

Merging Temporally-Related Clinical Data from Patients with Amyotrophic Lateral Sclerosis using Constraint-Based Hierarchical Clustering

Pedro M. T. Amaral^{*1}, Susana Pinto^{**2}, Mamede de Carvalho^{***2}, Pedro Tomás^{†1}, and Sara C. Madeira^{‡1}

¹INESC-ID, Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal

²Neuromuscular Unit, Institute of Molecular Medicine and Faculty of Medicine, University of Lisbon, Lisbon, Portugal

Abstract. Amyotrophic Lateral Sclerosis (ALS) is a devastating neurodegenerative disease causing motor impairment. It is the third in the ranking of incidence of neurodegenerative diseases. It has no cure and it makes survival time after diagnosis very limited. Since most patients die from respiratory failure, predicting the onset of hypoventilation is crucial to prolong survival and improve the patients' quality of life.

This paper describes the preprocessing of clinical data to be used in a later phase to predict the need for non-invasive ventilation in patients with ALS. We propose a hierarchical clustering based approach to preprocess longitudinal clinical data. We use this approach to cluster temporally-related clinical exams, while constructing the instances needed to train the machine learning algorithms. We consider that an instance is a snapshot of a patient's condition at a given time window, which is characterized by a set of attributes resulting from merging clinical exams. We compare our approach with the standard approach followed by clinicians, and show the superiority of our results.

Keywords: Amyotrophic Lateral Sclerosis, Preprocessing, Clinical Data, Constraint-Based Hierarchical Clustering.

1 Introduction

Amyotrophic Lateral Sclerosis (ALS) is a neurodegenerative disease characterized by a progressive muscular paralysis. The swift progress of this disorder causes a denervation of the critical skeletal muscles, namely the limb, axial, bulbar and respiratory muscles. This leads to progressive weakness, and ultimately death [5]. It is currently the third in the ranking of incidence of neurodegenerative diseases.

* pedro.m.t.amaral@ist.utl.pt

** susana.c.pinto@fm.ul.pt

*** mamedealves@fm.ul.pt

† pedro.tomas@inesc-id.pt

‡ sara.c.madeira@ist.utl.pt

ALS is a devastating disease from the point of view of motor impairment while almost, if not completely, innocuous from the point of view of cognitive decline [9]. It has no cure and its causes are yet to be discovered. As such, maintaining the patients' quality of life is nowadays of major relevance.

Respiratory complications account for the majority of deaths in ALS. Patients usually succumb from hypoventilation with hypoxaemia and hypercapnia, often associated with aspiration and pneumonia [7]. Predicting the onset of hypoventilation is, therefore, of major importance, in order to prevent the rapid deterioration of the patients' quality of life, as well as to prolong their survival time [10]. Ventilation management can be provided either by non-invasive ventilation (NIV), through a mask, or invasive ventilation, via tracheotomy. Non-invasive ventilation is preferable since it improves survival and quality of life [10], as opposed to invasive ventilation, which may cause the patients to develop a profound state of motor paralysis, termed the "totally locked-in state". For this reason, clinicians use mostly NIV to compensate respiratory insufficiency cases.

Some research has already been done in the ALS field. However, since the clinical results achieved are not completely related to the problem we address in this paper, they are just left as a reference (see [4, 6, 8, 3]).

This paper describes the preprocessing of clinical data, which will be used, in a later phase, to predict the need for non-invasive ventilation in patients with ALS. Throughout the entire learning process, we use clinical data containing respiratory tests and neurophysiological data, collected by the Neuromuscular Unit at the Molecular Medicine Institute of Lisbon, during a period of 10 years.

For data preprocessing, we propose a hierarchical clustering approach to group temporally related clinical exams, while constructing the instances needed for the supervised learning. This algorithm addresses problems found in the baseline approach that is frequently used by clinicians. The baseline technique consists in clustering the exam observations according to the date of execution of a particular exam – *pivot exam* – that is carried out at the regular disease-specific appointments. We show that the proposed hierarchical clustering approach outperforms the baseline strategy, and that it is crucial for the construction of informative and more realistic training instances. We consider an instance as a snapshot of a patient's condition at a given time window, characterized by a collection of results from clinical exams, and a label indicating whether the patient required non-invasive ventilation in that period of time. Missing values result from exams that were not performed in the considered time window.

