0027	0.60 ± 0.0057
	Fast	0.57 ± 0.0175	0.59 ± 0.0199	0.64 ± 0.0102	0.08 ± 0.1118
4	Slow	0.85 ± 0.0065	0.77 ± 0.0211	0.82 ± 0.0175	0.67 ± 0.0314
	Neutral	0.75 ± 0.0050	0.71 ± 0.0081	0.75 ± 0.0073	0.60 ± 0.0141
	Fast	0.69 ± 0.0170	0.66 ± 0.0285	0.68 ± 0.0165	0.57 ± 0.0748
5	Slow	0.84 ± 0.0047	0.77 ± 0.0203	0.81 ± 0.0166	0.67 ± 0.0318
	Neutral	0.69 ± 0.0048	0.68 ± 0.0100	0.72 ± 0.0075	0.52 ± 0.0225
	Fast				

overall, these results outperform baseline, improving not only classification accuracy and AUC but also the performance on predicting both classes (verified by similar values of Sensitivity and Specificity). The expressive improvements are in the fast progressors group, the group with the lower number of patients, whose results benefited from the increasing of features, resulting from biclustering and triclustering. These results show that using stratified data and considering static and temporal features, applying BicTric methodology can be better than learning the classifier with the original features, even when using a powerful classifier as Random Forests, that performs feature selection by itself.

5.2.5 Model Interpretability

Similarly to Chapter 4, besides the analysis of performance results in classification, we are interested in analyzing model interpretability by studying the patterns discovered by biclustering and triclustering and highlighted by the classifiers. To this aim, we chose to analyse the patterns discovered when applying BicTric in the 3 CS dataset. The goal is to identify the most relevant features, what features appear together, and whether the temporal patterns found to be relevant in the general model (putative patterns of the average patient) differ from those relevant to the specialized models (group-specific patterns highlighting disease progression patterns of Slow, Neutral and Fast progressors). Tables 5.6, 5.7, 5.8 and 5.9 depict the most important patterns used by the classifier, ranked by their feature importance.

As in the case of triclustering-based classifier (Chapter 4), an overall analysis of the most important patterns discovered shows that the majority of the patterns refer to the last snapshot/time-point of the

Table 5.5: Performance Evaluation Results obtained with BicTric Classifier (Specialized Models for each Disease Progression Group).

AP	CS	Group	AUC	Accuracy	Sensitivity	Specificity
D	3	Slow	0.87 ± 0.0461	0.76 ± 0.0650	0.78 ± 0.1075	0.74 ± 0.0857
		Neutral	0.84 ± 0.0390	0.75 ± 0.0430	0.72 ± 0.0609	0.79 ± 0.0639
		Fast	0.78 ± 0.0877	0.72 ± 0.0883	0.71 ± 0.1969	0.73 ± 0.1365
	4	Slow	0.85 ± 0.0361	0.76 ± 0.0604	0.78 ± 0.1027	0.73 ± 0.0776
		Neutral	0.83 ± 0.0391	0.74 ± 0.0436	0.73 ± 0.0611	0.76 ± 0.0684
		Fast	0.86 ± 0.1102	0.77 ± 0.1265	0.77 ± 0.1828	0.77 ± 0.1896
	5	Slow	0.89 ± 0.0422	0.81 ± 0.0526	0.83 ± 0.0660	0.79 ± 0.0771
		Neutral	0.80 ± 0.0570	0.73 ± 0.0554	0.70 ± 0.1075	0.76 ± 0.0564
		Fast				
C	3	Slow	0.86 ± 0.0513	0.76 ± 0.0713	0.77 ± 0.1094	0.76 ± 0.0911
		Neutral	0.83 ± 0.0452	0.74 ± 0.0496	0.70 ± 0.0879	0.78 ± 0.0510
		Fast	0.78 ± 0.0858	0.71 ± 0.1018	0.69 ± 0.1886	0.73 ± 0.1398
	4	Slow	0.86 ± 0.0334	0.76 ± 0.0499	0.77 ± 0.0990	0.75 ± 0.0793
		Neutral	0.84 ± 0.0370	0.75 ± 0.0386	0.72 ± 0.0666	0.77 ± 0.0649
		Fast	0.89 ± 0.0963	0.80 ± 0.1146	0.78 ± 0.1794	0.81 ± 0.1905
	5	Slow	0.89 ± 0.0396	0.80 ± 0.0628	0.81 ± 0.0663	0.78 ± 0.0950
		Neutral	0.81 ± 0.0557	0.72 ± 0.0566	0.67 ± 0.0972	0.77 ± 0.0651
		Fast				

tricluster. This is understandable as the last snapshots are closer to the horizon of prediction. However, other patterns, not corresponding to the last snapshot, and in this case, including static patterns, also remained important. These patterns are also relevant for further clinical analysis, since these features can prove to be important to identify disease progression patterns leading to the need of NIV for a given time-window.

Neutral patients is the group benefiting more from considering static patterns. This can actually make sense, since these patients are the most common type of patient, whose evolution does not suffer from temporal issues. Fast progressors, the minority class in the dataset, have more advantages when static features are used to complement the short time analysis that can be made with temporal data with more valuable information.

5.3 Final Remarks

This Chapter presented the BicTric approach to learn predictive models based on a mixture of biclustering and triclustering. The results are very promising, showing the impact that the biclusters mined from static data can have in the models, when compared with the results in Chapter 4 using only temporal data and triclustering.

Table 5.6: 20 Top patterns discovered by the general BicTric model trained with all data (with 3 CS)

Rank	#Tricluster	TP	Feat. Import	Pattern
1	29	2	0.010270	[ALS-FRS-R = 45.0]
2	7	2	0.010214	[ALS-FRS = 37.0]
3	4	2	0.009750	[ALS-FRS = 38.0]
4	45	2	0.009580	[ALS-FRS = 38.0; ALS-FRS-R = 46.0]
5	1	2	0.009246	[ALS-FRS = 31.0; ALS-FRS-R = 39.0]
6	693	2	0.008568	[ALS-FRSb = 12.0; ALS-FRSsUL = 11.0]
7	70	2	0.007903	[ALS-FRS = 36.0]
8	115	2	0.007147	[ALS-FRS-R = 45.0]
9	2	2	0.006982	[ALS-FRS = 37.0]
10	20	2	0.006513	[ALS-FRS = 36.0]
11	6043	1	0.006133	[ALS-FRS = 21.0; ALS-FRSsUL = 11.0]
12	56	2	0.005767	[ALS-FRS = 37.0]
13	428	1	0.005476	[ALS-FRS-R = 23.0; ALS-FRSsUL = 11.0]
14	106	2	0.005042	[ALS-FRS-R = 43.0]
15	878	2	0.004823	[ALS-FRSb = 12.0; ALS-FRSsLL = 12.0]
16	184	2	0.004744	[ALS-FRS = 37.0]
17	118	2	0.004324	[ALS-FRS-R = 43.0]
18	739	2	0.004233	[ALS-FRSb = 12.0; ALS-FRSsUL = 11.0; ALS-FRSr = 4.0; R = 12.0]
19	32	2	0.004115	[ALS-FRS-R = 43.0]
20	45	1	0.004026	[ALS-FRS = 38.0; ALS-FRS-R = 46.0]

Table 5.7: 20 Top patterns used by the specialized BicTric model for Slow group (with 3 CS)

Rank	#Tricluster	TP	Feat. Import	Pattern
1	1	2	0.043831	[ALS-FRS = 38.0; ALS-FRS-R = 46.0]
2	36	2	0.027715	[ALS-FRSb = 12.0; ALS-FRSr = 4.0]
3	101	2	0.026623	[ALS-FRSsUL = 12.0; R = 12.0]
4	71	2	0.024176	[ALS-FRSb = 12.0; ALS-FRSr = 4.0; R = 12.0]
5	78	2	0.022699	[ALS-FRSb = 12.0; R = 12.0]
6	14	2	0.018968	[ALS-FRSb = 12.0; ALS-FRSsUL = 12.0; ALS-FRSr = 4.0]
7	99	2	0.017771	[ALS-FRSsUL = 12.0; ALS-FRSr = 4.0; R = 12.0]
8	1	1	0.017201	[ALS-FRS = 38.0; ALS-FRS-R = 46.0]
9	9	2	0.017198	[ALS-FRSb = 12.0; ALS-FRSsUL = 12.0]
10	0	1	0.016896	[ALS-FRS = 38.0; ALS-FRS-R = 46.0]
11	91	2	0.015057	[ALS-FRSsUL = 12.0; ALS-FRSr = 4.0]
12	232	1	0.014542	[ALS-FRSsUL = 9.0; ALS-FRSr = 4.0]
13	1	0	0.013147	[ALS-FRS = 38.0; ALS-FRS-R = 46.0]
14	16	1	0.012980	[ALS-FRSb = 12.0; ALS-FRSsUL = 12.0; ALS-FRSr = 4.0; R = 12.0]
15	89	1	0.012738	[ALS-FRSsUL = 12.0; ALS-FRSr = 4.0]
16	115	1	0.011652	[ALS-FRSsLL = 12.0; ALS-FRSr = 4.0]
17	0	0	0.011566	[ALS-FRS = 38.0; ALS-FRS-R = 46.0]
18	98	1	0.011276	[ALS-FRSsUL = 12.0; ALS-FRSr = 4.0; R = 12.0]
19	192	2	0.011155	[ALS-FRSr = 3.0; R = 11.0]
20	100	1	0.010942	[ALS-FRSsUL = 12.0; R = 12.0]

Table 5.8: 20 Top patterns used by the specialized BicTric model for Neutral group (with 3 CS)

Rank	#Tricluster	TP	Feat. Import	Pattern
1	4	2	0.011258	[ALS-FRS = 35.0; ALS-FRS-R = 43.0]
2	430	2	0.006566	[ALS-FRSsUL = 12.0; ALS-FRSr = 4.0; R = 12.0]
3	1	2	0.006397	[ALS-FRS = 31.0; ALS-FRS-R = 39.0]
4	431	2	0.005939	[ALS-FRSsUL = 12.0; R = 12.0]
5	115	2	0.005829	[ALS-FRSsUL = 11.0]
6	450	2	0.005619	[ALS-FRSsUL = 11.0; R = 12.0]
7	3	2	0.005417	[ALS-FRS = 33.0; ALS-FRS-R = 41.0]
8	1	1	0.005299	[ALS-FRS = 33.0; ALS-FRS-R = 41.0]
9	95	*B*	0.005261	[Gen = 2.0; Age_onset = 60.0]
10	216	*B*	0.004814	[MND_fh = 2.0; Disea_dur = 12.0; Onset_f = 1.0]
11	2	2	0.004757	[ALS-FRS = 27.0; ALS-FRS-R = 35.0]
12	3	1	0.004706	[ALS-FRS = 33.0; ALS-FRS-R = 41.0]
13	415	2	0.004666	[ALS-FRSsUL = 12.0]
14	4	1	0.004604	[ALS-FRS = 37.0; ALS-FRS-R = 45.0]
15	292	*B*	0.004463	[Disea_dur = 12.0; UMN_LMN = 1.0]
16	249	2	0.004420	[ALS-FRSr = 3.0]
17	16	1	0.004416	[ALS-FRS = 34.0]
18	865	1	0.004403	[ALS-FRS = 33.0; ALS-FRS-R = 41.0]
19	306	2	0.004378	[ALS-FRSb = 12.0; ALS-FRSr = 4.0; R = 12.0]
20	453	2	0.004369	[ALS-FRSsUL = 11.0; ALS-FRSr = 4.0]

Table 5.9: 20 Top patterns used by the specialized BicTric model for Fast group (with 3 CS)

Rank	#Tricluster	TP	Feat. Import	Pattern
1	0	0	0.200922	[ALS-FRSr = 4.0]
2	0	1	0.079393	[ALS-FRSr = 4.0]
3	58	*B*	0.041575	[MND_fh = 2.0; UMN_LMN = 2.0; Onset_f = 1.0]
4	68	*B*	0.027542	[UMN_LMN = 2.0; Onset_f = 1.0908]
5	44	*B*	0.026792	[MND_fh = 2.0; EErC = 5.0; UMN_LMN = 1.0; C9orf72 = 2.0]
6	65	*B*	0.026326	[EErC = 2.0; UMN_LMN = 2.0]
7	12	*B*	0.024469	[Gen = 1.0; MND_fh = 2.0; Onset_f = 1.0; C9orf72 = 1.0]
8	32	*B*	0.024212	[Gen = 1.0; UMN_LMN = 2.0; Onset_f = 1.133]
9	2	*B*	0.022883	[Gen = 1.0; MND_fh = 2.0; EErC = 5.0; UMN_LMN = 1.0]
10	4	*B*	0.022584	[Gen = 1.0; MND_fh = 2.0; EErC = 5.0; UMN_LMN = 1.0; C9orf72 = 2.0]
11	43	*B*	0.019859	[MND_fh = 2.0; EErC = 5.0; UMN_LMN = 1.0; Onset_f = 2.0; C9orf72 = 2.0]
12	1	*B*	0.019478	[Gen = 1.0; MND_fh = 2.0; EErC = 5.0]
13	3	*B*	0.016088	[Gen = 1.0; MND_fh = 2.0; EErC = 5.0; UMN_LMN = 1.0; Onset_f = 1.0]
14	41	*B*	0.015916	[MND_fh = 2.0; EErC = 5.0; UMN_LMN = 1.0; Onset_f = 1.0; C9orf72 = 2.0]
15	69	*B*	0.015903	[UMN_LMN = 2.0; C9orf72 = 1.534]
16	34	*B*	0.015156	[Gen = 1.0; Onset_f = 1.4546; C9orf72 = 1.4546]
17	6	*B*	0.014366	[Gen = 1.0; MND_fh = 2.0; EErC = 5.0; C9orf72 = 2.0]
18	45	*B*	0.013959	[MND_fh = 2.0; EErC = 5.0; Onset_f = 1.0]
19	46	*B*	0.013859	[MND_fh = 2.0; EErC = 5.0; Onset_f = 1.0; C9orf72 = 2.0]
20	17	*B*	0.012948	[Gen = 2.0; MND_fh = 2.0; EErC = 5.0]

6 Conclusions and Future Work

We proposed a new methodology to learn predictive models using a triclustering-based classifier. It has three phases: Preprocessing Data, Learning Triclustering Best Parameters and Learning Triclustering-based Classifier. The key step uses TCtriCluster, an extension of triCluster, incorporating a temporal constraint when mining triclusters. This restriction was shown to be effective in improving the effectiveness of the predictive models, highlighting its importance when triclustering temporal data. We further show that triclustering-based classification enhances prediction with the potentialities of model interpretability, enabling the discovery of domain relevant temporal patterns, then used as features in the models.

