Mining Coherent Evolution Patterns in Education through Biclustering

André Carlos Rita do Vale

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Sara Alexandra Cordeiro Madeira

Prof. Cláudia Martins Antunes

Examination Committee

Chairperson: Prof. Joaquim Armando Pires Jorge

Supervisor: Prof. Sara Alexandra Cordeiro Madeira

Member of the Committee: Prof. Alexandre Paulo Lourenço Francisco

November 2014

Resumo

Com a expansão dos sistemas de informação e o aumento do interesse na área da educação nestes últimos anos, a quantidade de dados educacionais explodiu significativamente. No seguimento, surgiu uma nova área – Descoberta de Informação em Dados Educacionais, em inglês, *Educational Data Mining* (EDM). O EDM foca-se no desenvolvimento de métodos que permitem explorar os vários tipos de dados que provêem de contextos educacionais.

A previsão do desempenho dos estudantes já tem sido abordada por várias técnicas em EDM, mas a combinação de técnicas supervisionadas com as não supervisionadas apareceu recentemente como uma nova ferramenta para melhorar os resultados. Nesta dissertação, estudámos a inclusão de uma técnica não supervisionada que tem sido aplicada com sucesso em áreas como as de expressão genética e a recuperação de informação, mas nunca foi aplicada em dados educacionais - o *Biclustering*.

Apresentámos uma metodologia que nos permite usar os algoritmos de *Biclustering* em dados educacionais para obter novos padrões e usar esses resultados como complemento para a classificação. Em particular, usando matrizes com notas de estudantes da Licenciatura em Engenharia Informática e de Computadores (LEIC) do *Instituto Superior Técnico* conseguimos antecipar a média do Mestrado em Engenharia Informática e de Computadores (MEIC) desses mesmos estudantes.

Com a aplicação desta nova técnica conseguimos melhorar a precisão dos classificadores, de uma forma similar a outras técnicas anteriormente utilizadas, encontrando novos tipos de padrões que até aqui nunca tinham sido descobertos.

Abstract

With the expansion of information systems and the increased interest in the education field, the quantity of data about education has exploded along with a new field - Educational Data Mining (EDM). The focus of EDM is the development of methods for exploring the types of data that come from an educational context.

Predicting students' performance has been approached by several techniques, but the combination of supervised and non-supervised techniques appeared as a new tool for improving the results. In this dissertation, we studied the inclusion of an unsupervised technique, *Biclustering,* that has been successfully applied in areas such as gene expression and information retrieval, but not used in the educational context.

We presented a methodology that allows us to use *Biclustering* algorithms in educational data to get new patterns and use these results as a complement to the classification. In particular, using matrices with grades of graduate Computer Science students (LEIC) of *Instituto Superior Técnico* we are able to anticipate the average grade of the master Program (MEIC) of those students.

By applying this new technique we can improve the accuracy of the classifiers, similarly to other techniques previously used, finding new types of patterns which until now had never been discovered.

Palavras Chave

Keywords

Palavras Chave

Descoberta Informação Dados Educacionais

Biclustering

Descoberta de Padrões Evolução Coerente

Previsão de Resultados dos Estudantes

Keywords

Educational Data Mining

Biclustering

Coherent Evolution Patterns

Student’s Performance

Acknowledgments

I would like to thank those who have been present in these years and have in some way contributed to this thesis.

To my supervisors, Sara Madeira and Cláudia Antunes, for the opportunity to work with them in such amazing topic and for the guidance, patience and dedication to this work.

To my friends, who have always encouraged me to work on this thesis and for all the suggestions and help they have provided me.

To my parents and brother, for all the love and whom I could always count for unlimited support and encouragement.

And last, but not least, to Ângela for her patience, devotion and unconditional love that gave me strength to accomplish my goals.

This work was partially supported by Fundação para a Ciência e Tecnologia under research project educare (PTDC/EIA-EIA/110058/2009).

Lisbon, October 2014

André Vale

Table of Contents

List of Figures

List of Tables

List of Acronyms

DM	Data Mining
EDM	Educational Data Mining
FS	Feature Selection
FIM	Frequent Item-set Mining

Chapter 1

Introduction

Nowadays, is necessary to gather data and analyze every detail about it, this type of process is known as Data Mining (DM). The overall goal of the Data Mining process is to extract information from a dataset and transform it into an understandable structure for further use. Data Mining has incorporated many techniques from other domains such as statistics, machine learning, pattern recognition, database, among others. On the last decade, there has been an increasing interest in applying data mining methods on educational data, making Educational Data Mining (EDM) a new growing research community.

The discovery of information in educational data is an emerging discipline that aims for applying the techniques of information discovery to data from educational systems, such as computer-aided learning systems or traditional classroom learning systems. In both cases, the data contain the frequent behaviours of students and teachers, the organization and coordination of educational processes, strategies educators follow, among other things. In all these cases, data are classified in a particular and well-defined context that can and should be used to better understand the processes.

1.1. Motivation

The prediction of students' performance has deserved a significant attention in Educational Data Mining (EDM) research, with several distinct approaches being proposed, mostly using classification and regression techniques. With the advances and stabilization of these techniques, it is easy to accept that the accuracy results do not depend on the technique used, but on data themselves, both on the training data and on the target variable (Romero & Ventura, 2010).

While classification tries to find a model to predict an outcome, non-supervised techniques, as pattern mining and clustering, can explore the data for identifying frequent behaviours. Previous studies (Antunes, 2008; Barracosa & Antunes, 2011) have shown that sequential pattern mining is suited to discover patterns able to model students behaviours, which in turn can be used to enrich training data, improving global classification accuracy on more than 10% (Barracosa & Antunes, 2011).

Clustering is perhaps one of the most important tools for both exploratory and confirmatory analysis. Indeed, it is a technique to discern meaningful patterns in unlabeled data by grouping together data points that are similar. Biclustering algorithms (S. Madeira & Oliveira, 2004) are a recent alternative to traditional clustering methods that allows the discovery of local patterns rather than global ones.