The paper is organized as follows: Section 2 introduces the dataset used in our experiments. Section 3 describes the data preprocessing phase we applied to the data. Section 4 covers the baseline strategy, and Section 5 tackles the hierarchical clustering algorithm. Finally, Section 6 shows and discusses the results obtained with both these approaches.

2 Data Description

The dataset used in this work has clinical data from real ALS patients, containing: (a) demographic, patient, and disease characteristics; (b) clinical evaluations; (c) respiratory tests; and (d) neurophysiological data. We used data in (a) together with a set of disease-relevant exams from categories (b), (c) and (d). In the course of the patient follow ups, any respiratory insufficiency was compensated with NIV. In the original dataset provided by clinicians, each exam was placed in a separate partition, and arranged in a temporal fashion (see Figure 1).

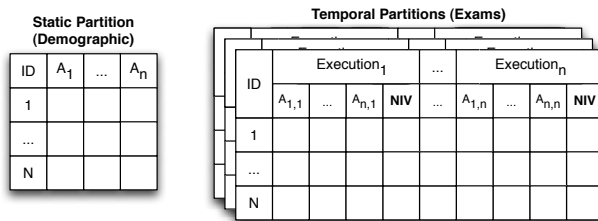


Fig. 1. Initial Dataset.

The data to be analyzed consisted of static information (the patients’ demographic data) and temporal information (the results collected after the execution of a set of disease-specific exams). Each row represented a different patient. In the static partition, the columns represented the demographic attributes that were relevant for the disease. The temporal partitions, contained a set of (exam) executions. We define the concept of execution as an unique observation of a given exam. In this context, each execution of the same exam tested the same set of attributes: A_1, \dots, A_n , and NIV (as illustrated in Figure 1). The NIV attribute represented the class of our learning problem. The columns in the temporal partitions were, however, not entirely filled for every patient. In fact, for each patient, the information available was only that relative to the exam executions held. Figure 2 clarifies this issue by showing a simplified example of an exam partition, where the attributes belonging to each execution were omitted. The tick symbol (✓) in a position (x, y) means that patient x held Execution _{y} .

	Execution_1	Execution_2	Execution_3	Execution_4
1	✓	✓	✓	
2	✓	✓		
3	✓	✓	✓	✓

Fig. 2. Example of an exam partition with up to four executions per patient.

In this example, patient 1, for instance, held the exam three times, while patient 2 held it only twice. We note that, Execution _{i} occurred temporally prior to Execution _{$i+1$} .

The original clinical dataset had 517 patients, 5 exam partitions, and 1 demographic partition. The total number of attributes before the preprocessing phase was 63. Each exam execution had a class label (NIV) indicating whether the patient required non-invasive ventilation by the time of each examination. Considering again Figure 2, a possible scenario would be that, patient 2 had only required NIV by the time the exam was held for the second time, resulting in: Execution_1 (NIV: $\neg v$) and Execution_2 (NIV: v), where v is used to denote that a patient required non-invasive ventilation, and $\neg v$ is used if the patient could breathe without help.

3 Data Preprocessing

Data, as described in Section 2, were not ready to be analysed for two reasons. First, the same attribute appeared more than once in each row. This was due to the fact that the exam information was arranged by execution, and each execution from the same exam tested the same set of properties. This makes, for instance, a certain attribute A_1 to be present in all of the executions from a particular exam: $A_{1,1}$ for Execution_1, $A_{1,2}$ for Execution_2, etc. (see Figure 1).

The format in which data was arranged suggested a temporal analysis using temporal series. However, our objective with this work was to analyze the learning problem, and to define a reliable decision boundary. We thus focused on the design of a classification model able to detect whether a patient needs non-invasive ventilation. This is still of extreme relevance to the medical doctors. The temporal problem will be tackled in future work.

In this context, we discarded the temporality of data by splitting the exam executions belonging to each patient (see Figure 3). The resulting dataset contained only one exam execution per patient. Since each execution belonged to a different patient, the population size increased (in the example in Figure 3 it increased from 3 to 5 instances).

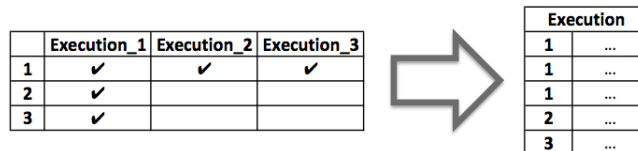


Fig. 3. Execution split example.