As case study we used clinical three-way data and tackled the challenge of predicting the need for NIV in ALS patients within a time window of 90 days. We performed experiments using all patients and patients stratified by their disease progression rate as Slow, Neutral and Fast progressors.

The prognostic prediction results are promising, in particular when patient stratification is performed, specially for Neutral progressors. Concerning model interpretation, it was interesting to confirm the existence of group-specific patterns, corresponding to different disease progression patterns, then used by the specialized models as important features. In particular, Fast progressors have unique patterns, whose quick identification could help to improve prognosis, by anticipating NIV.

We further proposed a new methodology combining biclustering and triclustering, BicTric, able to analyse not only temporal data but also static data, by using biclusters and triclusters as features.

The prognostic prediction models learned by BicTric in the ALS case study outperformed the previous results and baselines, meaning that in this case the application of this methodology can achieve higher accuracies in ALS data.

The proposed triclustering-based methodologies can further be used to learn predictive models with

different types of three-way-data, from other heterogeneous diseases or other domains, with the particularity that in these methodologies it is possible to use any biclustering/triclustering algorithm.

This work will be followed up in a PhD thesis that intends to tackle the problems identified during this research together with other challenges of triclustering in temporal three-way data and biomedical applications. More specifically, we plan to explore different types of algorithmic approaches to develop problem-specific triclustering algorithms, targeting specific biomedical data analysis problems.

References

[1] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010. 15

[2] D. Amar, D. Yekutieli, A. Maron-Katz, T. Hendler, and R. Shamir. A hierarchical bayesian model for flexible module discovery in three-way time-series data. *Bioinformatics*, 31(12):i17–i26, 2015. 1, 19, 20, 22

[3] S. A. Andersena, G. D. Borasioc, M. de Carvalho, A. Chioe, P. Van Dammef, O. Hardimang, K. Kolleweh, K. E. Morrisoni, et al. Efns guidelines on the clinical management of amyotrophic lateral sclerosis (mals)–revised report of an efns task force. *European Journal of Neurology*, 19: 360–375, 2011. 2, 4

[4] A. Bhar, M. Haubrock, A. Mukhopadhyay, and E. Wingender. Application of a novel triclustering method (δ-trimax) to mine 3d gene expression data of breast cancer cells. *GCB 2013 Göttingen-Highlight Papers*, 2013. 19, 20, 22

[5] S. C. Bourke, M. Tomlinson, T. L. Williams, R. E. Bullock, P. J. Shaw, and G. J. Gibson. Effects of non-invasive ventilation on survival and quality of life in patients with amyotrophic lateral sclerosis: a randomised controlled trial. *The Lancet Neurology*, 5(2):140–147, 2006. 2, 4

[6] A. V. Carreiro, O. Anunciação, J. A. Carriço, and S. C. Madeira. Prognostic prediction through biclustering-based classification of clinical gene expression time series. *Journal of integrative bioinformatics*, 8(3):73–89, 2011. 10, 21, 27

[7] A. V. Carreiro, A. J. Ferreira, M. A. Figueiredo, and S. C. Madeira. Towards a classification approach using meta-biclustering: impact of discretization in the analysis of expression time series. *Journal of integrative bioinformatics*, 9(3):105–120, 2012. 21

[8] A. V. Carreiro, P. M. Amaral, S. Pinto, P. Tomás, M. de Carvalho, and S. C. Madeira. Prognostic models based on patient snapshots and time windows: Predicting disease progression to assisted

ventilation in amyotrophic lateral sclerosis. *Journal of biomedical informatics*, 58:133–144, 2015. 2, 23, 35

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. 10, 36

[10] A. Chiò, G. Logroscino, B. Traynor, J. Collins, J. Simeone, L. Goldstein, and L. White. Global epidemiology of amyotrophic lateral sclerosis: a systematic review of the published literature. *Neuroepidemiology*, 41(2):118–130, 2013. 2, 3

[11] B. Conde, J. C. Winck, and L. F. Azevedo. Estimating amyotrophic lateral sclerosis and motor neuron disease prevalence in portugal using a pharmaco-epidemiological approach and a bayesian multiparameter evidence synthesis model. *Neuroepidemiology*, 53(1-2):73–83, 2019. 2, 3

[12] F. O. de França, G. P. Coelho, and F. J. Von Zuben. Predicting missing values with biclustering: A coherence-based approach. *Pattern Recognition*, 46(5):1255–1266, 2013. 9

[13] F. Divina, B. Pontes, R. Giráldez, and J. S. Aguilar-Ruiz. An effective measure for assessing the quality of biclusters. *Computers in biology and medicine*, 42(2):245–256, 2012. 28, 53

[14] ENCALS. Als functional rating scale revised (als-frs-r). version: May 2015. https://www.encals.eu/wp-content/uploads/2016/09/ALS-Functional-Rating-Scale-Revised-fill-in-form.pdf, 2015. [Online]. 4

[15] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282, 2010. 8

[16] F. Gerber, R. de Jong, M. E. Schaepman, G. Schaepman-Strub, and R. Furrer. Predicting missing values in spatio-temporal remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 56(5):2841–2853, 2018. 9

[17] D. Gutierrez-Aviles and C. Rubio-Escudero. Lsl: A new measure to evaluate triclusters. In *2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 30–37. IEEE, 2014. 19, 21, 22

[18] D. Gutiérrez-Avilés and C. Rubio-Escudero. Msl: a measure to evaluate three-dimensional patterns in gene expression data. *Evolutionary Bioinformatics*, 11:EBO–S25822, 2015. 19, 20, 21, 22

[19] D. Gutiérrez-Avilés, C. Rubio-Escudero, and J. Riquelme. Unravelling the yeast cell cycle using the trigen algorithm. 7023:155–163, 11 2011. doi: 10.1007/978-3-642-25274-7_16. 19

[20] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Tecniques: 3rd edition*. Morgan Kaufmann Publishers, 2012. 8, 9, 10, 11, 12, 13, 14

[21] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009. 9

[22] C. Heffernan, C. Jenkinson, T. Holmes, H. Macleod, W. Kinnear, D. Oliver, N. Leigh, and M. Ampong. Management of respiration in mnd/als patients: An evidence based review. *Amyotrophic Lateral Sclerosis*, 7(1):5–15, 2006. 2, 3

[23] R. Henriques and S. C. Madeira. Bicpam: Pattern-based biclustering for biomedical data analysis. *Algorithms for Molecular Biology*, 9(1):27, 2014. 21

[24] R. Henriques and S. C. Madeira. Triclustering algorithms for three-dimensional data analysis: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 51(5):95, 2019. 1, 7, 8, 15, 16, 18, 19, 20, 26

[25] H. Jiang, S. Zhou, J. Guan, and Y. Zheng. gtricluster: a more general and effective 3d clustering algorithm for gene-sample-time microarray data. In *International Workshop on Data Mining for Biomedical Applications*, pages 48–59. Springer, 2006. 19, 22

[26] T. Kakati, H. A. Ahmed, D. K. Bhattacharyya, and J. K. Kalita. Thd-tricluster: A robust triclustering technique and its application in condition specific change analysis in hiv-1 progression data. *Computational Biology and Chemistry*, 75:154 – 167, 2018. ISSN 1476-9271. doi: https://doi.org/10.1016/j.compbiolchem.2018.05.007. URL http://www.sciencedirect.com/science/article/pii/S1476927115302243. 19, 21, 22

[27] T. Kakati, H. A. Ahmed, D. K. Bhattacharyya, and J. K. Kalita. Thd-tricluster: A robust triclustering technique and its application in condition specific change analysis in hiv-1 progression data. *Computational biology and chemistry*, 75:154–167, 2018. 1

[28] P. Kaur and A. Gosain. Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise. In *ICT Based Innovations*, pages 23–30. Springer, 2018. 10

[29] A. Li and D. Tuck. An effective tri-clustering algorithm combining expression data with gene regulation information. *Gene regulation and systems biology*, 3:GRSB–S1150, 2009. 19, 20, 22

[30] X. Li, Y. Ye, M. Ng, and Q. Wu. Multifactv: module detection from higher-order time series biological data. *BMC genomics*, 14(4):S2, 2013. 19, 21, 22

[31] Y. Li and A. Ngom. Classification of clinical gene-sample-time microarray expression data via tensor decomposition methods. In *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 275–286. Springer, 2010. 1

[32] R. J. Little. Rd. statistical analysis with missing data. *Statistics. WsiPa, editor. New York2002*, 2002. 8

[33] J. Liu, Z. Li, X. Hu, and Y. Chen. Multi-objective evolutionary algorithm for mining 3d clusters in gene-sample-time microarray data. In *2008 IEEE International Conference on Granular Computing*, pages 442–447. IEEE, 2008. 19, 22

[34] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts. Understanding variable importances in forests of randomized trees. In *Advances in neural information processing systems*, pages 431–439, 2013. 15

[35] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(1):24–45, 2004. 1, 7, 15, 26

[36] S. C. Madeira, M. C. Teixeira, I. Sa-Correia, and A. L. Oliveira. Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(1):153–165, 2008. 27

[37] J. Matos, S. Pires, H. Aidos, M. Gromicho, S. Pinto, M. de Carvalho, and S. C. Madeira. Unravelling disease presentation patterns in als using biclustering for discriminative meta-features discovery. In *International Work-Conference on Bioinformatics and Biomedical Engineering*, pages 517–528. Springer, 2020. 2, 10, 21, 23

[38] C. D. Newgard and R. J. Lewis. Missing data: how to best account for what is not known. *Jama*, 314(9):940–941, 2015. 8, 9

[39] S. Pires, M. Gromicho, S. Pinto, M. Carvalho, and S. C. Madeira. Predicting non-invasive ventilation in als patients using stratified disease progression groups. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 748–757. IEEE, 2018. 2, 3, 4, 10, 23, 27, 35, 39, 40

[40] S. Pires, M. Gromicho, S. Pinto, M. de Carvalho, and S. C. Madeira. Patient stratification using clinical and patient profiles: Targeting personalized prognostic prediction in als. In *International Work-Conference on Bioinformatics and Biomedical Engineering*, pages 529–541. Springer, 2020. 2, 23

[41] J. A. Sáez, B. Krawczyk, and M. Woźniak. Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition*, 57:164–178, 2016. 10

[42] F. M. Shrive, H. Stuart, H. Quan, and W. A. Ghali. Dealing with missing data in a multi-question depression scale: a comparison of imputation methods. *BMC medical research methodology*, 6(1): 57, 2006. 9

[43] D. Soares, R. Henriques, M. Gromicho, S. Pinto, M. de Carvalho, and S. C. Madeira. Towards triclustering-based classification of three-way clinical data: A case study on predicting non-invasive ventilation in als. In *International Conference on Practical Applications of Computational Biology & Bioinformatics*, pages 112–122. Springer, 2020. 3

[44] A. B. Tchagang, S. Phan, F. Famili, H. Shearer, P. Fobert, Y. Huang, J. Zou, D. Huang, A. Cutler, Z. Liu, et al. Mining biological information from 3d short time-series gene expression data: the optricluster algorithm. *BMC bioinformatics*, 13(1):54, 2012. 19, 22

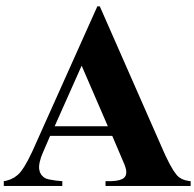
[45] G. Wang, L. Yin, Y. Zhao, and K. Mao. Efficiently mining time-delayed gene expression patterns. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(2):400–411, 2009. 19, 22

[46] L. Zhao and M. J. Zaki. Tricluster: An effective algorithm for mining coherent clusters in 3d microarray data. pages 694–705, 2005. doi: 10.1145/1066157.1066236. URL http://doi.acm.org/10.1145/1066157.1066236. 19, 22

[47] L. Zhao and M. J. Zaki. Tricluster: An effective algorithm for mining coherent clusters in 3d microarray data. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, pages 694–705, New York, NY, USA, 2005. ACM. 3, 27, 38, 56

[48] T. Zhu, Y. Lin, and Y. Liu. Synthetic minority oversampling technique for multiclass imbalance problems. *Pattern Recognition*, 72:327–340, 2017. 10

Scientific Paper: PACBB2020



Published Papper in proceedings of 14th International Conference on Practical Applications of Computational Biology and Bioinformatics (PACBB2020) - starts on next page.

Towards Triclustering-Based Classification of Three-Way Clinical Data: A Case Study on Predicting Non-invasive Ventilation in ALS

Diogo Soares[1(✉)], Rui Henriques[2], Marta Gromicho[3], Susana Pinto[3], Mamede de Carvalho[3], and Sara C. Madeira[1]

[1] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Lisbon, Portugal
{dfsoares,sacmadeira}@ciencias.ulisboa.pt
[2] INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal
rmch@tecnico.ulisboa.pt
[3] Instituto de Medicina Molecular, Instituto de Fisiologia, Faculdade de Medicina, Universidade de Lisboa, Lisbon, Portugal
martalgms@gmail.com,
{susana.c.pinto,mamedealves}@medicina.ulisboa.pt

Abstract. The importance to learn disease progression patterns from longitudinal clinical data and use them effectively to improve prognosis, triggers the need for new approaches for three-way data analysis. In this context, triclustering has been widely researched for its potential in biomedical problems, showing promising results in the discovery of putative biological modules, patient profiles, and disease progression patterns. In this work, we propose a triclustering-based approach for three-way data classification, resulting from a combination of triclustering with random forests, and use it to predict the need for non-invasive ventilation in ALS patients. We analyse ALSFRS-R functional scores together with respiratory function tests collected from patient follow-up. The results are promising, enabling to understand the potential of triclustering and pinpointing improvements towards an effective triclustering-based classifier for clinical domains, taking advantage of the benefits of exploring disease progression patterns mined from three-way clinical data.