Besides discovering sequential patterns identified by pattern mining algorithms, biclustering is able to discover other sequential patterns that reveal coherent evolutions (Ben-Dor & Chor, 2003; Murali & Kasif, 2003).

1.2. Goals

This dissertation intends to study the use of biclustering to discover patterns in educational data. Algorithms that are currently state of the art in the area will be studied and analyzed the suitability and adaptability to the problem of analysis of educational data. We will explore the ability of biclustering to discover new patterns in educational data and make use of these patterns to enrich training data in order to improve the prediction of students' performance.

Finally at the end of the dissertation, approaches based on biclustering for analyzing educational data will be proposed which will be tested using data from the *Educare* project[1].

1.3. Approach

Our main approach consists in doing case studies, in particular two cases, to realize if the biclustering can bring new improvements to education. For this, we used algorithms that are state of the art in matrices with real data in order to find patterns that are not so evident. We will interpret the results and compare with pattern mining algorithms that have been used previously in EDM.

1.4. Contributions

This, to the best of our knowledge, is the only work that effectively applies biclustering algorithms, which are commonly used in biological data, to educational data, what allowed us to understand the advantages and disadvantages of these methods in the field of EDM. We demonstrated that the patterns that the biclustering finds are complementary to the information already found by pattern algorithms.

The results show that biclustering can slightly improve the classifiers and that we can extract important information through obtained biclusters.

With the study of this work, we were pleased to present a poster paper on the largest conference of the area of EDM, the Educational Data Mining Conference[2]. The paper describes very briefly the first results of this dissertation (Vale, Madeira, & Antunes, 2014).

[1] Project *Educare* - https://sites.google.com/site/istprojecteducare/
[2] International Conference on Educational Data Mining - http://www.educationaldatamining.org/

1.5. Dissertation Outline

The rest of this document is organized as follows: in **Chapter 2** we present the Literature Review, with a small description of data mining, a description of educational data mining and what was done in this area with special attention to clustering, a description of biclustering and an overview of the existing work, and lastly an outline of the single paper who gathered this two main areas. **Chapter 3** presents our dissertation statement. In **Chapter 4** we describe our main case study, followed by the experiments performed and a comparison with a method of pattern mining. **Chapter 5** contains our second case study, a matrix with a time variable. Finally, in **Chapter 6**, we present the conclusions of this work and point out some future work.

Chapter 2

Literature Review

In this chapter, we describe the main definitions suitable for understanding our case study and related work about two of the main subjects.

2.1. Data Mining

Data mining (DM), also commonly well-known as knowledge discovery from data (KDD), is the automated extraction of patterns representing knowledge implicitly stored or captured in databases, data warehouses, the Web, other information repositories, or data streams (Han, Kamber, & Pei, 2012).

Nowadays, a vast amount of data is collected. Data mining meets the imminent need for the effective, scalable and flexible data analysis in our society. Data mining can be considered as a natural evolution of information technology and has incorporated many techniques from other domains such as statistics, machine learning, pattern recognition, database and data warehouse systems, information retrieval, visualization, algorithms, high-performance computing, and many application domains (**Figure 1**). The interdisciplinary nature of data mining research and development contributes significantly to the success of data mining and its extensive applications.

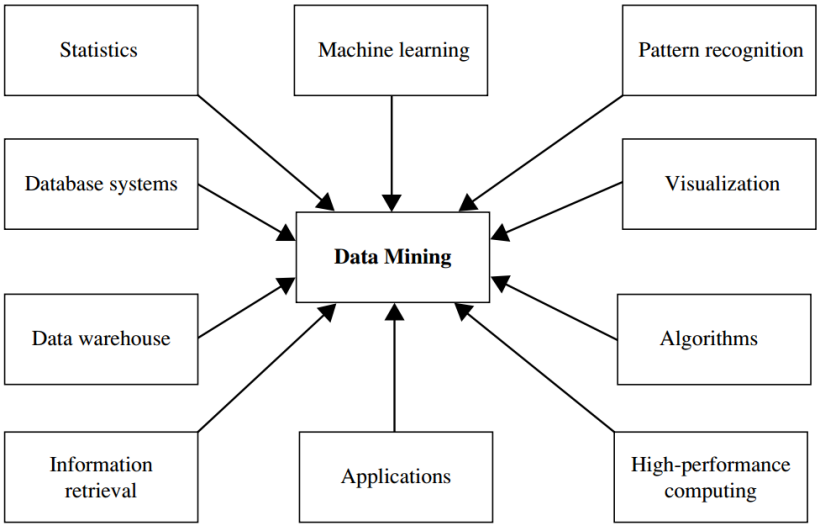

Figure 1. Data mining adopts techniques from many domains. (Adapted from (Han et al., 2012))

There are two main categories of data mining methods: supervised and unsupervised. Supervised methods use training data to predict a function, $f : X \rightarrow Y$, where X is the input space and Y is the

output space. The training data have pairs of input values and output labels or classes. Unsupervised data mining discovers classes within the data without a labeled training data. Examples of supervised mining are classification and regression. And examples of unsupervised mining are clustering, biclustering and associative rule mining.

For the scope of this work we are more interested in analysing data objects without consulting class-labelled (training) datasets, so we will focus on unsupervised methods, clustering and biclustering.

Clustering is the process of grouping a set of physical or abstract objects into classes of similar objects. We can define a *cluster* as a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. This data mining technique is a challenging field. Typical requirements of it include scalability, the ability to deal with different types of data and attributes, the discovery of clusters in arbitrary shapes, minimal requirements for domain knowledge to determine input parameters, the ability to deal with noisy data, incremental clustering and insensitivity to input order, the capability of clustering high-dimensionality data, constraint-based clustering, as well as interpretability and usability (Han et al., 2012).

However, a major drawback of clustering is the difficulty in identifying patterns that are common to only a part of the data matrix, i.e., it only clusters objects according to their attribute values. In some applications, objects and attributes are defined in a symmetric way, where data analysis involves searching data matrices for submatrices that show unique patterns as clusters. This kind of clustering technique belongs to the category of *biclustering*. Biclustering methods cluster objects and attributes simultaneously.