The second reason that prevented us from using the original dataset was related to the way the exam information was stored. While scattered in different partitions, it would be difficult, for the supervised learning algorithms, to achieve conclusions that combined information from more than one exam. In fact, if we separately applied a classifier to each exam partition, we would be imposing that the results from each exam were independent from each other. In other words, once we obtained the results for each exam partition individually, it would be

hard to relate them and, therefore, arrive to conclusions that could combine the information provided by the different exams. Due to this, we decided to merge the exam partitions into a single dataset. For this purpose, we tested two strategies using the date of occurrence of each exam execution to perform the merge.

The strategy described in Section 4 corresponds to the approach usually used by clinicians to perform the execution merge. We term it the baseline strategy. Section 6 points out the weaknesses of this approach, and demonstrates how they are overcome by the hierarchical clustering solution we propose in this paper, described in Section 5.

The purpose of the merge was to combine the executions belonging to different exams that were more temporally related. The combinations obtained represent snapshots of the patients' condition at a certain period of their lives, that is, their condition during a certain time window. Figure 4 presents the objective of the merge. In this illustration, we consider four exams (A, B, C, D) and the instant of time when they occurred.



Fig. 4. Exam execution combinations.

We notice that exam A was held twice. The rectangles indicate the combinations that merge the exam executions that are more temporally related. It is also important to note that, since we want to obtain coherent snapshots, two exam executions with incompatible class values (v and $\neg v$) cannot be combined into the same snapshot.

4 Baseline Strategy (With Pivot Exam)

The baseline strategy was proposed by the medical doctors, and corresponds to the common practice among the clinicians. It consists in using an exam, typically the one with the highest number of executions per patient, as pivot. The baseline version we present was adapted, in order to meet our problem-specific requirements.

The merging process is simple, and is composed by just one rule: every execution of a non-pivot exam held between two pivot executions is included inside one combination (or cluster). Figure 5 illustrates this process, where A^* is the pivot exam. The timeline contains the instants at which the exams were held.

In this example, there were found three combinations (S_1 , S_2 and S_3). The criterion used was the following: Exam B at instant 4 (B_4), and exam C at instant 5 (C_5) were the only ones that occurred between the two pivot exam executions (A_2^* and A_6^*). Due to this, they were placed inside the same combination (S_2). The previously mentioned merging rule does not cover situations in which an

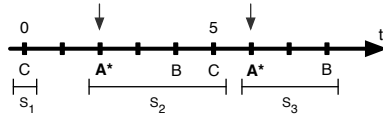


Fig. 5. Exam clustering using the baseline strategy where a pivot exam is used.

exam was not held between two pivots, i.e., when it was executed before the first, and after the last A^* . There are two possible approaches to deal with this situation: to keep, or to delete these exams. The latter approach would lead to the erase of a considerable amount of relevant data, and to the construction of poorer final files. For this reason, we decided to keep these entries, by placing them inside new combinations. Examples of these are C_0 (placed in S_1) and B_8 (clustered in S_3).

The careful reader may be wondering what we did to the pivot exam executions. As Figure 5 illustrates, each A^* was placed inside the combination that used it as the left boundary. A_2^* was assigned to S_2 , and A_6^* to S_3 . The pivot exams were not discarded, since they also provided relevant information for the learning process.

Bear in mind that, each combination must have the same class value for each exam execution. As such, if between two pivot executions there existed incompatible examinations, that is, examinations with different class labels, more than one combination was generated. In the example in Figure 5, if, for instance, B_4 had V as NIV, and C_5 had $\neg V$, they could not belong to the same cluster. One of the executions would need to be placed in a new combination (S_2' , for example).

Despite being simple to understand and to implement, this algorithm presents several flaws, which are explored in Section 6.

5 Hierarchical Clustering Strategy

The second strategy, and the one we propose in this paper, is based on bottom-up hierarchical clustering, and produces, as output, a dendrogram – a tree structure that represents a sequence of nested clusters [2]. Our algorithm belongs to the linkage-based algorithm’s class, which begins by placing every element of the input dataset into its own cluster, and then repeatedly merges the ‘closest’ clusters [1]. We use single-linkage as closeness metric.