Keywords: Triclustering · Three-dimensional data · Three-way clinical data · Amyotrophic lateral sclerosis · Prognostic prediction

1 Introduction

Given a (real-valued, symbolic or heterogenous) three-dimensional dataset (three-way data), triclustering aims to discover subsets of observations, attributes, and contexts (triclusters) satisfying certain homogeneity and statistical significance

G. Panuccio et al. (Eds.): PACBB 2020, AISC 1240, pp. 112–122, 2021.
https://doi.org/10.1007/978-3-030-54568-0_12

criteria [10]. Promising triclustering applications in clinical domains are: multivariate physiological signal data analysis, where triclusters can capture coherent physiological responses for a group of individuals; neuroimaging data analysis, where triclusters can capture hemodynamic response functions and connectivity between brain regions; and clinical records analysis (patient-feature-time data), where triclusters correspond to groups of patients with correlated clinical features along time [10]. This work focuses on this latter class.

Amyotrophic Lateral Sclerosis (ALS) is a highly heterogeneous neurodegenerative disease characterized by a rapidly progressive muscular weakness. In general, patients with ALS generally die from respiratory failure within 3 to 5 years. However, some patients can live for less than one year, while others can live more than 10 years [9]. Worldwide, ALS affects between 5.9 and 39 people per 100.000 inhabitants [6]. In Portugal, 10 in 100.000 inhabitants suffer from this disease [7]. Most patients develop hypoventilation with hypoxemia and hypercapnia, requiring non-invasive ventilation (NIV) support [9]. In this context, foreseeing the beginning of hypoventilation is key to anticipate opportune interventions, such us the start of NIV. NIV was demonstrated to be effective in prolonging life and improving quality of life in ALS, in particular in patients without major bulbar muscles weakness [2,3]. In clinical practice, the Revised ALS Functional Rating Scale (ALSFRS-R) is broadly used to help clinicians disclose the state of disease progression [2]. In this scenario, Carreiro et al. [4] proposed the first prognostic models based on clinically defined time windows to predict the need for NIV in ALS. Following this work, Pires et al. [12] stratified patients according to their state of disease progression, and proposed specialized learning models based on three ALS progression groups (slow, normal and fast). Despite the promising results concerning using patient stratification for prognostic prediction, their prognostic models did not take into account the temporal dependence between the features. Matos [11] used biclustering-based classification. Biclustering was used to find groups of patients with coherent values in subsets of clinical features (biclusters), then used as features together with static demographic data. The results were interesting but no temporal data were used.

In this work, we propose to couple triclustering with Random Forests and train a triclustering-based classifier able to use disease progression patterns as features. The goal is to predict if a given patient will need NIV in the next 90 days using a classifier learned from temporal data from the follow-up of patients. We use the Lisbon ALS clinical dataset (version September 2019), developed at Hospital de Santa Maria (CHULN) in Lisbon since 1995, which was preprocessed and first used for prognostic prediction using time windows by Carreiro et al. [4].

In this case study, we use triclustering to find disease progression patterns in three-way clinical data, corresponding to groups of patients with coherent temporal evolution, which are then used for prognostic prediction. To this aim, we use triCluster [13], a pioneer and highly cited triclustering algorithm, proposed by Zhao and Zaki to mine patterns in three-way gene expression data. Despite not being proposed for clinical data, we believe it is a good starting point due to its algorithmic approach and type of patterns that it is able to find: quasi-exhaustive approach, mining arbitrarily positioned, potentially overlapping, scaling and shifting patterns.

In what follows we propose the triclustering-based classification approach for three-way clinical data, present results in ALS case study, and draw conclusions.

2 Methods

This section describes a new triclustering-based classification approach, where triclusters are discovered and then used as features in a Random Forest classifier. Figure 1 depicts the workflow. In what follows, we first cover triclustering clinical three-way data, briefly explaining triCluster, the triclustering algorithm used here to mine patterns in three-way data [13]. We then propose the triclustering-based Random Forests approach for three-way clinical data classification.

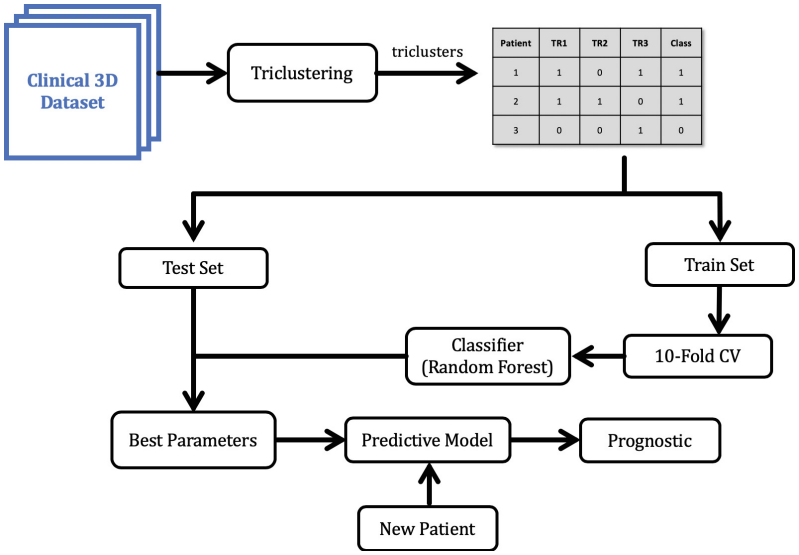
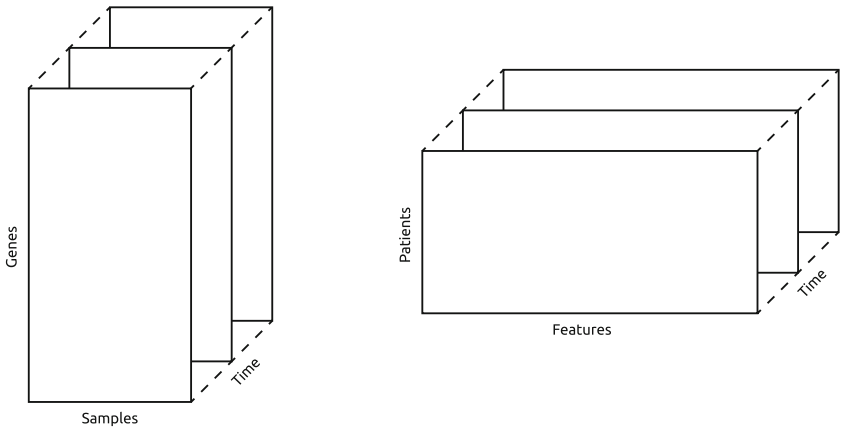


Fig. 1. Workflow of the proposed triclustering-based classifier.

2.1 Triclustering Three-Way Clinical Data

In this work, we use triCluster to identify triclusters on three-way clinical data from ALS patients. triCluster [13], proposed and implemented by Zhao and Zaki in 2005, is a pioneer and highly cited triclustering approach. It is a quasi-exhaustive approach, able to mine arbitrarily positioned and overlapping triclusters with constant, scaling, and shifting patterns from three-way data. Given triCluster was proposed to mine coherent triclusters in three-way gene expression data (gene-sample-time), at this point it is important to understand that clinical data can be preprocessed in order to have a similar structure, in which gene-sample-time data becomes patient-feature-time data. Figure 2 presents the analogy we made between these two different, but similar, three-way types of data, that enables triclustering three-way clinical data using triCluster.

triCluster has 3 main steps: 1) construct a multigraph with similar value ranges between all pairs of samples; 2) mine maximal biclusters from the multigraph formed for each time point (slices of the 3D dataset); and 3) extract triclusters by merging similar biclusters from different time-points. Optionally, it can delete or merge triclusters, according to the overlapping criteria used.



(a) Gene Expression Time-series. (b) Electronic Health Records.

Fig. 2. Gene expression and clinical three-way data representation.

2.2 Triclustering-Based Random Forest

After running triclustering the goal is to use triclusters for patient classification. To tackle this goal, we consider triclusters as features and construct a matrix of patients × triclusters to be used by the classifier. The approach followed here was to build a binary matrix, where the relation between a patient i and tricluster j is 1 if patient i is in tricluster j and 0 otherwise. After computing the class for each learning example a Random Forest is used to learn the predictive model.

In our ALS case study, three-way clinical data are composed by observations of patients at different appointments during follow-up (patient snapshots computed as in Carreiro et al. [4]. Since the goal is to predict the need for non-invasive ventilation within a given time window (90 days, corresponding to the next clinical appointment, in our case), the class used for each patient in the learning examples is binary and represents the patient evolution/non-evolution to a state where NIV is needed within 90 days (see [4] for details on computing patient snapshots and learning examples). The designed experimental pipeline mines triclusters from patients with at least two appointments and uses the binary matrix with labelled patients as input to a Random Forest classifier.

3 Results and Discussion

This section presents the results and discusses the challenges of learning a triclustering-based classifier, able to use disease progression patterns as features, from three-way clinical data. We used triCluster for triclustering and Random Forests for classification as described above. The goal is to predict if a given ALS patient will need NIV within 90 days given current and past clinical evaluations. We learn from the Lisbon ALS clinical dataset described below.

3.1 Data

We use the Lisbon ALS clinic dataset containing Electronic Health Records from ALS Patients regularly followed in our clinic, since 1995 and last updated in September 2019. Its current version (updated after the work of Pires et al. [12]) contains 1319 patients. Each patient has a set of static features (demographics, disease severity, co-morbidities, medication, genetic information, habits, trauma/surgery information and occupations) together with temporal features (collected repeatedly at follow-up), such as disease progression tests (ALSFRS-R scale, respiratory tests, etc.) and clinical laboratory investigations.

Since the focus of this work is three-way clinical data analysis, we focus on temporal data, discarding static data. We used 10 features per time point, the Functional Scores (ALSFRS-R), briefly described below, and respiratory tests: Forced Vital Capacity (FVC), Maximal Sniff Nasal Inspiratory Pressure (SNIP), Maximal Inspiratory Pressure (MIP) and Maximal Expiratory Pressure (MEP).

ALSFRS-R scores for disease progression rating are an aggregation of integers on a scale of 0 to 4 (where 0 is the worst and 4 is the best), providing different evaluations of the patient functional abilities at a given time point [8]. This functional evaluation is based on 13 questions, explained in Table 1. Different functional scores are then computed using subsets of scores, as shown in Table 2.

Table 1. ALSFRS-R Questions

Q1 - Speech
Q2 - Salivation
Q3 - Swallowing
Q4 - Handwriting
Q5 - Cutting food and handling utensils
Q6 - Dressing and hygiene
Q7 - Turning bed ans adjusting bed clothes
Q8 - Walking
Q9 - Climbing stairs
Q10 - Respiration
QR1 - Dyspnea
QR2 - Orthopnea
QR3 - Respiratory insufficiency

3.2 Data Preprocessing

The above ALS dataset with static and temporal features, was preprocessed as described by Carreiro et al. [4] and Pires et al. [12] to obtain patient snapshots and then compute the Evolution class for each snapshot using NIV administration date: a patient is labelled `'Y'`, if 90 days after the snapshot he/she was administrated NIV, and `'N'` otherwise. In this work, and in order to apply triCluster [13], we performed experiments using training examples computed as follows: 2, 3 and 4 consecutive snapshots for each patient (corresponding to clinical evaluations at 2, 3 and 4 consecutive appointments, respectively) were used as features, and the NIV evolution value of the last snapshot was used as class.

The first challenge was dealing with missing values, since triCluster does not support them. This is certainly a drawback, since missing values are common in clinical data, that should be taken into account in the triclustering step. In this work, and in order to test triCluster, we were thus forced to select only features with low levels of missing values for further analysis, leading to a subset of respiratory tests and the ALSFRS-R scores described above. We then removed all patients with missing values in more than 2 snapshots. For each remaining patient, we performed missing value imputation by using values in previous appointments to input latter missing values (Last Observation Carried Forward), when possible, and mean/mode of all patients values, otherwise.

After tackling missing values, we had to deal with class imbalance. In our case, and due to the time window of 90 days (next appointment) used as case study, the number of patients labeled as `'N'`, non-evolutions (2179, 1666 and 1283 examples, for 2, 3 and 4 snapshots, respectively), largely outnumbered those labelled as `'Y'` (326, 224 and 162 examples, for 2, 3 and 4 snapshots, respectively), identifying the patients requiring NIV within 90 days, key for the learning task. To deal with this issue we first used a Random Undersampler to reduce `'N'` examples until obtaining a class proportion of 2/3–1/3 (652, 448, 324 of `'N'` and 326, 224, 162 of `'Y'`, for 2, 3 and 4 snapshots, respectively), and then used SMOTE [5] to balance datasets to 50%/50% class proportion, leading to 1304, 896 and 648 learning examples, for 2, 3 and 4 snapshots, respectively.

Table 2. Functional scores and sub-scores according to ALSFRS-R.

Functional score	Description
ALSFRS	Sum of Q1 to Q10
ALSFRS-R	Sum of Q1 to Q9 + QR1 + QR2 + QR3
ALSFRSb	Q1 + Q2 + Q3
ALSFRSsUL	Q4 + Q5 + Q6
ALSFRSsLL	Q7 + Q8 + Q9
ALSFRSr	Q10
R	QR1 + QR2 + QR3

3.3 Model Evaluation

Since the main goal of this work is to evaluate the results of learning models for prognostic prediction in ALS patients using triclusters as features, we compare the performance of the proposed triclustering-based classification approach with a baseline obtained by training Random Forests with the original temporal features (used to compute the triclusters). To evaluate results, we use 5 × 10-fold Stratified Cross-Validation (CV) and compute Area Under the Curve (AUC), accuracy, and sensitivity and specificity, commonly used in clinical applications.