Data mining has many successful applications, such as business intelligence, Web search, bioinformatics, health informatics, finance, digital libraries, and engineering (Han et al., 2012). There are many challenging issues in data mining research. Areas include mining methodology, user interaction, efficiency and scalability, and dealing with diverse data types. Data mining research has strongly impacted society and will continue to do so in the future.

2.2. Educational Data Mining

Educational Data Mining (EDM) is an emerging discipline with the goal of applying data mining techniques to data that come from educational settings, i.e., use large-scale educational datasets to better understand learning and to provide information about the learning process (Romero, Ventura, Pechenizkiy, & Baker, 2010). The field of EDM has grown substantially recently, with the first workshop referred to as "Educational data mining" occurring in 2005. EDM research is composed of people from multiple disciplines, from psychology to computer science.

The application of data mining to the design of educational systems is an iterative cycle of hypothesis formation, testing, and refinement (**Figure 2**). Data mining should enter the design loop towards guiding, facilitating, and enhancing learning as a whole. In this process, the goal is not just to turn data into knowledge, but also to filter data mining for decision-making.

As we can see in **Figure 2**, educators design, plan, build, and maintain educational systems. Students use those educational systems to learn. With the available information about courses, students, and their usage and interaction, data mining techniques can be applied in order to discover useful knowledge that helps to improve educational designs. The discovered knowledge can be used not only by educational designers and teachers, but also by students. Thus, the application of data mining in educational systems can be oriented for supporting the specific needs of each of these categories of stakeholders (Romero et al., 2010).

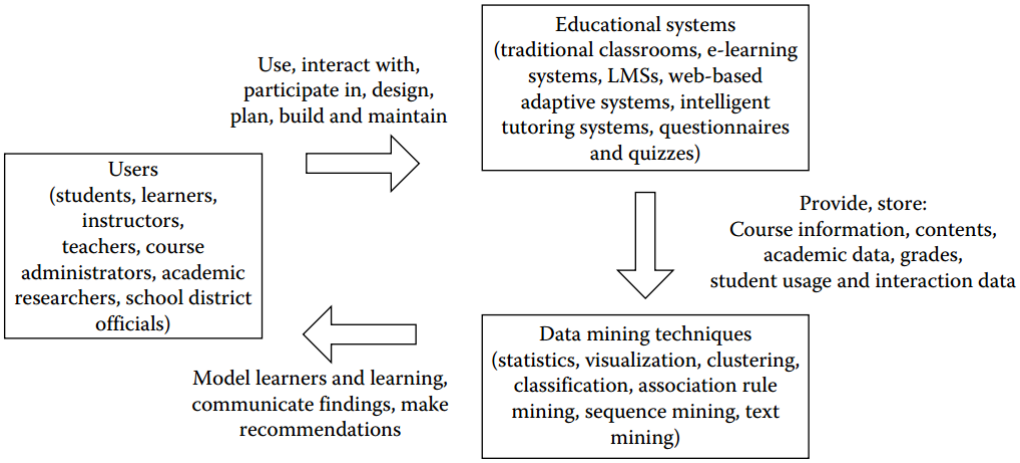

Figure 2. Data Mining on Educational Systems (Adapted from (Romero et al., 2010)).

The EDM process of converting data from educational systems into useful information does not differ much from other application areas of DM, like business, genetics, medicine, etc., because it follows the same steps as the general DM process (Sebastion Ventura, Bra, & Romero, 2004): pre-processing, DM, and post-processing. However, there are some important issues that differentiate the application of DM specifically to education from how it is applied in other domains (S. Ventura & Romero, 2007):

- **Objective**: The objective of DM in each application area is different. For example, in EDM, there are both applied research objectives, such as improving the learning process and guiding students' learning, as well as pure research objectives, such as achieving a deeper understanding of educational phenomena. These goals are sometimes difficult to quantify and require their unique set of measurement techniques.
- **Data:** In educational environments, there are many different types of data available for mining. These data are specific to the educational area, and therefore have intrinsic semantic information, relationships with other data, and multiple levels of meaningful hierarchy. Furthermore, it is also necessary to take pedagogical aspects of the learner and the system into account.
- **Techniques:** Educational data and problems have some special characteristics that require the issue of mining to be treated in a different way. Although most of the traditional DM techniques can be applied directly, others cannot and have to be adapted to the specific educational problem

at hand. Furthermore, specific DM techniques can be used for particular educational problems as is the case of this thesis.

Supervised methods are the most used in EDM. While the addition of the classifier significantly improved accuracy of these methods, they have some downsides, are time-consuming and error-prone. Humans have to supply the labels for the dataset, and it needs that they have to know relevant behaviours when there is insufficient knowledge of what these behaviours may be. Thus, supervised methods can be highly dependent on domain expertise.

On the other hand, the application of unsupervised methods on EDM has grown lately. Researchers have gotten very interesting results with this type of methods, especially when are used together with supervised methods. We will discuss these results in the next section.

Nowadays, there is a great variety of educational environments such as: the traditional classroom, e-learning, learning management system (LMS), adaptive hypermedia (AH) educational systems, intelligent tutoring systems (ITS), tests/quizzes, texts/contents, and others such as: learning object (LO) repositories, concept maps, social networks, forums, educational game environments, virtual environments, ubiquitous computing environments, etc (Romero & Ventura, 2010). All data provided by each of these educational environments are different, thus enabling different problems and tasks to be resolved using DM techniques.

EDM's contributions have influenced thinking on pedagogy and learning, and have promoted the improvement of educational software, improving software's capacity to individualize students' learning experiences.

2.2.1. Related Work on Educational Data Mining

The number of publications about EDM has grown exponentially in the past few years. Next, we are going to provide an overview of the current state of art more relevant for our work.