Our approach differs from standard hierarchical clustering since it incorporates two constraints: (a) two executions from the same exam cannot be combined, and (b) two class-incompatible executions cannot be merged. These constraints are checked whenever two clusters are close enough to be merged.

Lets see how the constraint-based hierarchical clustering works, by starting with a constraint-free version of the algorithm. Consider the exams on the left side in Figure 6. There are represented three exams (A, B, C), the instant of time they occurred, and the NIV value a certain patient had by the time of their execution. On the right side, there is represented the dendrogram that results

from the application of a standard hierarchical clustering algorithm. The labels on the right of the dendrogram indicate the maximum diameter of the clusters comprised in each level.

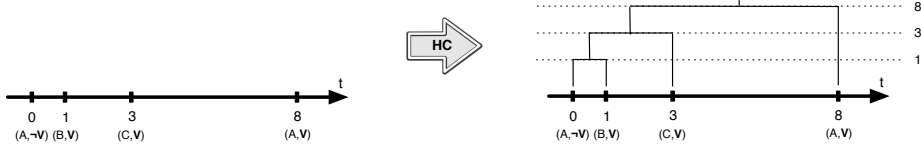


Fig. 6. Exam clustering using a standard hierarchical clustering (HC) algorithm.

A standard hierarchical clustering algorithm starts by placing each instance in a different cluster. It then merges the clusters that are closer to each other. In the example, clusters $\{A_0\}$ and $\{B\}$ are at a distance of 1 ($\overline{\{A_0\}\{B\}} = 1$) from each other, which makes them the closest, since $\overline{\{A_0\}\{C\}} = 3$, $\overline{\{B\}\{C\}} = 2$, etc.. Therefore, the first merged cluster to be formed is $\{A_0, B\}$. Since the standard algorithm only stops when there exists just one cluster containing all the others, it restarts the search for the closest clusters. It finds that $\overline{\{A_0, B\}\{C\}} = 2$ (recall that we use single-linkage as closeness metric), which makes $\{A_0, B\}$ and $\{C\}$ the next candidates for the merge. Afterwards, since there are just two clusters $\{A_0, B, C\}$ and $\{A_8\}$, which are at a distance of 5 instants, the algorithm merges them and stops. The labels on the right indicate the maximum diameter of the clusters that are present in each level. The second level (label 3), for instance, comprises clusters $\{A_0, B, C\}$ and $\{A_8\}$, with diameters 3 and 0, respectively.

The problem with the standard approach is that it generates invalid clusters, containing more than one execution of the same exam, and executions with different values for NIV. The solution we propose for this problem is the introduction of constraints to the standard algorithm. However, these need to be inserted in a specific point of the method. We stress this by showing the impact of two approaches, which are the introduction of constraints:

- After the algorithm is complete (that is, after the dendrogram is built).
- Before merging the two closest clusters.

Lets explore the first alternative. This approach starts by the most general cluster (or level), that is, the one that comprises all the others, and checks its validity. If it respects the constraints, nothing occurs. Otherwise, the two clusters merged by the most general one are separated. The introduction of constraints proceeds until all clusters, even the ones with one element, are verified. Figure 7 shows the impact of applying this approach to the example in Figure 6.

As it is possible to observe, clusters $\{A_0, B, C\}$ and $\{A_8\}$ (on the left) are separated (on the right), since they are class-incompatible, and have two executions of exam A. All the other clusters are also splitted due to the class-compatibility constraint. It is easy to conclude that this approach, by not being able to combine any exam execution, fails to meet the merge objective described in Section 3. For this reason it was discarded.



Fig. 7. Standard HC - constraints introduced after dendrogram is built.

The second approach works as follows: Before merging any two closest clusters, the constraints are verified. If the clusters are incompatible, the merge does not occur. Whenever this happens, the algorithm tries to find the next compatible closest candidates, in order to proceed with the merge.

The introduction of such modifications called for a new terminating condition: the algorithm now stops when there are no clusters able to produce valid combinations. For our specific problem, however, we slightly altered this terminating condition. Since we did not want to generate overlapping dendrograms, that is, dendrograms whose lines, if infinitely prolonged upwards, intersected between each other, we considered that the algorithm stopped whenever there did not exist two compatible adjacent clusters. In Figure 7, $\{A_0\}$ and $\{B\}$ are two incompatible adjacent clusters. $\{B\}$ and $\{A_8\}$, on the other hand, despite being compatible, could not be integrated in the same dendrogram if $\{C\}$ had $\neg V$ as NIV value.