3.4 Baseline Results Using Random Forests and Original Features

Table 3 shows the baseline results using the original features: respiratory tests and ALSFRS-R scores for each appointment and treated as independent features (3 × 10 features). We can observe that baseline classifiers achieved classification accuracies around 0.78 in CV with low standard deviation. These are good results, considering those obtained by Pires et al. [12], using patient stratification and a large number of features. Sensitivity and specificity show approximately the same values, meaning all classifiers perform well when predicting both classes.

Table 3. Baseline results (Random Forest with original features).

	AUC	Accuracy	Sensitivity	Specificity
2TP	0.87 ± 0.0024	0.79 ± 0.0076	0.81 ± 0.0054	0.77 ± 0.0079
3TP	0.87 ± 0.0019	0.78 ± 0.0042	0.80 ± 0.0021	0.78 ± 0.0020
4TP	0.87 ± 0.0030	0.78 ± 0.0142	0.80 ± 0.0087	0.76 ± 0.0026

3.5 Results Using Random Forests and Triclusters as Features

Since triCluster allows different parameterizations, potentially discovering triclusters with different types of coherence, we run the algorithm using 3 different parameterizations: Case 1 - Unconstrained, to capture all coherent triclusters across the three dimensions (x-patient, y-feature and z-time); Case 2 - $\delta^x = \delta^y = \delta^z = 0$, to capture triclusters with constant values across the three dimensions; and Case 3 – $\delta^x = 0$, to force constant coherence on patient dimension while relaxing the others two. In all cases we set the minimum number of patients, features and time-points in each tricluster as 25, 2 and 3, respectively. triCluster discovered a total of 460 (121, 61 and 278), 179 (22, 32 and 125), 1250 triclusters (392, 459 and 399), for 2, 3 and 4 snapshots, respectively. In parentheses we show the triclusters found in Case 1, 2 and 3, respectively. We then used these triclusters as features for each case independently and for all cases altogether.

Can Triclusters Outperform Original Features? Table 4 shows the results obtained by the triclustering-based classifier using the triclusters obtained in each case above. It is possible to see that constant triclusters (Case 2) are the worst performing of the 3 cases. As expected, since the different cases capture different progression patterns, the performance improved when we trained the classifier using the triclusters obtained in the 3 cases. We tried to remove constant triclusters to evaluate if they are needless, but results were worse. Unfortunately, these results are not better than those obtained by baseline, meaning these triclusters cannot outperform the original features. Potential causes might rely on triCluster limitations: it was designed to analyse gene expression time series (real-valued) and not clinical data (mix of real-valued and categorical features); and its approach to deal with highly overlapping triclusters leads to the creation of redundant features, probably preventing other relevant triclusters to be discovered. Nevertheless, we are still interested to know if these triclusters (temporal features) can be used to improve baseline performance.

Table 4. Triclustering-based results using different parameterizations.

	AUC	**Accuracy**	**Sensitivity**	**Specificity**
		2TP		
Case 1	0.77 ± 0.0018	0.71 ± 0.0113	0.72 ± 0.0068	0.70 ± 0,0077
Case 2	0.73 ± 0.0016	0.68 ± 0.0028	0.75 ± 0.0042	0.64 ± 0.0026
Case 3	0.77 ± 0.0004	0.71 ± 0.0062	0.73 ± 0.0069	0.69 ± 0.0043
All	0.79 ± 0.0023	0.72 ± 0.0054	0.72 ± 0.0047	0.71 ± 0.0054
		3TP		
Case 1	0.71 ± 0.0013	0.68 ± 0.0016	0.68 ± 0.0013	0.66 ± 0.0014
Case 2	0.68 ± 0.0010	0.66 ± 0.0019	0.77 ± 0.0011	0.61 ± 0.0013
Case 3	0.74 ± 0.0011	0.69 ± 0.0037	0.76 ± 0.0023	0.66 ± 0.0028
All	0.77 ± 0.0008	0.71 ± 0.0043	0.74 ± 0.0031	0.69 ± 0.0021
		4TP		
Case 1	0.74 ± 0.0016	0.65 ± 0.0016	0.66 ± 0,0165	0.62 ± 0.0093
Case 2	0.72 ± 0.0014	0.66 ± 0.0073	0.71 ± 0.0031	0.63 ± 0.0020
Case 3	0.72 ± 0.0022	0.66 ± 0.0030	0.70 ± 0.0059	0.67 ± 0.0072
All	0.74 ± 0.0008	0.76 ± 0.0031	0.67 ± 0.0072	0.66 ± 0.0056

Can Triclusters Be Used to Improve Baseline Performance? In order to evaluate if triclusters can improve the results of baseline classifiers, we trained Random Forests using the triclusters together with the original features. As seen in Table 5, results without feature selection are approximately the same as those obtained as baseline. Furthermore, we also expected that when using temporal

features performance would improve. These results might be misleading, leading us to conclude triclusters are not important and original features are enough.

In this context, we decided to inspect feature importance at baseline (original features), and when triclusters are used together with original features. Figure 3 depicts the importance of used features for three snapshots and shows that, as we expected, some triclusters (being temporal features) arouse as more important than some original features. This lead us to believe that Random Forests are not dealing well with the 256 features. We thus decided to perform feature selection (FS), selecting the best 120 features (including triclusters) to be used by the classifier. The slight improvement in FS results in Table 5 confirms our intuition.

Table 5. Classification results with triclusters and original features.

	AUC	Accuracy	Sensitivity	Specificity
2TP				
Before FS	0.88 ± 0.0018	0.79 ± 0.0024	0.82 ± 0.0056	0.78 ± 0.0062
After FS	0.88 ± 0.0026	0.80 ± 0.0041	0.83 ± 0.0046	0.78 ± 0.0037
3TP				
Before FS	0.87 ± 0.0026	0.78 ± 0.0092	0.79 ± 0.0025	0.78 ± 0.0019
After FS	0.87 ± 0.0025	0.79 ± 0.0052	0.79 ± 0.0021	0.77 ± 0.0018
4TP				
Before FS	0.85 ± 0.0029	0.76 ± 0.0045	0.77 ± 0.0072	0.75 ± 0.0062
After FS	0.86 ± 0.0011	0.77 ± 0.0046	0.78 ± 0.0027	0.75 ± 0.0078

	#	Gain ratio	Gini
N 2ALS-FRSr		0.125	0.088
N 2R		0.073	0.068
N 2ALS-FRSb		0.066	0.083
N 2ALS-FRS-R		0.059	0.077
N 2ALS-FRS		0.057	0.075
N 2MEP		0.054	0.072
N 2FVC		0.052	0.069
N 1ALS-FRS-R		0.045	0.060
N 1ALS-FRS		0.044	0.058
N 1MEP		0.038	0.051
N 2ALS-FRSsUL		0.036	0.048
N 2MIP		0.033	0.044
N 0ALS-FRS		0.027	0.037
N 0MEP		0.026	0.036
N 1FVC		0.020	0.027

	#	Gain ratio	Gini
N 2ALS-FRSr		0.145	0.107
C 2TRI27	2	0.099	0.028
C 3TRI73	2	0.099	0.028
C 3TRI29	2	0.093	0.023
C 3TRI50	2	0.089	0.026
C 2TRI9	2	0.087	0.020
C 3TRI30	2	0.087	0.020
C 3TRI31	2	0.082	0.023
N 1ALS-FRSr		0.082	0.045
N 2R		0.081	0.077
C 3TRI97	2	0.080	0.021
N 1ALS-FRSb		0.079	0.084
C 3TRI54	2	0.075	0.023
C 3TRI26	2	0.073	0.016
C 3TRI28	2	0.072	0.017

Fig. 3. Feature ranking (top 15): feature importance for original features (left) and including triclusters (right). Images from Orange3 Data Mining Toolkit [1].

4 Conclusions

We proposed to couple triclustering with Random Forests and train a triclustering-based classifier able to use disease progression patterns as features. The goal was to predict if a given patient will need NIV in the next 90 days using temporal data from follow-up. In this case study, we used triclustering to find disease progression patterns in three-way clinical data, corresponding to groups of patients with coherent temporal evolution, then used for prognostic prediction.

The results are promising but pinpoint the limitations of the triclustering algorithm used when dealing with clinical data. In our opinion, a key advantage of a triclustering-based classification is the possibility to provide a better understanding of the results, promoting model interpretability (critical in clinical applications) together with potential improvements in classification (by incorporating temporal features). Since the triclusters identify subsets of patients with subsets of features showing coherent evolution patterns over contiguous time-points, we hypothesize they may uncover disease progression patterns, that might be key to boost classification results. We will thus work towards an effective triclustering-based classifier starting by improving triclustering results and in next make it able to yield improvements against state-of-the-art classifiers.

Acknowledgements. This work was partially supported by FCT funding to Neuroclinomics2 (PTDC/EEI-SII/1937/2014) and iCare4U (LISBOA-01-0145-FEDER-031474 + PTDC/EME-SIS/31474/2017) research projects, and LASIGE Research Unit (UIDB/00408/2020).

References

1. Orange3 Data Mining Toolkit. https://orange.biolab.si
2. Andersena, S.A., Borasioc, G.D., de Carvalho, M., Chioe, A., Van Dammef, P., Hardimang, O., Kolleweh, K., Morrisoni, K.E., et al.: Efns guidelines on the clinical management of amyotrophic lateral sclerosis (MALS)-revised report of an EFNS task force. Eur. J. Neurol. **19**, 360–375 (2011)
3. Bourke, S.C., Tomlinson, M., Williams, T.L., Bullock, R.E., Shaw, P.J., Gibson, G.J.: Effects of non-invasive ventilation on survival and quality of life in patients with amyotrophic lateral sclerosis: a randomised controlled trial. Lancet Neurol. **5**(2), 140–147 (2006)
4. Carreiro, A.V., Amaral, P.M., Pinto, S., Tomás, P., de Carvalho, M., Madeira, S.C.: Prognostic models based on patient snapshots and time windows: predicting disease progression to assisted ventilation in amyotrophic lateral sclerosis. J. Biomed. Inform. **58**, 133–144 (2015)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
6. Chiò, A., Logroscino, G., Traynor, B., Collins, J., Simeone, J., Goldstein, L., White, L.: Global epidemiology of amyotrophic lateral sclerosis: a systematic review of the published literature. Neuroepidemiology **41**(2), 118–130 (2013)

7. Conde, B., Winck, J.C., Azevedo, L.F.: Estimating amyotrophic lateral sclerosis and motor neuron disease prevalence in Portugal using a pharmaco-epidemiological approach and a bayesian multiparameter evidence synthesis model. Neuroepidemiology **53**(1–2), 73–83 (2019)
8. ENCALS: ALS functional rating scale revised (ALS-FRS-R). version (May 2015)
9. Heffernan, C., Jenkinson, C., Holmes, T., Macleod, H., Kinnear, W., Oliver, D., Leigh, N., Ampong, M.: Management of respiration in mnd/als patients: an evidence based review. Amyotroph. Lateral Scler. **7**(1), 5–15 (2006)
10. Henriques, R., Madeira, S.C.: Triclustering algorithms for three-dimensional data analysis: a comprehensive survey. ACM Comput. Surv. **51**(5), 95 (2019)
11. Matos, J.: Biclustering electronic health records to unravel disease presentation patterns. MSc Thesis (2019)
12. Pires, S., Gromicho, M., Pinto, S., Carvalho, M., Madeira, S.C.: Predicting non-invasive ventilation in ALS patients using stratified disease progression groups. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 748–757. IEEE (2018)
13. Zhao, L., Zaki, M.J.: Tricluster: an effective algorithm for mining coherent clusters in 3D microarray data, pp. 694–705 (2005)

Scientific Paper: Journal BMC MIDM

B

Submitted to Scientific Journal BMC Medical Informatics and Decision Making - starts on next page.

RESEARCH

Learning Explainable Predictive Models with a Triclustering-based Classifier: A Case Study on Aiding ALS Prognostics

Diogo F. Soares[1*], Rui Henriques[2], Marta Gromicho[3], Susana Pinto[3], Mamede de Carvalho[3] and Sara C. Madeira[1]

*Correspondence: dfsoares@ciencias.ulisboa.pt
[1]LASIGE, Faculdade de Ciências, Universiadade de Lisboa, Campo Grande, Lisbon, Portugal
Full list of author information is available at the end of the article
†Equal contributor

Abstract

Triclustering extends biclustering to three-dimensional data spaces, offering the possibility to retrieve interpretable temporal patterns from three-way time series data. Despite the highlighted relevance of triclustering for comprehensively mining discriminative patterns of disease progression, the role of these patterns in aiding clinical prognostics remains largely unexplored. This work proposes a new class of explainable predictive models from three-way data using discriminative triclusters. Triclustering searches are dynamically hyperparameterized to comprehensively find for informative triclusters (groups of individuals with a coherent pattern along a subset of variables and time points) and then use these temporal patterns as features within a state-of-the-art classifier with strict guarantees of interpretability. To this end, a temporally constrained triclustering algorithm, termed TCtriCluster algorithm, is devised to mine time-contiguous triclusters. The proposed methodology is used to predict the need for non-invasive ventilation (NIV) in Amyotrophic Lateral Sclerosis (ALS) patients. In this case study, we learnt a general prognostic model from all patients data and specialized models after patient stratification into Slow, Neutral and Fast progressors. The gathered results show that besides comparable and sometimes outperforming results against state-of-the-art alternatives, our triclustering-based classifier enhances prediction with the potentialities of model explainability by revealing the clinically relevant disease progression patterns underlying prognostics. Results further show that the temporal restriction is effective in improving the effectiveness of the predictive models. The proposed approach was validated in clinical practice, supporting healthcare professionals understanding the link between the highly heterogeneous patterns of ALS disease progression and NIV needs.