In 2010, Romero and Ventura surveyed the most relevant studies carried out in the field of EDM (Romero & Ventura, 2010). The paper was divided into eleven tasks/categories:

1. Analysis and Visualization of Data.
2. Providing Feedback.
3. Recommendation.
4. Predicting Performance.
5. Student Modeling.
6. Detecting Behavior.
7. Grouping Students.
8. Social Network Analysis.
9. Developing Concept Map.
10. Planning and Scheduling.
11. Constructing Courseware.

Since we are interested in exploring biclustering in educational data, we will study the most common tasks that have been resolved by using unsupervised methods, *Student Modeling* and *Grouping Students*. The objective of *Student Modeling* is to develop cognitive models of students, including a modeling of their skills and declarative knowledge. Through *Grouping Students* we can create groups of students according to their customized features, personal characteristics, etc.

Some ensemble methods, unsupervised plus supervised methods, were used in student modeling to obtain good results: Clustering and classification machine learning have been proposed to reduce development costs in building user models and to facilitate transferability in intelligent learning environments (Amershi, 2009); Clustering and classification of learning variables have been used to measure the online learner's motivation (Hershkovitz & Nachmias, 2009); Pairwise Clustering and a supervised online classification were applied to learning trajectories of students suffering from learning disabilities and to determine subgroups who share similar learning patterns. The usefulness of clustering for the analysis of learning patterns and further training individualization contribute to a better support for children with learning difficulties (Kaser et al., 2013).

Clustering techniques also have been used to obtain and characterize groups of students with different profiles, discriminating features and external profiling features (pass/fail) to support teachers in collaborative student modeling (Hernandez-del-Olmo, Montero, Gaudioso, & Talavera, 2009). The clustering algorithm *k-means* has been used to model student's behaviour with a very small set of parameters without compromising the behaviour of the system. They managed to drastically reduce the parameter space used to model students (Ritter et al., 2009). And recently, Li et al. proposed an automated approach that finds student models using a clustering algorithm based on automatically-generated problem content features. They demonstrated that it can produce models of good prediction accuracies, and showed how the discovered model could provide important instructional implications (N. Li, Cohen, & Koedinger, 2013).

Different clustering algorithms have been used to group students. Some are ensemble methods: a hybrid method of clustering and Bayesian networks was implemented to group students according to their skills. The results indicated that it could help a distant education teacher improve exercises, schedule a course, and identify potential dropouts at an early phase (Hämäläinen, 2004). Trivedi et al. introduced a method by which they can ensemble together multiple models based upon clustering students. They showed that the assessment quality using Intelligent Tutoring Systems can save much instructional time that is currently used for just assessment (Trivedi, Pardos, & Heffernan, 2011).

Variations of well-known clustering algorithm, *k-means,* were implemented to differentiate the student's skills: A version of k-means clustering algorithm for effectively grouping students who demonstrate similar learning portfolios (students' assignment scores, exam scores, and online learning records) (Chen, Chen, & Li, 2007); A version of k-means clustering algorithm to discover interesting patterns that characterize the work of stronger and weaker students (Perera, Kay, Koprinska, Yacef, & Zaiane, 2009); Nugent et al. presented a flexible version of k-means that allows empty clusters

which can generate groups of students with similar skills, useful when the number of clusters is not known (Nugent, Dean, & Ayers, 2010).

Myller et al. used other well-known clustering algorithm, Expectation–Maximization algorithm, to form heterogeneous groups according to student's skills. This paper was one of the first to give steps on the way towards an environment targeted for assisting students in project assignments (Myller, Suhonen, & Sutinen, 2002).

Hierarchical clustering was used to group students with different characteristic features: A hierarchical clustering algorithm for user modeling (learning styles) in intelligent e-learning systems in order to group students according to their individual learning style preferences (Zakrzewska, 2008); Hierarchical agglomerative clustering, k-means and model-based clustering were applied to identify groups of students with similar skill profiles (Ayers, Nugent, & Dean, 2009); Hierarchical cluster analysis to establish the proportion of students who get an exercise wrong or right (Barker-Plummer, Cox, & Dale, 2009); A strategy based on time series and agglomerative hierarchical clustering was proposed which aim at determining what different behavior patterns are adopted by students in online discussion forums (Cobo & García-Solórzano, 2011).

Other types of clustering algorithms not so well known were introduced in educational data. They all try to find group of students by skills/characteristics:

- A clustering algorithm based on large generalized sequences to find groups of students with similar learning characteristics based on their traversal path patterns and the content of each page they have visited (Tang & McCalla, 2002).
- Model-based clustering to automatically discover useful groups from learning management system data to obtain profiles of student's behavior (Talavera & Gaudioso, 2004).
- An improvement in the matrix-based clustering method for grouping learners by characteristics in e-learning (Zhang, Cui, Wang, & Sui, 2007).
- A fuzzy clustering algorithm to find interested groups of learners according to their personality and learning strategy data collected from an online course (Tian, Wang, Zheng, & Zheng, 2008).
- A conditional subspace clustering algorithm to identify skills that differentiate students (Nugent, Ayers, & Dean, 2009).
- A two-step cluster analysis to classify how students organize personal information spaces (piling, one-folder, small-folders, and big-folder filing) (Hardof-jaffe, Hershkovitz, Abu-kishk, Bergman, & Nachmias, 2009).
- Trivedi et al. used Spectral Clustering, a graph theoretic technique for metric modification, to improve the student performance prediction. It gives a much more global notion of similarity between data points as compared to other clustering methods such as k-means (Trivedi, Pardos, Sárközy, & Heffernan, 2011).

New algorithms based on clustering have been implemented to obtain new results on education data: A genetic clustering algorithm to solve the problem of allocating new students (which places new students into classes so that the gaps between learning levels in each class is minimum and the

number of students in each class does not exceed the limit) (Zukhri & Omar, 2007); A simple unsupervised clustering algorithm for hidden Markov models has been developed, *Stepwise-HMM-Cluster*, can discover student learning tactics while incorporating student-level outcome data, constraining the results to interpretable models that also predict student learning (Shih, Koedinger, & Scheines, 2010); And a new unsupervised framework, *query-likelihood clustering*, for classifying student dialogue acts was developed. One limitation of the proposed approach, like the clustering algorithms in general, is that a significant amount of human intelligence is often required to decide on the number of suitable clusters (Ezen-Can & Boyer, 2013).