Figure 8 illustrates the application of the second approach to the example in analysis. According to this, $\{A_0\}$ and $\{B\}$, in spite of being the closest, are not merged due to their incompatibility. This allows the creation of cluster $\{B,C\}$, and afterwards $\{B,C,A_8\}$. The results produced by this approach respect the merging objective, which is why it was selected to be part of our method.

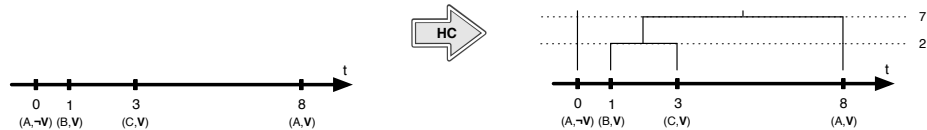


Fig. 8. Standard HC - constraints introduced before merging the two closest clusters.

A careful reader may have noticed the difference between the results produced by our constraint-based algorithm and the ones from the standard hierarchical clustering method. While the latter produces just one, our approach is able to generate a variable amount of dendrograms. In Figure 7, for instance, there are created four dendrograms, each composed by just a unitary cluster. In Figure 8, we have two dendrograms: one composed by execution A_0 , and other composed by B , C and A_8 .

Finally, let's explain how we use the dendrograms to produce the files for the preprocessing phase. As you can see, each dendrogram has a set of cut regions, which correspond to the different levels. Each level, on the other hand, contain one

or more clusters of executions, which were held within a certain time window. The size of the time window comprised by each level is described in its label. Since we did not know, *a priori*, which time window was the best to merge the executions, we used a set of different values, and checked the quality of the files they helped to produce.

The cuts were performed in the following way: For each time window value, we performed the cut of each dendrogram using as reference the level whose label was most proximate, by default, of that value. Consider Figure 8. If, for instance, we considered an interval of 4 instants for the time window, we would perform the cut in the first level (label 2), and obtain three clusters: $\{A_0\}$, $\{B, C\}$ and $\{A_8\}$. The analysis of the best values for the time windows we tested is out of the scope of this paper.

Before closing this section, it is important to note that, a distinct instance of the hierarchical clustering algorithm was executed for each patient. Only by doing so, we were able to capture snapshots that belonged to a single patient. The prove for this is left as an exercise.

6 Results and Discussion

In spite of being simpler, the baseline strategy does not produce the expected results. In the example in Figure 4, it produces clusters $\{A_0, B, C\}$ and $\{A_4, D\}$, thus not obtaining the combinations that contain the exams that are temporally more related. The same occurs in Figure 5: execution C_0 should be merged with A_2^* , and C_5 should be merged with A_6^* . In that example, the best clusters would be: $\{C_0, A_2^*\}$, $\{B_4, C_5, A_6^*\}$ and $\{B_8\}$.

The explanation for these results is rather simple. Since we are dealing with clinical data from real patients, we must not expect the information to have a fixed format or to follow a certain pattern. This means that, if a patient has to execute a set of exams between two medical appointments, we cannot expect him to perform them all in a very small period of time. Moreover, we cannot expect the patients to execute all the exams right after or right before a medical appointment. If such prediction was possible, we could improve this algorithm by inserting in each cluster the pivot exam that occurred closer to it, instead of using the one that served as left boundary. However, such improvement would go beyond the simplicity of the specification of the approach followed by the clinicians, which would then become more and more specific to our problem.

We cannot rely on any theory nor short-path. The uncertainty of the data can produce situations such as the one illustrated in Figure 9, where even the most biased improvement to the baseline approach could produce bad results.

In this example, the solution would only require two clusters ($\{B, A_2^*, C\}$, $\{D, A_2^*, E\}$) and not three, as obtained with the pivot exam strategy.