Keywords: Triclustering; Three-way Data; Predictive Models; Disease Progression Patterns; Amyotrophic Lateral Sclerosis

1 Introduction

Considering a (real-valued, symbolic or heterogenous) three-dimensional dataset (three-way data), triclustering aims to discover subsets of objects, features and contexts (triclusters), satisfying certain homogeneity and statistical significance criteria. Given the increasing prevalence of three-way data across biomedical and social domains, triclustering — the discovery of coherent subspaces within three-way data — became a key technique to enhance the understanding of complex biological, individual, and societal systems [1]. Clustering is limited in this context, since objects in

three-way data domains are typically only meaningfully correlated on subspaces of the overall space, and although biclustering is more powerfull by enabling subspaces of both objects and features, it disregards context [2].

Promising triclustering applications in clinical domains are multivariate physiological signal data analysis, where triclusters can capture coherent physiological responses for a group of individuals; neuroimaging data analysis, where triclusters can capture hemodynamic response functions and connectivity between brain regions; and clinical records analysis, where triclusters identify groups of patients with correlated clinical features along time [1,3,4].

In this context, we propose a triclustering-based classifier to learn predictive models from three-way data, taking advantage of the temporal dependence between the features, and enhancing model interpretability by learning from local temporal patterns. In the proposed methodology, we designed and use TCTRICLUSTER, a temporally constrained triclustering algorithm able to mine time-contiguous triclusters. TCTRICLUSTER is an extension of TRICLUSTER [5], a pioneer and highly cited triclustering algorithm, proposed by Zhao and Zaki to mine patterns in three-way gene expression data, extended to cope with three-way heterogeneous data and incorporate a temporal contiguity constraint.

As case study, we use the analysis of three-way data from clinical records (patient-feature-time data). We target prognostic prediction in Amyotrophic Lateral Sclerosis (ALS) using a large cohort of Portuguese patients, where the triclusters learnt from patients' follow-up data can be interpreted as disease progression patterns. ALS is a highly heterogeneous neurodegenerative disease characterized by a rapidly progressive muscular weakness. In general, patients with ALS generally die from respiratory failure within 3 to 5 years. However, some patients can live for less than one year, while others can live more than 10 years [6]. Worldwide, ALS affects between 5.9 and 39 people per 100.000 inhabitants [7]. In Portugal, 10 in 100.000 inhabitants suffer from this disease [8]. Most patients develop hypoventilation with hypoxemia and hypercapnia, requiring non-invasive ventilation (NIV) support [6]. In this context, foreseeing the beginning of hypoventilation is key to anticipate opportune interventions, such us the start of NIV. NIV was demonstrated to be effective in prolonging life and improving quality of life in ALS, in particular in patients without major bulbar muscles weakness [9,10]. In clinical practice, the Revised ALS Functional Rating Scale (ALSFRS-R) is broadly used by clinicians to evaluate disease progression [9]. However, the highly heterogeneity of this disease turns prognosis into a challenge, pinpointing the need for advanced interpretable machine learning models, to help unravelling the complexity of the disease by identifying disease progression patterns that can be effectively used by clinicians to increase survival and promote patient care.

In this scenario, Carreiro et al. [11] proposed the first prognostic models based on clinically defined time windows to predict the need for NIV in ALS. Following this work, Pires et al. [12] stratified patients according to their state of disease progression, and proposed specialized learning models based on three ALS progression groups (slow, neutral and fast). They further used patient and clinical profiles with promising results [13]. Nevertheless, neither the approach using patient stratification or using clinical and patient profiles for prognostic prediction, took into account

images/Metodologias-IV.pdf

Figure 1 Proposed Workflow to Learn a Triclustering-based Classifier

the temporal dependence between the features. Matos et al. [14] used biclustering-based classification. Biclustering was used to find groups of patients with coherent values in subsets of clinical features (biclusters), then used as features together with static demographic data. The results were interesting but no temporal data were used.

In this work, we use triclustering to find disease progression patterns in three-way clinical data, corresponding to groups of patients with coherent temporal evolution, which are then used for prognostic prediction. To this aim, we used the proposed methodology to learn a prognostic model using a triclustering-based classifier to predict whether a patient will need NIV in the next 90 days. The proposed TCTRICLUSTER algorithm is used to mine the triclusters, ALS progression patterns, then used in the predictive models. We learnt a general prognostic model, from all patients data, and specialized models after patient stratification into Slow, Neutral and Fast progressors, as performed by Pires et al. [12]. The results are promising, highlighting the potential of the proposed methodology, not only concerning predictability, but also interpretability.

Furthermore, the proposed triclustering-based classifier can be used straightforwardly to learn other prognostic models from follow-up data in other diseases, or other predictive models from 3-way data, from other domains. The TCTRICLUSTER algorithm can be further used as standalone tool to mine arbitrarily positioned, overlapping, and temporally constrained triclusters, with constant, scaling, and shifting patterns from three-way data.

The paper is organized as follows: Section II presents the proposed methodology to learn and use a triclustering-based classifier, together with the TCTRICLUSTER algorithm; Section III presents and discusses the results obtained in the ALS case study; Section IV shows the performed model interpretability analysis, magnifying the patterns discovered and used by the predictive models; finally Section IV presents the conclusions and future work.

2 Methods

This section describes the proposed methodology to learn a triclustering-based classifier from three-way data, from preprocessing (including creating learning examples) to classifier performance evaluation, exploring several approaches in each step. It further describes TCTRICLUSTER, the proposed triclustering algorithm to mine temporally constrained triclusters. Figure 1 depicts the overall workflow.

In what follows, consider that a three-way dataset, D, is defined by n objects $X = \{x_1, ..., x_n\}$, m features $Y = \{y_1, ..., y_m\}$, and p contexts $Z = \{z_1, ..., z_p\}$, where the elements d_{ijk} relate object x_i, feature y_j, and context z_k. Consider also that, a bicluster $B = (I, J)$ is a subspace given by a subset of objects, $I \subseteq X$, and a subset of features, $J \subseteq Y$ [2]. Similarly, a tricluster $\mathcal{T} = (I, J, Z)$, contains $I \subseteq X$ objects, $J \subseteq Y$ features and $K \subseteq Z$ contexts, and t_{ijk} denote the elements of $\mathcal{T}$,

[width=.485]images/Metodologias-I.pdf

Figure 2 Learning Triclustering Best Parameters: Workflow

where $1 \leq i \leq I$, $1 \leq j \leq J$ and $1 \leq k \leq K$ [1]. In this context, each tricluster $\mathcal{T}$ can be represented as a set of K biclusters $\mathcal{T} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_K\}$:

$$\mathcal{B}_1 = \begin{bmatrix} t_{111} & t_{121} & \cdots & t_{1J1} \\ t_{211} & t_{221} & \cdots & t_{2J1} \\ \vdots & \vdots & \ddots & \vdots \\ t_{I11} & t_{I21} & \cdots & t_{IJ1} \end{bmatrix}$$

$$\mathcal{B}_2 = \begin{bmatrix} t_{112} & t_{122} & \cdots & t_{1J2} \\ t_{212} & t_{222} & \cdots & t_{2J2} \\ \vdots & \vdots & \ddots & \vdots \\ t_{I12} & t_{I22} & \cdots & t_{IJ2} \end{bmatrix}$$

$$\vdots$$

$$\mathcal{B}_K = \begin{bmatrix} t_{11K} & t_{12K} & \cdots & t_{1JK} \\ t_{21K} & t_{12K} & \cdots & t_{1JK} \\ \vdots & \vdots & \ddots & \vdots \\ t_{I1K} & t_{I2K} & \cdots & t_{IJK} \end{bmatrix}$$

2.1 Preprocessing Data

The three-way dataset, composed by several heterogeneous features measured over a number of time-points, is first preprocessed to obtain learning examples. Depending on the dataset dealing with missing values and class imbalance, might also be needed.

2.2 Learning Triclustering Best Parameters

In this step, the goal is to compute the best parameters to be used as input by the triclustering algorithm (described later) in order to obtain the best classification performance. The workflow, depicted in Figure 2, starts by performing triclustering on the preprocessed data to obtain triclusters. Next, the virtual pattern 3D is computed for each tricluster. The proposed virtual pattern 3D, extended from the 2D version defined in [15], is computed as follows, using the virtual pattern of each bicluster composing the tricluster.

Definition 1 (Virtual Pattern 3D). Given a tricluster $\mathcal{T}$, its virtual pattern $\mathcal{P}$ is defined as a set of elements $\mathcal{P} = \{\rho_1, \rho_2, ..., \rho_I\}$, where $\rho_i, 1 \leq i \leq I$ is defined as the mean (or the mode, in case of categorical features) of values in the i^{th} row for each context:

$$\rho_i = \frac{1}{J \cdot Z} \sum_{z=1}^{Z} \sum_{j=1}^{J} b_{ijz} \tag{1}$$

We use the computed virtual pattern 3D, $\mathcal{P}$, to assess how well a specific object (patient), p_i, follows the general tendency of a given tricluster $\mathcal{T}$. We propose two approaches: (1) compute the Euclidean distance; or (2) compute Pearson correlation between the 3D virtual pattern $\mathcal{P}$ and the equivalent pattern (same features and contexts) of p_i.

We denote these assessments as Virtual Distance 3D and Virtual Correlation 3D, and define them as follows:

Definition 2 (Virtual Distance 3D). The virtual distance between an observation p_i and a tricluster $\mathcal{T}$ is defined as

$$\texttt{VD}_{3D}(p_i, \mathcal{T}) = E(p_i, \rho) = \sqrt{\sum_{e=1}^{I}(p_{i_e} - \rho_e)^2} \tag{2}$$

Definition 3 (Virtual Correlation 3D). The virtual correlation between an object p_i and a tricluster $\mathcal{T}$ is defined as

$$\texttt{VC}_{3D}(p_i, \mathcal{T}) = r(p_i, \rho) = \frac{\sum_{e=1}^{I}(p_{i_e} - \bar{p_i})(\rho_e - \bar{\rho})}{\sqrt{\sum_{e=1}^{I}(p_{i_e} - \bar{p_i})^2 \sum_{e=1}^{I}(\rho_e - \bar{\rho})^2}} \tag{3}$$

Considering as an example a Tricluster $T(I, J, K)$, mined from three-way data $S(X, Y, Z)$, composed by 3 objects, 3 features and 3 contexts, such that $I = \{X_1, X_3, X_7\}$, $J = \{Y_1, Y_3, Y_7\}$, $K = \{Z_2, Z_3, Z_4\}$. Y_1 and Y_3 contains only categorical values. For simplicity, consider $T = \{B_2, B_3, B_4\}$:

$$B_2 = \begin{bmatrix} 1 & 3.1 & 5 \\ 1 & 2.8 & 3 \\ 3 & 2.1 & 10 \end{bmatrix}; B_3 = \begin{bmatrix} 2 & 3.0 & 3 \\ 3 & 2.8 & 3 \\ 3 & 2.9 & 9 \end{bmatrix}; B_4 = \begin{bmatrix} 3 & 2.9 & 3 \\ 2 & 2.9 & 3 \\ 3 & 2.4 & 8 \end{bmatrix}$$

and an object (patient) $P(X_p, I, K)$ defined as $P = \{C_2, C_3, C_4\} : C_2 = \begin{bmatrix} 1 & 2.22 & 5 \end{bmatrix}$; $C_3 = \begin{bmatrix} 1 & 2.26 & 7 \end{bmatrix}$; $C_4 = \begin{bmatrix} 2 & 2.35 & 8 \end{bmatrix}$

The Virtual Patterns are: $\rho(B_2) = \begin{bmatrix} 1 & 2.6667 & 5 \end{bmatrix}$; $\rho(B_3) = \begin{bmatrix} 3 & 2.9 & 3 \end{bmatrix}$; $\rho(B_4) = \begin{bmatrix} 3 & 2.7333 & 3 \end{bmatrix}$; and $\rho(T) = \begin{bmatrix} 3 & 2.7667 & 3 \end{bmatrix}$.

2.2.1 Triclustering Algorithm

For these methodology, we considered TriCluster [5], the pioneer and highly cited triclustering approach, proposed and implemented by Zhao and Zaki in 2005. It is a quasi-exhaustive approach, able to mine arbitrarily positioned and overlapping triclusters with constant, scaling, and shifting patterns from three-way data. Given that TriCluster was proposed to mine coherent triclusters in three-way gene expression data (gene-sample-time), at this point it is important to understand that clinical data can be preprocessed in order to have a similar structure, in which gene-sample-time data becomes patient-feature-time data, for instance. TriCluster has

3 main steps: 1) construct a multigraph with similar value ranges between all pairs of samples; 2) mine maximal biclusters from the multigraph formed for each time point (slices of the 3D dataset); and 3) extract triclusters by merging similar biclusters from different time-points. Optionally, it can delete or merge triclusters, according to the overlapping criteria used.

However, since our goal is to mine temporal three-way data, meaning the Z dimension (contexts) is time, we borrowed the idea in CCC-Biclustering [16], a state of the art and highly efficient temporal biclustering algorithm, and introduced a temporal constraint in triclustering. The goal thus became to mine Time-Contiguous Triclusters (TCTriclusters), triclusters with consecutive time-points. In this context, we re-implemented TriCluster in Python and extended it to cope with a time constrain. The new TCTriCluster algorithm implements this time constrain on its 3rd phase, as shown in Algorithm 1 (line 9).

TCTriCluster allows different combinations of input parameters, that should be explored in order to discover the best parameters, with which the final classifier should be learnt. The input parameters are: $\varepsilon, mx, my, mz, \delta^x, \delta^y, \delta^z, \eta$ and γ, corresponding to maximum ratio value, minimum size of tricluster dimensions x, y and z, maximum range threshold along dimensions x, y and z, overlapping and merging threshold, respectively.