All the literature reviewed uses clustering algorithms that allowed to find different groups of students with similar skills/characteristics, improving the educational system or contributing to a better support of supervised methods. However, clustering generally fails in identifying patterns that are common to only a subgroup of skill/characteristics. With the biclustering technique, we can find new types of patterns that clustering technique cannot find.

2.3. Biclustering

Biclustering refers to a distinct class of clustering algorithms that perform simultaneous row and column clustering of a matrix. It is used mainly in gene expression data analysis, but have also been proposed and used in other application fields under the names of co-clustering (Dhillon, 2001), subspace clustering (Rakesh Agrawal, Gehrke, Gunopulos, & Raghavan, 1998), direct clustering (Hartigan, 1972) and block clustering (Mirkin, 1996). Although the majority of the recent applications of biclustering are in biological data analysis, there are several interesting application of biclustering in other domains, such as: information retrieval and text mining, collaborative filtering, recommendation systems, target marketing and market research, database research, and data mining (S. Madeira & Oliveira, 2004). In this work, we study its application to educational data research.

Biclustering can be applied whenever the data to analyze has the form of a real-valued or symbolic matrix A, where the value a_{ij} represents the relation between row i and column j, and the goal is to identify subsets of rows with certain coherence properties in a subset of the columns.

The goal of biclustering algorithms is to identify a set of biclusters. Let A be a matrix defined by its set of rows, R, and its set of columns, C. Then we can define bicluster as follows (S. C. Madeira, Teixeira, Sá-Correia, & Oliveira, 2010):

Definition 1 (Bicluster). A bicluster $B = (I, J)$ is a submatrix A_{IJ} defined by $I \subseteq R$, a subset of rows, and $J \subseteq C$, a subset of columns. A bicluster with only one row or one column is called *trivial*.

This set of biclusters $B_k = (I_k, J_k)$ satisfies specific characteristics of homogeneity. We use the classification proposed by Madeira and Oliveira (S. Madeira & Oliveira, 2004), according to which there are four types of biclusters that the algorithms aim to find:

1. Biclusters with constant values.

2. Biclusters with constant values on rows or columns.
3. Biclusters with coherent values.
4. Biclusters with coherent evolutions.

The first three classes analyze the numeric values in the data matrix directly and try to find subsets of rows and subsets of columns with similar behaviours. The fourth class aims to find coherent behaviours regardless of the exact numeric values in the data matrix. Biclusters with constant values are just biclusters with similar values. Biclustering algorithms that look for biclusters with coherent values, *(I, J)*, can be viewed as based on an additive (1) or multiplicative (2) model,

$$a_{ij} = \mu + \alpha_i + \beta_j \quad (1)$$

$$a_{ij} = \mu \times \alpha_i \times \beta_j \quad (2)$$

where μ is the typical value within the bicluster, α_i is the adjustment for row $i \in I$, and β_j is the adjustment for column $j \in J$.

The last type address the problem of finding coherent evolutions across the rows and/or columns of the data matrix regardless of their exact values by viewing the elements in the data matrix as symbols, which can be purely nominal, correspond to a given order or represent coherent positive and negative changes relative to a normal value. The biclusters presented in **Figure 3** are examples of biclusters with coherent evolutions, which are the type of biclusters that we believe that can bring more interesting patterns to our case study.

In the case of gene expression data, identifying a bicluster with coherent evolutions may be helpful if one is interested in finding a subset of genes that are *upregulated* or *downregulated* across a subset of conditions without taking into account their actual expression values; or if one is interested in identifying a subset of conditions that have always the same or opposite effects on a subset of genes. In the case of Educational Data Mining, we can have, for example, a matrix relating students and subjects using grades where we might be interested in finding a group of students that always have the same grades for the same subset of subjects.

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

(a)

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

(b)

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

(c)

70	13	19	10
49	40	49	35
40	20	27	15
90	15	20	12

(d)

Figure 3. Examples of different types of biclusters with coherent evolutions. (a) Overall coherent evolution, (b) coherent evolution on the rows, (c) coherent evolution on the columns, and (d) coherent evolution on the columns. (Adapted from (S. Madeira & Oliveira, 2004))

Most biclustering algorithms assume the existence of several biclusters, others only aim at finding one bicluster. Madeira and Oliveira (S. Madeira & Oliveira, 2004) discern nine different structures of biclusters:

1. Only one bicluster.

2. Exclusive row and column biclusters (rectangular diagonal blocks after row and column reorder).
3. Nonoverlapping biclusters with checkerboard structure.
4. Exclusive-rows biclusters.
5. Exclusive-columns biclusters.
6. Nonoverlapping biclusters with tree structure.
7. Nonoverlapping nonexclusive biclusters.
8. Overlapping biclusters with hierarchical structure.
9. Arbitrarily positioned overlapping biclusters.

We are interested in arbitrarily positioned overlapping biclusters which is the most general structure that allows the existence of many possibly overlapping biclusters without taking into account their direct observation on the data matrix with a common reordering of its rows and columns. Moreover, is a structure that several biclustering algorithms can obtain.

In **Table 1** we illustrate a $n \times m$ grade matrix where *n* students (rows) are evaluated in *m* subjects (columns) and each element a_{ij} is an integer value between 10 and 20 corresponding to the grade of student *i* in subject *j*. This is an example of the type of matrix where we aim at finding biclusters.

Table 1. Grade Matrix.