Other problems were also found during the implementation of the baseline approach. On the one hand, there existed patients without any execution of the pivot exam, for whom this approach was not applicable. On the other hand, there existed situations where more than one execution of the same exam occurred

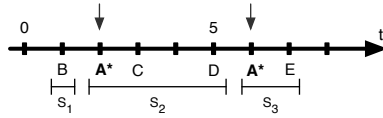


Fig. 9. Baseline clustering – poor temporally related combinations.

between two pivot exam executions (consider, for instance, two executions of exams B or C between the two A* in Figure 5). Since these cases are not covered by the baseline specification, the solution is either to discard one of the executions, or to place it in another combination. While the first case discards possibly relevant information, the second adds a significant amount of combinations with just a few (one or two) executions, which would contribute to the increase of the number of null values in the final file.

All of the previously mentioned problems do not persist when we use the hierarchical clustering approach. Our dataset analysis showed that this strategy produced results that were more coherent, and with a much lower percentage of missing values. For this reason, it was selected to integrate the preprocessing task that was applied to the clinical data.

For the time window, we decided, with the approval of the medical doctors, to use an interval of 100 days. Figure 10 shows the results obtained using this interval and both strategies. The first column has the ids of the patients, and the remaining columns indicate the date of execution of some disease-relevant exams (ALS-FRS (pivot), NPO, RFT, PHRENIC and ADMEMG). A missing value indicates that the patient did not execute a certain exam during the time window.

Hierarchical Clustering Algorithm						Baseline Approach					
Name	ALS-FRS	NPO	RFT	PHRENIC	ADMEMG	Name	ALS-FRS	NPO	RFT	PHRENIC	ADMEMG
1	28/6/10	7/7/10	17/5/10	7/7/10		1			17/5/10		
1	6/10/10		20/9/10			1	28/6/10	7/7/10	20/9/10	7/7/10	
1	5/1/11			1/1/11		1	6/10/10			1/1/11	
1	5/5/11	4/4/11	28/3/11	5/4/11		1	5/1/11	4/4/11	28/3/11	5/4/11	
1	6/7/11			1/7/11		1	6/7/11				
4			15/12/99	14/1/00		1	5/5/11			1/7/11	
4	2/2/00					4			15/12/99	14/1/00	
4	27/3/00					4	2/2/00				
4	8/5/00			1/5/00		4	8/5/00				
6	7/6/06	11/7/06	7/6/06	25/5/06		4	27/3/00			1/5/00	
6	17/8/06	16/8/06				6	8/5/00				
6	9/11/06	20/9/06				6	27/3/00			1/5/00	
7	25/10/99		13/10/99	1/11/99	26/8/99	6	7/6/06	11/7/06	7/6/06		
7	24/1/00		5/1/00			6	17/8/06	16/8/06			
						6	9/11/06	20/9/06			
						6	17/8/06	20/9/06			
						7			13/10/99		26/8/99
						7	24/1/00				
						7	25/10/99		5/1/00	1/11/99	

Fig. 10. Results - Hierarchical Clustering solution vs. Baseline approach.

The first thing to notice is the number of combinations found by each algorithm. While the hierarchical clustering solution captures 14 snapshots, the baseline approach finds 18, showing also a larger amount of null values. Regard-

ing the quality of the clusters, the former strategy is the one that produces the best results. By analyzing the snapshots captured for patient 7, we can verify that the baseline algorithm groups the ALS-FRS exam that occurred on 25/10/99 with the RFT exam on 5/1/00, which is much more temporally distant than the execution of this same exam that occurred on 13/10/99. This happened since the pivot exam execution (ALS-FRS) occurred after the RFT execution that was closest to it. This behavior represents a real example of the problems discussed in the beginning of this section. Figure 10 contains more examples that emphasize the superiority of our approach, which, due to space constraints, will not be further explored in this paper.

After the preprocessing phase, the dataset consisted of a single file, 2694 patients, and 34 attributes (including the id and the class attribute). Some attributes, such as the date at which the exams were held, were discarded, since they did not provide useful information to the subsequent analysis. Regarding the class distribution, there were 999 patients classified as requiring non-invasive ventilation (v), and 1695 that could breathe without help ($\neg v$).

7 Conclusions and Future Work

ALS is a fatal disease that degrades significantly the patients' quality of life. The most frequent cause of death from this impairment is related to respiratory failures resulting from the weakening of respiratory muscles. The most effective ventilation management technique is non-invasive ventilation, which simultaneously improves the survival and the quality of life of patients.