Algorithm 1: TCTriCluster: Extension of triCluster able to mine TC-Triclusters

Input: $\varepsilon, mx, my, mz, \delta^x, \delta^y, \delta^z$, bicluster sets $\{\mathcal{C}^t\}$ of all contexts (time-points), set of objects X, features Y and contexts (time-points) Z
Output: cluster set $\mathcal{C}$

1 **Initialisation**: $\mathcal{C} = \varnothing$, call TCTriCluster($\mathcal{T} = X \times Y \times \varnothing$, Z)
2 TCTriCluster ($\mathcal{T} = I \times J \times K$, U)
3 **if** $\mathcal{T}$ *satisfies* $\delta^x, \delta^y, \delta^z$ **then**
4 **if** $|\mathcal{T}.K| \geq mz$ **then**
5 **if** $\mathcal{T} \not\subset \mathcal{T}' \in \mathcal{C}$ **then**
6 Delete any $\mathcal{T}'' \in \mathcal{C}$, if $\mathcal{T}'' \subset \mathcal{T}$
7 Add $\mathcal{T}$ to $\mathcal{C}$
8 **foreach** $t_i \in U$ **do**
9 **if** $(\mathcal{T}^{new} = \varnothing) \vee (\mathcal{T}.K_{-1} + 1 = t_i)$ **then**
10 $\mathcal{T}^{new}.K \longleftarrow \mathcal{T}.K + t_i$
11 Remove t_i from U
12 **forall** $t_k \in \mathcal{T}.K$ *and each bicluster* $b_i^{t_k} \in \mathcal{C}^{t_k}$, *such that* $|b_i^{t_k}.I \cap \mathcal{T}.I| \geq mx$ *and* $|b_i^{t_k}.J \cap \mathcal{T}.J| \geq my$ **do**
13 $\mathcal{T}^{new}.I \longleftarrow b_i^{t_k}.I \cap \mathcal{T}.I$
14 $\mathcal{T}^{new}.J \longleftarrow b_i^{t_k}.J \cap \mathcal{T}.J$
15 **if** $|\mathcal{T}^{new}.I| \geq mx$ *and* $|\mathcal{T}^{new}.J| \geq my$ *and the ratios at time* t_i, t_k *are coherent* **then**
16 TCTriCluster ($\mathcal{T}^{new}, P$)

After computing similarities matrices based on the virtual patterns (based on distance or correlation) a 5×10-fold Stratified Cross-Validation is performed using the Triclustering-based Classifier in order to find the best triclustering parameters, using classification performance as metric. The best parameters and then fed to the next step.

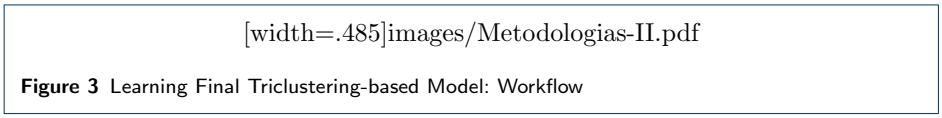


Figure 3 Learning Final Triclustering-based Model: Workflow

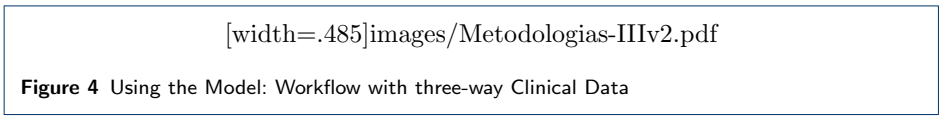


Figure 4 Using the Model: Workflow with three-way Clinical Data

2.3 Learning Final Classifier

Figure 4 depicts the steps involved in learning the final model. With the best parameters output in the previous step, a final iteration is perform in order to obtain the final Triclustering-based predictive model and the final triclusters, then used to make predictions in the next step.

2.4 Using Model

After learning the final triclustering-based predictive model, it can be used to classify new three-way objects. To do this, it is necessary to first calculate the array of similarities between the new object and the triclusters obtained in the previous steps. This array will be fed to the classifier that will in turn return the classification for the new object with a percentage of accuracy. Figure 4 depicts an example using clinical three-way data (case study described in the next section).

3 Results and Evaluation

3.1 Data

We use the Lisbon ALS clinic dataset containing Electronic Health Records from ALS Patients regularly followed at the local ALS clinic, since 1995 and last updated in March 2020. Its current version contains 1374 patients. Each patient has a set of static features (demographics, disease severity, co-morbidities, medication, genetic information, habits, trauma/surgery information and occupations) together with temporal features (collected repeatedly at follow-up), such as disease progression tests (ALSFRS-R scale, respiratory tests, etc) and clinical laboratory investigations.

Since the focus of the proposed methodology in three-way clinical data analysis, we focus on temporal data, discarding static data. We used 10 features per time point, the Functional Scores (ALSFRS-R), briefly described below, and respiratory tests: Forced Vital Capacity (FVC), Maximal Inspiratory Pressure (MIP) and Maximal Expiratory Pressure (MEP).

ALSFRS-R scores for disease progression rating are an aggregation of integers on a scale of 0 to 4 (where 0 is the worst and 4 is the best), providing different evaluations of the patient functional abilities at a given time point [17]. This functional evaluation is based on 13 questions, explained in Table 1. Different functional scores are then computed using subsets of scores, as shown in Table 2.

3.2 Preprocessing

The Lisbon ALS clinic dataset with several different features, was preprocessed as described by Carreiro et al. [11] and Pires et al. [12] to obtain patient snapshots and then compute the Evolution class for each snapshot using NIV administration

Table 1 ALSFRS-R Questions

Q1 - Speech
Q2 - Salivation
Q3 - Swallowing
Q4 - Handwriting
Q5 - Cutting food and Handling Utensils
Q6 - Dressing and Hygiene
Q7 - Turning bed ans adjusting bed clothes
Q8 - Walking
Q9 - Climbing Stairs
Q10 - Respiration
QR1 - Dyspnea
QR2 - Orthopnea
QR3 - Respiratory Insufficiency

Table 2 Functional Scores and Sub-scores according to ALSFRS-R.

Functional Score	Description
ALSFRS	sum of Q1 to Q10
ALSFRS-R	sum of Q1 to Q9 + QR1 + QR2 + QR3
ALSFRSb	Q1 + Q2 + Q3
ALSFRSsUL	Q4 + Q5 + Q6
ALSFRSsLL	Q7 + Q8 + Q9
ALSFRSr	Q10
R	QR1 + QR2 + QR3

date: a patient is labelled `Y`, if 90 days after the snapshot he/she was administrated NIV, and `N` otherwise.

We performed experiments using training examples composed by: 3, 4 and 5 consecutive snapshots (CS) for each patient (corresponding to clinical evaluations at 3, 4 and 5 consecutive appointments, respectively) were used as features, and the NIV evolution value of the last snapshot was used as class. For each remaining patient, we performed missing value imputation by using values in previous appointments to input latter missing values (Last Observation Carried Forward), when possible, and mean/mode of all patients values, otherwise.

After tackling missing values, we had to deal with class imbalance. In our case, and due to the time window of 90 days (next appointment) used as case study, the number of patients labeled as `N`, non-evolutions, largely outnumbered those labelled as `Y`, identifying the patients requiring NIV within 90 days, key for the learning task. To deal with this issue, we first used a Random Undersampler (RU) to reduce `N` examples until obtaining a class proportion of 2/3 - 1/3 and then used SMOTE [18] to balance datasets to 50%/50% class proportion. Values for the class proportions obtained after RU and SMOTE are depicted in Table 3.

Table 3 Number and Class Distribution of Learning Examples.

	Total		RU		SMOTE	
	N	Y	N	Y	N	Y
3 CS	1721	227	457	227	454	454
4 CS	1335	166	332	166	332	332
5 CS	1038	121	242	121	242	242

3.3 Baseline Results: Random Forests with Original Features

To obtain baseline results, to whom compare the predictive models learned with triclustering-based classifiers, we trained a Random Forest using the original features (respiratory tests and ALSFRS-R scores) treated as independent features (number of CS $\times$ 10 features). Table 4 shows the baseline results. We can observe that baseline classifier achieved classification accuracies around 0.80 in CV with low standard deviation. Sensitivity and specificity show approximately the same values, meaning all classifiers perform well when predicting both classes.

Table 4 Baseline Results using Random Forests and Original Features

CS	AUC	Accuracy	Sensitivity	Specificity
3	0.87 $\pm$ 0.0326	0.79 $\pm$ 0.0371	0.83 $\pm$ 0.0437	0.74 $\pm$ 0.0702
4	0.89 $\pm$ 0.0370	0.80 $\pm$ 0.0509	0.84 $\pm$ 0.0659	0.75 $\pm$ 0.0899
5	0.89 $\pm$ 0.0462	0.80 $\pm$ 0.0577	0.78 $\pm$ 0.0818	0.75 $\pm$ 0.0879

3.4 Triclustering-based Classification Results

To test the proposed triclustering-based classifier on the ALS case study, allowing fair comparisons with the baseline, we also use Random Forests as classifier.

Since TriCluster allows different parameterizations, potentially discovering triclusters with different types of coherence, we run the algorithm using three different settings: Unconstrained, to capture all coherent triclusters across the three dimensions (x-patient, y-feature and z-time); $\delta^x = \delta^y = \delta^z = 0$, to capture triclusters with constant values across the three dimensions; and $\delta^x = 0$, to force constant coherence on patient dimension while relaxing the others two. Together with these parameterizations, we varied deletion (η) and merging (γ) thresholds from 0.45 to 0.95 (and none). We set minimum number of features and time-points in each tricluster as 1 and 2 respectively, and minimum number of patients as 25, 20 and 15 for 3, 4 and 5 consecutive snapshots (CS).

To compute the similarities matrix, to be used by each triclustering-based classifier, we tried the two proposed approaches: distance (D) and correlation (C). We decided to compute the similarities between the patients and the different biclusters that compose the triclusters, since according to the case study and for interpretability concerns this will make the features more informative, than when considering similarities with the general trend of the complete tricluster.

Table 5 shows the best triclustering parameters obtained with the different CS experiments. As we can see, Unconstrained was chosen as best parameter with all the experiments.

Table 5 Learned Triclustering Best Parameters

A	CS	Best Parameters
D	3	Unconstrained; $\eta = 0.95$
	4	Unconstrained
	5	Unconstrained
C	3	Unconstrained; $\gamma = 0.50$
	4	Unconstrained
	5	Unconstrained; $\gamma = 0.95$

Table 6 shows the results obtained by Triclustering-based classifier using the original version of triCluster [5] and Table 7 depicts the results obtained with the

Table 6 Performance Evaluation Results of Triclustering-based Classifier using original TriCluster Algorithm

AP	CS	AUC	Accuracy	Sensitivity	Specificity
D	3	0.79 ± 0.0384	0.72 ± 0.0364	0.72 ± 0.0561	0.72 ± 0.0631
	4	0.79 ± 0.0021	0.70 ± 0.0071	0.71 ± 0.0096	0.72 ± 0.0045
	5	0.82 ± 0.0016	0.76 ± 0.0048	0.76 ± 0.0088	0.75 ± 0.0044
C	3	0.79 ± 0.0010	0.71 ± 0.0064	0.71 ± 0.0089	0.71 ± 0.0050
	4	0.78 ± 0.0023	0.70 ± 0.0066	0.69 ± 0.0045	0.71 ± 0.0096
	5	0.85 ± 0.0019	0.75 ± 0.0086	0.74 ± 0.0146	0.75 ± 0.0049

extended version TCTRICLUSTER (see Algorithm 1). As we can see although these results still do not exceed the same results obtained by the baseline, they generally exceed the results of Table 6, which proves the effectiveness of the temporal constraint, allowing to improve group coherence and consequently classifier performance.

Table 7 Performance Evaluation Results of Triclustering-based Classifier learned from ALS Lisbon Clinic Data

AP	CS	AUC	Accuracy	Sensitivity	Specificity
D	3	0.84 ± 0.0384	0.76 ± 0.0364	0.78 ± 0.0561	**0.74 ± 0.0631**
	4	0.86 ± 0.0389	0.78 ± 0.0432	0.80 ± 0.0706	0.76 ± 0.0830
	5	0.85 ± 0.0615	0.76 ± 0.0669	0.78 ± 0.0967	0.75 ± 0.0979
C	3	0.84 ± 0.0380	0.75 ± 0.0379	0.78 ± 0.0560	0.72 ± 0.0649
	4	0.85 ± 0.0428	0.75 ± 0.0456	0.76 ± 0.0780	0.74 ± 0.0767
	5	0.85 ± 0.0546	0.75 ± 0.0598	**0.80 ± 0.0856**	0.71 ± 0.0967

Analyzing the results in Table 7, we can verify that the highest precision values were obtained by the classifiers using the distance approach to compute the similarity matrix, and that these are comparable to those obtained by the baseline. We note that despite not outperforming the baseline, these results improve the interpretability of the model, since the different temporal patterns used can uncover disease progression patterns and help clinicians in their predictive task.

3.5 Results using Patient Stratification

Using the same data, we stratified the patients in three groups according with their disease progression and following the approach used by Pires et al. [12]. The patients thus were stratified in Slow, Neutral and Fast progressors according to a Progression Rate (PR) value, computed by:

$$\mathrm{PR} = \frac{48 - \text{ALSFRS-R}_{\text{1st Visit}}}{\Delta t_{\text{1st Symptoms; 1st Visit}}}, \tag{4}$$

where 48 is the maximum score for ALSFRS-R feature, $\text{ALSFRS-R}_{\text{1st Visit}}$ is the ALSFRS-R score in the first appointment (diagnosis) and $\Delta t_{\text{1st Symptoms; 1st Visit}}$ is the time in months between the dates of first symptoms and the first appointment [12]. Progression rates are computed for each patient, and then based on these values they are divided into three groups based the distribution of PR values, and as suggested by the clinicians: 25% of patients with lower and higher values are stratified as Slow and Fast progressors, respectively. The remaining 50% are grouped and considered Neutral progressors.