	Subject 1	…	Subject *j*	…	Subject *m*
Student 1	a_{11}	…	a_{1j}	…	a_{1m}
Student …	…	…	…	…	…
Student *i*	a_{i1}	…	a_{ij}	…	a_{im}
Student …	…	…	…	…	…
Student *n*	a_{n1}	…	a_{nj}	…	a_{nm}

2.3.1. Related Work on Biclustering

The earliest algorithm that introduced the technique of biclustering was direct clustering (Hartigan, 1972). Later, Cheng and Church introduced biclustering in gene expression data analysis (Cheng & Church, 2000). In the past years several studies have been made on comparison and evaluation of biclustering methods (S. Madeira & Oliveira, 2004), (Bozdağ, Kumar, & Catalyurek, 2010), (Verma, Singh, & Cui, 2010), (Eren, Deveci, Küçüktunç, & Çatalyürek, 2013). Algorithms like *Cheng and Church*, *Plaid Model* (Lazzeroni & Owen, 2002), *SAMBA* (Tanay, Sharan, & Shamir, 2002), *OPSM* (Ben-Dor & Chor, 2003), *ISA* (Bergmann, Ihmels, & Barkai, 2003) and *xMOTIFs* (Murali & Kasif, 2003) are highly cited in the literature. Other algorithms have appeared recently such as *BiMax* (Prelić et al., 2006), *Bayesian biclustering* (Gu & Liu, 2008), *COALESCE* (Huttenhower, Mutungu, Indik, & Yang, 2009) and *FABIA* (Hochreiter et al., 2010).

In what follows we first describe the most well-known algorithms and then we review the algorithms with more relevance to our case study.

Cheng and Church is a deterministic greedy algorithm that seeks to find the biclusters with low variance, as defined by the mean squared residue (MSR) (Cheng & Church, 2000). The *Plaid Model* is a statistical modelling approach which represents the input matrix as a superposition of layers where each layer corresponds to a bicluster. *Iterative Signature Algorithm* (*ISA*) is a nondeterministic greedy algorithm that seeks biclusters with two symmetric requirements, each column in the bicluster must have an average value above some threshold, T_C, and likewise each row must have an average value above some threshold, T_R.

BiMax is a divide and conquer algorithm that seek the rectangles of 1's in a binary matrix. *Bayesian biclustering* uses Gibbs sampling to fit a hierarchical Bayesian version of the plaid model. *QUBIC* is a deterministic algorithm that reduces the biclustering problem to finding heavy subgraphs in a bipartite graph representation of the data (G. Li, Ma, Tang, Paterson, & Xu, 2009). *Combinatorial algorithm for expression and sequence-based cluster extraction* (*COALESCE*) is a nondeterministic greedy algorithm that seeks biclusters representing regulatory modules in genetics. *FABIA* is based on a multiplicative model, which accounts for linear dependencies between gene expression and conditions, and also captures heavy-tailed distributions.

Follows a review of algorithms that seek biclusters with coherent evolutions. *OPSM* is a deterministic greedy algorithm that seeks biclusters with ordered rows (Ben-Dor & Chor, 2003), we will detail this algorithm later. *OP-Cluster* is similar with *OPSM*, but it can capture the consistent tendency exhibited by a subset of objects in a subset of dimensions in a high dimensional space (Liu & Wang, 2003). *xMOTIFs* is a nondeterministic greedy algorithm that seeks biclusters with conserved (coherent) rows in discretized dataset. *SAMBA* (Statistical-Algorithmic Method for Bicluster Analysis) is a biclustering algorithm that performs simultaneous bicluster identification by using exhaustive enumeration (Tanay et al., 2002).

In **Table 2** we have an example of a matrix with ten students and seven subjects where we have some types of biclusters that algorithms like *OPSM*, *SAMBA* and *xMOTIFs* can find. More specifically, we have four arbitrarily positioned overlapping biclusters where $\boldsymbol{B_1}$ is a bicluster with constant values that the algorithm *SAMBA* can get and $\boldsymbol{B_2}$, $\boldsymbol{B_3}$, $\boldsymbol{B_4}$ are biclusters with coherent evolutions that the algorithm *OPSM* can find. Also, if we do the transpose of the matrix, we can find the $\boldsymbol{B_3}$ bicluster using the *xMOTIFs* algorithm.

Table 2. Example of matrix with grades of ten students at seven subjects.

	Subject 1	*Subject 2*	*Subject 3*	*Subject 4*	*Subject 5*	*Subject 6*	*Subject 7*
Student 1	18 B_4	12	18	10	12	18	12
Student 2	17	16	17	13	15	17	16
Student 3	19	16	19	17	16	19	15
Student 4	17	15	16	B_1 19	19	16	17
Student 5	B_2 18	16	14	19	19	19	18
Student 6	17	15	13	19	14	17	16
Student 7	19	18	16	20	B_3 18	17	19
Student 8	14	16	13	14	18	17	19
Student 9	16	18	15	16	18	17	19
Student 10	13	13	10	11	13	12	10

Below we detail the main and most successful algorithms that seek biclusters with coherent evolutions: *OPSM*, *SAMBA* and *xMotifs*.

OPSM Algorithm. Order-preserving submatrix (*OPSM*) algorithm was first proposed by Ben-Dor et al. (Ben-Dor & Chor, 2003). *OPSM* as the name suggests extracts biclusters with columns organized in a monotonically increasing order.

Given a data matrix *D* with respectively *n* rows and *m* columns, they defined a complete model as the pair (J, π), where *J* is a set of *s* columns and $\pi = (j_1, j_2, \ldots, j_s)$ is a linear ordering of the columns in *J*. A row supports (J, π) if the *s* corresponding values, ordered according to the permutation π, are monotonically increasing.

To find the best model given *s*, an exhaustive algorithm could try all complete models but this approach is not feasible for $s \geq 4$ and to large values of *n* and *m*. So, the idea is to grow partial models iteratively until they become complete models. A partial model of order (a, b) specifies, in order, the indices of the *a* "smallest" elements $<j_1, \ldots, j_a>$ and the indices of the *b* "largest" elements $<j_{s-b+1}, \ldots, j_s>$ of a complete model (J, π) and its size *s*. The algorithm starts by evaluating all (1, 1) partial models and keeping the best *l* of them, i.e. the partial models with highest statistically significant support. It then expands them to (2, 1) models and keeps the best *l* of them. After that, it expands them to (2, 2) models, (3, 2) models, and so on, until it gets *l* $([s/2], [s/2])$ models, which are complete models. It then outputs only the best partial bicluster. The bicluster presented in **Figure 3d** is an example of the type of bicluster that this algorithm produces.