This paper describes the preprocessing of clinical data that will be used, in a later phase, to predict the need for non-invasive ventilation in patients with ALS. We used data from patients with ALS containing information on a set of disease-related exams. In the preprocessing task, we removed the temporality of the data, and merged all the exam information into a single file. To this end, and in order to group the most temporally related exams, we tested two strategies: one using an exam as pivot, corresponding to the standard practice among clinicians, and another based on hierarchical clustering, proposed in this paper. A careful analysis of the quality of the results produced by these approaches found several flaws in the former's specification, and showed that the latter was much more effective.

As future work, we will proceed with the static analysis of the data. To this end, some state of the art machine learning methods, including supervised classification and feature selection algorithms, will be applied. In a final phase, we intend to predict the time that a patient breathing without help will take to develop the characteristics of a patient requiring NIV.

Acknowledgments

This work was partially supported by FCT (Fundação para a Ciência e a Tecnologia) under projects: PTDC/EIA-EIA/ 111239/2009 (NEUROCLINOMICS

- Understanding NEUROdegenerative diseases through CLINical and OMICS data integration) and PEst-OE/EEI/LA0021/2011.

References

1. Ackerman, M., Ben-David, S., Loker, D.: Characterization of Linkage-based Clustering. In: Proceedings of the Conference on Learning Theory. p. 12. COLT (2010), <http://seed.ucsd.edu/joomla/colt2010/119Ackerman.pdf>
2. Anders, K.: A hierarchical graph-clustering approach to find groups of objects. In: Proceedings 5th Workshop on Progress in Automated Map Generalization. pp. 1–8. Citeseer (2003), <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.9973&rep=rep1&type=pdf>
3. Baumann, F., Henderson, R.D., Morrison, S.C., Brown, M., Hutchinson, N., Douglas, J.a., Robinson, P.J., McCombe, P.a.: Use of respiratory function tests to predict survival in amyotrophic lateral sclerosis. Amyotrophic lateral sclerosis : official publication of the World Federation of Neurology Research Group on Motor Neuron Diseases 11(1-2), 194–202 (Jan 2010), <http://www.ncbi.nlm.nih.gov/pubmed/19452343>
4. Carvalho, M., Pinto, S., Swash, M.: Paraspinal and limb motor neuron involvement within homologous spinal segments in ALS. Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology 119(7), 1607–13 (Jul 2008), <http://www.ncbi.nlm.nih.gov/pubmed/18468480>
5. Cudkowicz, M., Qureshi, M., Shefner, J.: Measures and markers in amyotrophic lateral sclerosis. NeuroRx : the journal of the American Society for Experimental NeuroTherapeutics 1(2), 273–83 (Apr 2004), <http://www.ncbi.nlm.nih.gov/pubmed/15717028>
6. Gupta, P.K., Prabhakar, S., Sharma, S., Anand, a.: A predictive model for amyotrophic lateral sclerosis (ALS) diagnosis. Journal of the neurological sciences (Sep 2011), <http://www.ncbi.nlm.nih.gov/pubmed/21907354>
7. Heffernan, C., Jenkinson, C., Holmes, T., Macleod, H., Kinnear, W., Oliver, D., Leigh, N., Ampong, M.A.: Management of respiration in mnd/als patients: an evidence based review. Amyotrophic lateral sclerosis official publication of the World Federation of Neurology Research Group on Motor Neuron Diseases 7(1), 5–15 (2006), <http://ovidsp.ovid.com/ovidweb.cgi?T=JS&CSC=Y&NEWS=N&PAGE=fulltext&D=emed7&AN=2006140699>
8. Pinto, S., Turkman, A., Pinto, A., Swash, M., de Carvalho, M.: Predicting respiratory insufficiency in amyotrophic lateral sclerosis: the role of phrenic nerve studies. Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology 120(5), 941–6 (May 2009), <http://www.ncbi.nlm.nih.gov/pubmed/19342290>
9. Stein, J., Schettler, T., Rohrer, B., Valenti, M.: Environmental Threats to Healthy Aging With a Closer Look at Alzheimer’s & Parkinson’s Diseases. Greater Boston Physicians for Social Responsibility and Science and Environmental Health Network (2008), http://www.agehealthy.org/pdf/GBPSRSEHN_HealthyAging1017.pdf
10. Wijesekera, L.C., Leigh, P.N.: Amyotrophic lateral sclerosis. Orphanet journal of rare diseases 4, 3 (Jan 2009), <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2656493&tool=pmcentrez&rendertype=abstract>