After stratifying the patients in the three groups of patients, we apply our triclustering-based methodology to learn a specialized predictive model for each group of patients. Table 8 shows the results, obtained in the same way as those for the model learned with all data, as baseline. As expected, the results after patient stratification outperform the results obtained when learning from all patients, since patients are now more homogeneous. The small number patients with 4 and 5 CS in the Fast group prevented us from obtaining reliable results in these cases.

Table 8 Baseline Results: Random Forests with Original Features per Disease Progression Group

CS	Group	AUC	Accuracy	Sensitivity	Specificity
3	Slow	0.89 ± 0.0713	0.80 ± 0.0841	0.84 ± 0.1083	0.76 ± 0.1201
	Neutral	0.87 ± 0.0407	0.79 ± 0.0474	0.80 ± 0.0677	0.78 ± 0.0787
	Fast	0.81 ± 0.1668	0.74 ± 0.1454	0.74 ± 0.2627	0.74 ± 0.1998
4	Slow	0.93 ± 0.0986	0.86 ± 0.0895	0.87 ± 0.0996	0.85 ± 0.1236
	Neutral	0.85 ± 0.0683	0.771 ± 0.0645	0.81 ± 0.0797	0.73 ± 0.1112
	Fast				
5	Slow	0.92 ± 0.0562	0.839 ± 0.0730	0.84 ± 0.1166	0.84 ± 0.1178
	Neutral	0.87 ± 0.0729	0.79 ± 0.0788	0.79 ± 0.1021	0.79 ± 0.1235
	Fast				

Table 9 shows the results obtained by the specialized models for each disease progression group, according to the number of consecutive snapshots considered. We can see that the results outperform the baseline for 3 and 4 CS in the Neutral group, using the distance approach for similarities, confirming the evidence that this approach is the most appropriate to this type of data. Comparing these results with those obtained by the general classifier we can see that there is a noticeable increase in classification performance, resulting from being able to discover and use group specific disease progression patterns. Since the general model learns from heterogeneous patients, with a wide spectrum of disease progression patterns, where the Neutral group is dominant, it potentially misses significant patterns from patients who not follow a common disease progression trend and whose specific disease progression patterns are not discovered. Despite the good results in the Neutral group, the results concerning the Slow and Fast groups slightly decreased, since the class imbalance in these groups is even more accentuated than in complete dataset or in the neutral group. This imbalance has hampered the computation of some metrics given the bias.

In order to assess whether the general model would also be better for Slow and Faster progressors in the same way that the specialized model would be better for the Neutral group, we perform experiments with the general model to evaluate its performance with patients from the different three groups. Table 10 shows the results obtained, and we can see that even for Slow and Fast progressors the general model performs well, but specialized models achieved more balanced results in the prediction of both classes (values of sensitivity and specificity are closer). As already mentioned the general model by the specialized model in the Neutral group.

We note however, that these results could be improved with a larger collection of patients data from the Slow and Fast groups, as shown by the slightly higher values of standard deviation in these cases.

Table 9 Performance Evaluation Results obtained with Triclustering-based Classifier (Specialized for each Disease Progression Group)

AP	CS	Group	AUC	Accuracy	Sensitivity	Specificity
D	3	Slow	0.84 ± 0.0933	0.77 ± 0.0924	0.81 ± 0.1386	0.73 ± 0.1377
		Neutral	0.86 ± 0.0415	**0.794 ± 0.0393**	0.79 ± 0.0583	0.80 ± 0.0694
		Fast	0.70 ± 0.1798	0.66 ± 0.1582	0.81 ± 0.2211	0.51 ± 0.2397
	4	Slow	0.87 ± 0.0797	0.81 ± 0.0835	0.85 ± 0.1063	0.77 ± 0.1221
		Neutral	0.84 ± 0.0635	**0.774 ± 0.0731**	0.77 ± 0.0826	0.78 ± 0.1059
		Fast	0.70 ± 0.2666	0.74 ± 0.2094	0.80 ± 0.2442	0.68 ± 0.3177
	5	Slow	0.89 ± 0.0739	**0.841 ± 0.0908**	0.88 ± 0.1221	0.81 ± 0.1178
		Neutral	0.81 ± 0.0808	0.72 ± 0.0839	0.71 ± 0.1118	0.74 ± 0.1204
		Fast				
C	3	Slow	0.85 ± 0.0827	0.75 ± 0.0874	0.76 ± 0.1394	0.75 ± 0.1470
		Neutral	0.86 ± 0.0404	0.78 ± 0.0471	0.76 ± 0.0769	0.79 ± 0.0746
		Fast	0.66 ± 0.1768	0.66 ± 0.1586	0.80 ± 0.2201	0.51 ± 0.2447
	4	Slow	0.85 ± 0.0762	0.79 ± 0.0933	0.81 ± 0.1237	0.78 ± 0.1436
		Neutral	0.83 ± 0.0655	0.77 ± 0.0660	0.78 ± 0.0819	0.76 ± 0.0968
		Fast	0.63 ± 0.2414	0.70 ± 0.1854	0.77 ± 0.2424	0.62 ± 0.2978
	5	Slow	0.88 ± 0.0822	0.83 ± 0.0858	0.88 ± 0.1021	0.78 ± 0.1343
		Neutral	0.81 ± 0.0780	0.73 ± 0.0800	0.71 ± 0.1159	0.75 ± 0.1194
		Fast				

Table 10 Performance Evaluation Results obtained with Triclustering-based Classifier (General) per Disease Progression Group

CS	Group	AUC	Accuracy	Sensitivity	Specificity
3	Slow	0.85 ± 0.1094	0.81 ± 0.0754	0.67 ± 0.3053	0.84 ± 0.0765
	Neutral	0.82 ± 0.0536	0.74 ± 0.0924	0.78 ± 0.0745	0.69 ± 0.0879
	Fast	0.78 ± 0.0965	0.70 ± 0.0956	0.76 ± 0.1082	0.59 ± 0.1629
4	Slow	0.89 ± 0.0986	0.82 ± 0.0895	0.70 ± 0.2680	0.86 ± 0.0923
	Neutral	0.84 ± 0.0683	0.77 ± 0.0776	0.80 ± 0.0897	0.74 ± 0.1420
	Fast	0.81 ± 0.1100	0.75 ± 0.0937	0.85 ± 0.1086	0.57 ± 0.2116
5	Slow	0.86 ± 0.0474	0.80 ± 0.0844	0.65 ± 0.2256	0.85 ± 0.1192
	Neutral	0.82 ± 0.0760	0.72 ± 0.0508	0.78 ± 0.0784	0.63 ± 0.1192
	Fast	0.75 ± 0.1198	0.74 ± 0.0890	0.85 ± 0.0794	0.52 ± 0.1532

4 Model Interpretability

The relevance of a triclustering-based classifier methodology should be evaluated not only by analysing its performance regarding classification results, but also by its potential concerning model interpretability. This means analysing the temporal patterns uncovered through triclustering, regarding their domain relevance, clinical in this case, and their importance for the predictive model, by computing the pattern/feature importance according to the learnt classifier. To this aim, we chose to analyse the patterns discovered when triclustering the 3 CS dataset. The goal is to understand what are the most relevant features, what features appear together, and whether the patterns found to be relevant in the general model, putative patterns of the average patient, differ from those relevant to the specialized models, that should be group-specific, highlighting disease progression patterns of Slow, Neutral and Fast progressors.

Table 15 shows the characterization of triclusters obtained by the 4 learnt models (General, Slow, Neutral and Fast), where $|I|$, $|J|$ and $|K|$ represent the number of patients, features and time-points composing each tricluster. The patterns are represented by feature values across the time-points. Tables 11, 12, 13 and 14 depict the most important as used patterns by the classifier, ranked by their feature importance.

An overall analysis of the most important patterns discovered shows that the majority of the patterns refer to the last snapshot/time-point of the tricluster. This makes sense, since this is the snapshot closer to the target. However, other patterns not corresponding to the last snapshot that remain important, are also relevant for further clinical analysis, since these features can be relevant in identifying disease progression pattern leading to the need of NIV in a given time-window.

As expected, most of the important set of features used by the General model are the same as those discovered for the Neutral group, corresponding to the common tendency within the set of all patients, corresponding to the average patient (Neutral progressor). However, the highest values observed for the features in Neutral model expose the influence that the other groups (Slow and Fast progressors) created in the General model. Comparing patterns of Slow progressors with those of Neutral progressors, we can confirm that, as expected the values for the same set of features are smaller in the first.

When analysing the most important patterns used by the model learnt for Fast progressors, it is interesting to witness that these patterns are typically very different from those of the other models. This confirms the fact that Fast progressors are very different patients and can be useful to help understanding their unique progression patterns. Furthermore, a quick identification of Fast progressors, whose clinical condition degrades very quickly, through their progression patterns, can promote timely intervention and prolong survival.

5 Conclusions

We proposed a new methodology to learn predictive models using a triclustering-based classifier.

The key step uses TCtriCluster, an extension of triCluster, incorporating a temporal constraint when mining triclusters. This restriction was shown to be effective

Table 11 20 Best discovered patterns in the general model trained with all data (with 3 CS)

Rank	#Tricluster	TP	Feat. Import	Pattern
1	10	2	0.031960	ALS-FRS = 38
2	18	2	0.029058	ALS-FRS-R = 46
3	54	0	0.027954	MIP = 53.5705; MEP = 65.0705
4	53	0	0.023513	FVC = 88.2423; MEP = 66.6423
5	12	2	0.022829	ALS-FRS = 31
6	54	1	0.022564	MIP = 53.2824; MEP = 66.1471
7	35	2	0.018606	ALS-FRSsUL = 9; R = 12
8	13	2	0.017853	ALS-FRS = 30
9	53	1	0.017221	FVC = 89.15; MEP = 67.5577
10	55	1	0.012898	ALS-FRS = 24; ALS-FRS-R = 32
11	25	2	0.012194	ALS-FRS-R = 33
12	66	1	0.011938	ALS-FRS = 38
13	7	1	0.011871	ALS-FRS = 26; ALS-FRS-R = 34
14	15	2	0.011854	ALS-FRS = 27
15	69	1	0.011695	ALS-FRS = 27; R = 12
16	0	1	0.011637	ALS-FRS = 24
17	87	1	0.011274	ALS-FRSsUL = 12; ALS-FRSsLL = 12
18	29	2	0.011250	ALS-FRSb = 12; R = 12
19	22	2	0.010727	ALS-FRS-R = 35
20	83	1	0.010561	ALS-FRS-R = 46

Table 12 20 Best discovered patterns in the specialized model for Slow group (with 3 CS)

Rank	#Tricluster	TP	Feat. Import	Pattern
1	1	2	0.059431	ALS-FRS = 37; ALS-FRS-R = 45
2	2	2	0.043924	ALS-FRS-R = 45
3	120	0	0.027597	ALS-FRS = 37; ALS-FRS-R = 45
4	120	1	0.027366	ALS-FRS = 37; ALS-FRS-R = 45
5	46	2	0.026735	ALS-FRSb = 12; ALS-FRSr = 4; R = 12
6	71	2	0.026688	ALS-FRSsLL = 12; ALS-FRSr = 4
7	47	2	0.025992	ALS-FRSsLL = 12; R = 12
8	3	0	0.025549	ALS-FRS = 37
9	0	1	0.024103	ALS-FRS = 37; ALS-FRS-R = 45
10	4	0	0.023486	ALS-FRS-R = 45
11	62	1	0.023258	ALS-FRSsUL = 9; ALS-FRSr = 4; R = 12
12	31	2	0.023019	ALS-FRS-R = 12
13	2	0	0.022919	ALS-FRS-R = 45
14	1	1	0.022609	ALS-FRS = 37; ALS-FRS-R = 45
15	2	1	0.022548	ALS-FRS-R = 45
16	3	1	0.021562	ALS-FRS = 37
17	4	1	0.020758	ALS-FRS-R = 45
18	71	1	0.019888	ALS-FRSsLL = 12; ALS-FRSr = 4
19	60	1	0.019324	ALS-FRSsUL = 12; R = 12
20	55	1	0.019007	ALS-FRSsUL = 12; ALS-FRSr = 4

in improving the effectiveness of the predictive models, highlighting its importance when triclustering temporal data. We further show that triclustering-based classification enhances prediction with the potentialities of model interpretability, enabling the discovery of domain relevant temporal patterns, then used as features in the models.

As case study we used clinical three-way data and tackled the challenge of predicting the need for NIV in ALS patients within a time window of 90 days. We performed experiments using all patients and patients stratified by their disease progression rate as Slow, Neutral and Fast progressors.

The prognostic prediction results are promising when patient stratification is performed, specially for Neutral progressors. Concerning model interpretation, it was interesting to confirm the existence of group-specific patterns, corresponding to different disease progression patterns, then used by the specialized models as important features. In particular, Fast progressors have unique patterns, whose quick identification could help to improve prognosis, by antecipating NIV.

The proposed triclustering-based methodology can further be used to learn predictive models with different types of three-way-data, from other heterogeneous diseases or other domains.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work was partially supported by FCT funding through projects NEUROCLINOMICS2 (PTDC/EEI-SII/1937/2014) and iCare4U (LISBOA-01-0145-FEDER-031474 + PTDC/EME-SIS/31474/2017), and the LASIGE Research Unit (UIDB/00408/2020 + UIDP/00408/2020).