SAMBA Algorithm. Statistical-Algorithmic Method for Bicluster Analysis (*SAMBA*) was proposed by Tanay et al. (Tanay et al., 2002). They defined a bicluster as a subset of genes (rows) that jointly respond across a subset of conditions (columns). A gene is considered to respond to a certain condition if its expression level changes significantly at that condition with respect to its normal level. Before *SAMBA* is applied, the expression data matrix is modeled as a bipartite graph whose two parts correspond to conditions (columns) and genes (rows), respectively, with one edge for each significant expression change. Tanay et al. present two statistical models for the resulting graph. In the simpler model, they are looking for biclusters that manifest changes relatively to their normal

level, without considering if the change was an increase or a decrease in the expression level. In the refined model, they look for consistent biclusters, in which every two conditions must always have the same effect or always have the opposite effect on each of the genes.

In the simpler model, it is assumed that all the genes in a given bicluster are regulated (up or down). This means that their values changed relatively to its normal level, in the subset of conditions that form the bicluster. The goal is then to find the largest biclusters with the regulation property. In order to do that, *SAMBA* does not try to find any kind of coherence on the values a_{ij}. It assumes that regardless of its true values, a_{ij} can be represented by two symbols: *S0* or *S1*, where *S1* means change and *S0* means no-change. As such, the model graph has an edge between a gene and a column when there is a change in the expression level of that gene in that specific condition. No edge means no change. A large bicluster is, in this case, one with a maximum number of genes (rows) whose symbol standing for a_{ij} is expected to be *S1*. The bicluster presented in **Figure 3a** is an example of the type of bicluster that this simple model produces, if we say that *S1* is the symbol that represents a coherent change relative to normal expression.

In the refined model, the sign of the change is taken into account. This is achieved by assigning a signal $c_{ij} \in \{-1, 1\}$ to each edge of the graph, and then looking for a bicluster *(I, J)* and an assignment $\tau : I \cup J \rightarrow \{-1, 1\}$ such that $c_{ij} = \tau(i)\tau(j)$. This is equivalent to the selection of a set of columns (conditions) that have always the same or opposite effects on the set of rows.

However, the approach of Tanay et al. is not purely symbolic since the merit function used to evaluate the quality of a computed bicluster using *SAMBA* is the weight of the subgraph that models it. Its statistical significance is evaluated by computing the probability of finding at random a bicluster with at least its weight. Given that the weight of a subgraph is defined as the sum of the weights of gene-condition (row-column) pairs in it including edges and non-edges, weights are assigned to the edges of the bipartite subgraph so that heavy subgraphs correspond to statistical significant biclusters, the biclusters that the algorithm returns.

xMOTIFs Algorithm. Murali and Kasif (Murali & Kasif, 2003) introduced an algorithm that aims at finding *xMOTIFs*. They defined an *xMOTIF* as a subset of genes (rows) that is simultaneously conserved across a subset of the conditions (columns), i.e., a bicluster with coherent evolutions on its rows. The expression level of a gene is conserved across a subset of conditions if the gene is in the same state in each of the conditions in this subset. They consider that a gene state is a range of expression values and assume that there are a fixed given number of states. These states can simply be *upregulation* and *downregulation*, when only two states are considered. To determine an *xMOTIF*, it is necessary to compute the set of conserved rows, *I*, the states that these rows are in, and the set of columns, *J*, that match the *xMOTIF*. Given the set of conserved rows, *I*, the states of the conserved rows, and one column *c* that matches a given motif, it is easy to compute the remaining conditions in *J* simply by checking, for each column *c'*, if the rows in *I* are in the same state in *c* and *c'*. Column *c* is called a "seed" from which the entire motif can be computed. The motifs are computed starting with a set of randomly chosen columns that act as seeds. For each column, an additional randomly

chosen set *D* of columns is selected, called a *discriminating set*. The selected bicluster contains all the rows that have states equal in the seed column and in the columns contained in the discriminating set *D*. The motif is discarded if less than an *α-fraction* of the columns match it. After all the seeds have been used to produce *xMOTIFs*, the largest *xMOTIF* (one with the largest number of rows) is returned. Therefore, *xMOTIFs* can find biclusters with constant values at rows, an example of a bicluster is the one presented in **Figure 3b**.

2.3.2. Biclustering Tools

Some tools have been made to evaluate and compare algorithms. One that implements some algorithms that are relevant for our case study is *BicAT* (Barkow, Bleuler, Prelic, Zimmermann, & Zitzler, 2006). Biclustering Analysis Toolbox (*BicAT*) is a graphical platform used for data analysis utilizing various clustering and biclustering methods. The toolbox provides the facility for data normalization, discretization, filtering the bicluster across a specific condition or gene pair analysis for bipartite graphs.

Another existing tool is the *Expander* (Sharan, Maron-Katz, & Shamir, 2003). *Expander* is a gene expression analysis and visualization software. It implements clustering and biclustering algorithms and provides data pre-processing and normalization, among other things.

2.4. Related Work on EDM and Biclustering

The literature about biclustering on educational data is almost non-existent. Only recently Trivedi et al. proposed this technique on education area (Trivedi, Pardos, Sarkozy, & Heffernan, 2012). In their work they used the idea of co-clustering (namely biclustering) students and their tutor interaction features and interleave it with a bagging strategy which was used previously with clustering (Trivedi, Pardos, & Heffernan, 2011; Trivedi, Pardos, Sárközy, et al., 2011).