Table 13 20 Best discovered patterns in the specialized model for Neutral group (with 3 CS)

Rank	#Tricluster	TP	Feat. Import	Pattern
1	4	2	0.018618	ALS-FRS = 38; ALS-FRS-R = 46
2	260	2	0.016051	ALS-FRSsUL = 12; ALS-FRSsLL = 12
3	238	2	0.014534	ALS-FRSsUL = 12; ALS-FRSr = 4
4	5	2	0.013115	ALS-FRS = 35
5	291	2	0.013027	ALS-FRSsUL = 12; ALS-FRSr = 4; R = 12
6	551	1	0.012610	MIP = 58.355; MEP = 66.5
7	261	2	0.012074	ALS-FRSsUL = 12
8	1	2	0.011431	ALS-FRS = 29; ALS-FRS-R = 37
9	11	2	0.011142	ALS-FRS-R = 43
10	551	0	0.011140	MIP = 54.2; MEP = 63.8
11	3	2	0.011110	ALS-FRS = 33; ALS-FRS-R = 41
12	190	2	0.010652	ALS-FRSb = 12; ALS-FRSr = 4
13	550	0	0.010239	FVC = 87.3571; MEP = 65.7071
14	550	1	0.010007	FVC = 88.7429; MEP = 66.9928
15	2	2	0.009997	ALS-FRS = 31; ALS-FRS-R = 39
16	13	2	0.009706	ALS-FRS-R = 41
17	10	2	0.009301	ALS-FRS-R = 39
18	226	2	0.008809	ALS-FRSb = 12; ALS-FRSr = 4; R = 12
19	244	2	0.007572	ALS-FRSb = 12; R = 12
20	342	2	0.007409	ALS-FRSsLL = 12; R = 12

Table 14 Discovered patterns in the specialized model for Fast group (with 3 CS)

Rank	#Tricluster	TP	Feat. Import	Pattern
1	1	2	0.251100	R = 12
2	2	2	0.198099	ALS-FRSr = 3
3	2	1	0.100364	ALS-FRSr = 4
4	0	0	0.100323	R = 12
5	1	0	0.093527	R = 12
6	2	0	0.093015	ALS-FRSr = 4
7	1	1	0.091529	R = 12
8	0	1	0.072043	R = 12

Author details
[1]LASIGE, Faculdade de Ciências, Universiadade de Lisboa, Campo Grande, Lisbon, Portugal. [2]INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Alameda, Lisbon, Portugal. [3]Instituto de Medicina Molecular, Faculdade de Medicina, Universidade de Lisboa, Lisbon, Portugal.

References

1. Henriques, R., Madeira, S.C.: Triclustering algorithms for three-dimensional data analysis: A comprehensive survey. ACM Computing Surveys **51**(5), 95 (2019)
2. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM transactions on computational biology and bioinformatics **1**(1), 24–45 (2004)
3. Amar, D., Yekutieli, D., Maron-Katz, A., Hendler, T., Shamir, R.: A hierarchical bayesian model for flexible module discovery in three-way time-series data. Bioinformatics **31**(12), 17–26 (2015)
4. Kakati, T., Ahmed, H.A., Bhattacharyya, D.K., Kalita, J.K.: Thd-tricluster: A robust triclustering technique and its application in condition specific change analysis in hiv-1 progression data. Computational biology and chemistry **75**, 154–167 (2018)
5. Zhao, L., Zaki, M.J.: Tricluster: An effective algorithm for mining coherent clusters in 3d microarray data. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data. SIGMOD '05, pp. 694–705. ACM, New York, NY, USA (2005)
6. Heffernan, C., Jenkinson, C., Holmes, T., Macleod, H., Kinnear, W., Oliver, D., Leigh, N., Ampong, M.: Management of respiration in mnd/als patients: An evidence based review. Amyotrophic Lateral Sclerosis **7**(1), 5–15 (2006)
7. Chiò, A., Logroscino, G., Traynor, B., Collins, J., Simeone, J., Goldstein, L., White, L.: Global epidemiology of amyotrophic lateral sclerosis: a systematic review of the published literature. Neuroepidemiology **41**(2), 118–130 (2013)
8. Conde, B., Winck, J.C., Azevedo, L.F.: Estimating amyotrophic lateral sclerosis and motor neuron disease prevalence in portugal using a pharmaco-epidemiological approach and a bayesian multiparameter evidence synthesis model. Neuroepidemiology **53**(1-2), 73–83 (2019)
9. Andersena, S.A., Borasioc, G.D., de Carvalho, M., Chioe, A., Van Dammef, P., Hardimang, O., Kolleweh, K., Morrisoni, K.E., *et al.*: Efns guidelines on the clinical management of amyotrophic lateral sclerosis (mals)–revised report of an efns task force. European Journal of Neurology **19**, 360–375 (2011)
10. Bourke, S.C., Tomlinson, M., Williams, T.L., Bullock, R.E., Shaw, P.J., Gibson, G.J.: Effects of non-invasive ventilation on survival and quality of life in patients with amyotrophic lateral sclerosis: a randomised controlled trial. The Lancet Neurology **5**(2), 140–147 (2006)
11. Carreiro, A.V., Amaral, P.M., Pinto, S., Tomás, P., de Carvalho, M., Madeira, S.C.: Prognostic models based on patient snapshots and time windows: Predicting disease progression to assisted ventilation in amyotrophic lateral sclerosis. Journal of biomedical informatics **58**, 133–144 (2015)
12. Pires, S., Gromicho, M., Pinto, S., Carvalho, M., Madeira, S.C.: Predicting non-invasive ventilation in als patients using stratified disease progression groups. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 748–757 (2018). IEEE
13. Pires, S., Gromicho, M., Pinto, S., de Carvalho, M., Madeira, S.C.: Patient stratification using clinical and patient profiles: Targeting personalized prognostic prediction in als. In: International Work-Conference on Bioinformatics and Biomedical Engineering, pp. 529–541 (2020). Springer

Table 15 Characterization of Triclusters obtained with different learned models (with 3 CS)

General Model				
#Tricluster	$\|I\|$	$\|J\|$	$\|K\|$	Patterns
0	24	1	2	[ALS-FRS=28], [ALS-FRS=24]
7	25	2	2	[ALS-FRS=27; ALS-FRS-R=35], [ALS-FRS=26; ALS-FRS-R=34]
10	59	1	3	[ALS-FRS=38], [ALS-FRS=38], [ALS-FRS=38]
12	54	1	3	[ALS-FRS=35], [ALS-FRS=35], [ALS-FRS=31]
13	42	1	3	[ALS-FRS=36], [ALS-FRS=36], [ALS-FRS=30]
15	41	1	3	[ALS-FRS=33], [ALS-FRS=33], [ALS-FRS=27]
18	56	1	3	[ALS-FRS-R=46], [ALS-FRS-R=46], [ALS-FRS-R=46]
22	38	1	3	[ALS-FRS-R=41], [ALS-FRS-R=41], [ALS-FRS-R=35]
25	28	1	3	[ALS-FRS-R=40], [ALS-FRS-R=36], [ALS-FRS-R=33]
29	151	2	3	[ALS-FRSb=12; R=12], [ALS-FRSb=12; R=12], [ALS-FRSb=12; R=12]
35	20	2	3	[ALS-FRSsUL=9; R=12], [ALS-FRSsUL=9; R=12], [ALS-FRSsUL=9; R=12]
53	26	2	2	[FVC=88.2423; MEP=66.6423], [FVC=89.15; MEP=67.5577]
54	34	2	2	[MIP=53.5706; MEP=65.0706], [MIP=57.2824; MEP=66.1471]
55	22	2	2	[ALS-FRS=27; ALS-FRS-R=35], [ALS-FRS=24; ALS-FRS-R=32]
66	31	1	2	[ALS-FRS=38], [ALS-FRS=38]
69	23	2	2	[ALS-FRS=37; R=12], [ALS-FRS=37; R=12]
83	29	1	2	[ALS-FRS-R=46], [ALS-FRS-R=46]
87	44	2	2	[ALS-FRSsUL=12; ALS-FRSsLL=12], [ALS-FRSsUL=12; ALS-FRSsLL=12]
Specialized Model for Slow Progressors				
#Tricluster	$\|I\|$	$\|J\|$	$\|K\|$	Patterns
0	29	2	2	[ALS-FRS=38.0; ALS-FRS-R=46.0], [ALS-FRS=37.0; ALS-FRS-R=45.0]
1	26	2	3	[ALS-FRS=38.0; ALS-FRS-R=46.0], [ALS-FRS=37.0; ALS-FRS-R=45.0], [ALS-FRS=37.0; ALS-FRS-R=45.0]
2	13	1	3	[ALS-FRS-R=45.0], [ALS-FRS-R=45.0], [ALS-FRS-R=45.0]
3	13	1	2	[ALS-FRS=37.0], [ALS-FRS=37.0]
4	14	1	2	[ALS-FRS-R=45.0], [ALS-FRS-R=45.0]
31	58	2	3	[ALS-FRSb=12.0; ALS-FRSr=4.0], [ALS-FRSb=12.0; ALS-FRSr=4.0], [ALS-FRSb=12.0; ALS-FRSr=4.0]
46	51	3	3	[ALS-FRSb=12.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSb=12.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSb=12.0; ALS-FRSr=4.0; R=12.0]
47	51	2	3	[ALS-FRSb=12.0; R=12.0], [ALS-FRSb=12.0; R=12.0], [ALS-FRSb=12.0; R=12.0]
55	17	2	2	[ALS-FRSsUL=12.0; ALS-FRSr=4.0], [ALS-FRSsUL=12.0; ALS-FRSr=4.0]
60	14	2	2	[ALS-FRSsUL=12.0; R=12.0], [ALS-FRSsUL=12.0; R=12.0]
62	15	3	2	[ALS-FRSsUL=9.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSsUL=9.0; ALS-FRSr=4.0; R=12.0]
71	14	2	3	[ALS-FRSsLL=12.0; ALS-FRSr=4.0], [ALS-FRSsLL=12.0; ALS-FRSr=4.0], [ALS-FRSsLL=12.0; ALS-FRSr=4.0]
120	28	2	2	[ALS-FRS=37.0; ALS-FRS-R=45.0], [ALS-FRS=37.0; ALS-FRS-R=45.0]
Specialized Model for Neutral Progressors				
#Tricluster	$\|I\|$	$\|J\|$	$\|K\|$	Patterns
1	59	2	3	[ALS-FRS=33.0; ALS-FRS-R=41.0], [ALS-FRS=31.0; ALS-FRS-R=38.0], [ALS-FRS=29.0; ALS-FRS-R=37.0]
2	48	2	3	[ALS-FRS=33.0; ALS-FRS-R=41.0], [ALS-FRS=33.0; ALS-FRS-R=41.0], [ALS-FRS=31.0; ALS-FRS-R=39.0]
3	53	2	3	[ALS-FRS=35.0; ALS-FRS-R=43.0], [ALS-FRS=35.0; ALS-FRS-R=43.0], [ALS-FRS=33.0; ALS-FRS-R=41.0]
4	12	2	3	[ALS-FRS=38.0; ALS-FRS-R=46.0], [ALS-FRS=36.0; ALS-FRS-R=44.0], [ALS-FRS=37.0; ALS-FRS-R=45.0]
5	17	1	3	[ALS-FRS=37.0], [ALS-FRS=35.0], [ALS-FRS=35.0]
10	18	1	3	[ALS-FRS=37.0], [ALS-FRS=35.0], [ALS-FRS=35.0]
11	14	1	3	[ALS-FRS-R=45.0], [ALS-FRS-R=43.0], [ALS-FRS-R=43.0]
13	19	1	3	[ALS-FRS-R=46.0], [ALS-FRS-R=41.0], [ALS-FRS-R=41.0]
190	18	2	3	[ALS-FRSb=12.0; ALS-FRSr=4.0], [ALS-FRSb=12.0; ALS-FRSr=4.0], [ALS-FRSb=12.0; ALS-FRSr=4.0]
226	64	3	3	[ALS-FRSb=12.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSb=12.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSb=12.0; ALS-FRSr=4.0; R=12.0]
238	19	1	3	[ALS-FRSb=12.0], [ALS-FRSb=12.0], [ALS-FRSb=12.0]
244	67	2	3	[ALS-FRSb=12.0; R=12.0], [ALS-FRSb=12.0; R=12.0], [ALS-FRSb=12.0; R=12.0]
260	20	2	3	[ALS-FRSsUL=12.0; ALS-FRSsLL=12.0], [ALS-FRSsUL=12.0; ALS-FRSsLL=12.0], [ALS-FRSsUL=12.0; ALS-FRSsLL=12.0]
261	19	1	3	[ALS-FRSsUL=12.0], [ALS-FRSsUL=12.0], [ALS-FRSsUL=12.0]
291	19	3	3	[ALS-FRSsUL=12.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSsUL=12.0; ALS-FRSr=4.0; R=12.0], [ALS-FRSsUL=12.0; ALS-FRSr=4.0; R=12.0]
342	20	2	3	[ALS-FRSsLL=12.0; R=12.0], [ALS-FRSsLL=12.0; R=12.0], [ALS-FRSsLL=12.0; R=12.0]
550	14	2	2	[FVC=87.3571; MEP=65.7071], [FVC=88.7429; MEP=66.9929]
551	20	2	2	[MIP=54.2; MEP=63.8], [MIP=58.355; MEP=66.5]
Specialized Model for Fast Progressors				
#Tricluster	$\|I\|$	$\|J\|$	$\|K\|$	Patterns
0	10	1	2	[R=12.0], [R=12.0]
1	8	1	3	[R=12.0], [R=12.0], [R=12.0]
2	8	1	3	[ALS-FRSr=4.0], [ALS-FRSr=4.0], [ALS-FRSr=3.0]

14. Matos, J., Pires, S., Aidos, H., Gromicho, M., Pinto, S., de Carvalho, M., Madeira, S.C.: Unravelling disease presentation patterns in als using biclustering for discriminative meta-features discovery. In: International Work-Conference on Bioinformatics and Biomedical Engineering, pp. 517–528 (2020). Springer
15. Divina, F., Pontes, B., Giráldez, R., Aguilar-Ruiz, J.S.: An effective measure for assessing the quality of biclusters. Computers in biology and medicine **42**(2), 245–256 (2012)
16. Madeira, S.C., Teixeira, M.C., Sa-Correia, I., Oliveira, A.L.: Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm. IEEE/ACM Transactions on Computational Biology and Bioinformatics **7**(1), 153–165 (2008)
17. ENCALS: ALS Functional Rating Scale Revised (ALS-FRS-R). Version: May 2015
18. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)