They considered the approach proposed by Dhillon (Dhillon, 2001) which formulates the problem of co-clustering as a bipartite graph partitioning problem. An undirected bipartite graph is a triple represented by $G = (S, F, E)$ where S and F are two sets of vertices and E is the set of edges. Since it is a bipartite graph, one end of the edges in set E have an endpoint in S and another in F. In their case the set S is the set of students while the set F is the set of features or items.

Then they used the spectral approach demonstrated by Dhillon combining the *Laplacian matrix*, eigenvectors, eigenvalues and the classical *k-means* clustering algorithm, to give a simultaneous clustering of the rows and the columns to find the optimal biclusters of students and features.

Lastly, the authors proposed an ensemble learning algorithm that uses a *bagging technique*. The basic idea behind ensemble methods is that they involve running a “base learning algorithm” multiple times, each time with some change in the representation of the input (e.g. only considering a subset of features in each run) so that a number of diverse predictions (or maps) could be obtained. This

diversity in prediction is then exploited to get better predictions. Thus ensemble methods try to learn multiple functional maps and learn a more distributed and hence richer representation of the input space at the same time.

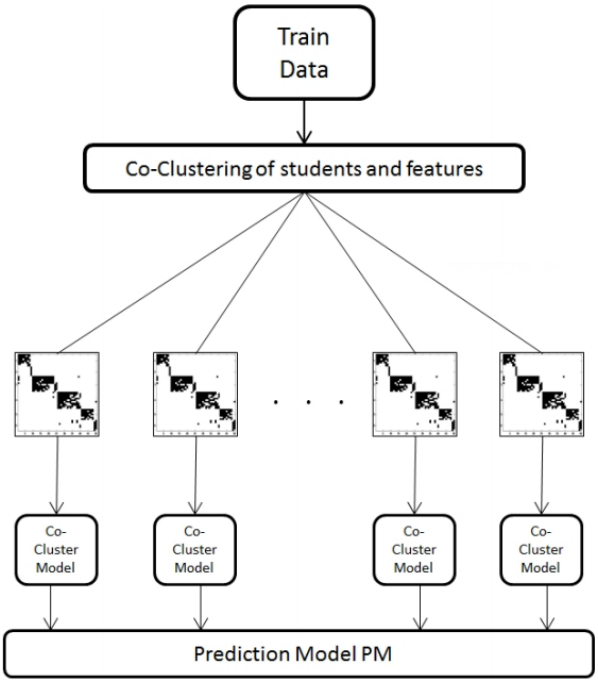

Figure 4. Finding a Prediction Model, *PM*. Adapted from (Trivedi et al., 2012).

The bagging methodology was defined as follows. For each co-cluster they trained a separate linear regression model only using the data instances and features assigned to it. So as a result they obtained *k* co-cluster models. These predictions were different and so this diversity in prediction could be used by averaging all the (or half) the predictions obtained to get a single much stronger prediction. The combination of the *k* co-cluster models would be considered to be a Prediction Model which makes a single prediction on the test set (see **Figure 4**).

The above strategy, using co-clustering if averaged properly could definitely have more predictive power as it generates diversity by considering a different subset of data instances and features each time, consequently also generating a much larger set of predictions.

In their study they used two datasets. The datasets come from the 2004-05 and 2005-06 school years, the first two full years when *ASSISTments*, an e-learning tutoring system, was used in schools in Massachusetts. These datasets had a total number of six features (columns). The data in the 2004-05 set is for 628 students (rows), while the 2005-06 data is for 761 students (rows).

The results that they obtained were better than the baseline and also indicated that the dynamic assessment condition returns a much better prediction of student test scores as compared to the static condition.

However, this technique has some limitations. The datasets that were used were not vast and did not have a large number of columns (only six). Moreover, they used the *k-means* algorithm twice, on columns and rows, not getting such good results as would obtain using biclustering algorithms.

Thus, to the best of our knowledge, this dissertation is the first contribution of biclustering in EDM.

2.5. Open Issues on Educational Data Mining

Despite major problems in EDM have been resolved in the past years, there are still some open issues.

It remains difficult to dealing with the amount of data that is collected and the small amount that is collected. In a classroom there are, at most, few hundreds of students and they probably don't take the same exercises. Collecting years of data can be an option, although course offerings change from year to year in their curriculum. Far from unanimity is also to know which data to log and how to do it.

A more fundamental problem of EDM is the fact that there are no specific and standardized methods for this research area, being urgent the implementation of these methods for the evolution of this area.

Summary

In this chapter we get to know the key concepts needed to understand the rest of the document, more specifically the definition of EDM and Biclustering. We introduced the biclustering algorithms and the existing EDM work related to the topic of this dissertation. And we also realize that this work is the first biclustering approach in EDM.

Chapter 3

Dissertation Statement

In this chapter, we offer a concise summary of the main purpose of this dissertation.

This dissertation is intended to demonstrate the added value that the use of biclustering brings to the discovery of information in educational data.

Biclustering has already been used with very success in other areas, with this study we purpose to bring the application of biclustering for the first time in education. We intend to realize if the biclustering in EDM can bring new results that have not been previously accomplished. Biclustering also allows us to discover information that other techniques cannot easily find in EDM.

In order to reach some conclusions from the application of biclustering in EDM, we decided to study two cases in particular:

- Case study 1: We applied four biclustering algorithms that allow to find different types of patterns. We used these biclusters to improve classifiers and compared the results with another technique that finds some similar patterns that had already been used in EDM.

- Case Study 2: We applied biclustering to a matrix that has the time variable, which allowed us to realize how biclustering behaves when we have time educational data.

With these case studies we intend to introduce and study the suitability and adaptability of biclustering algorithms to the problem of analysis of educational data.

Chapter 4

Case Study - Students' Grades

In this chapter, we will obtain biclusters from a students' grade matrix, use these results as feature in classification and compare with an area of pattern mining.

This Chapter presents the pre-processing task that was applied to Educare data (Section **4.1**), the bi-clustering analysis (Section **4.2**) that show how are the biclusters discovered by the algorithms, the experiments using this biclusters on the dataset (Section **4.3**) and a comparison with Frequent Item-Set Mining (Section **4.4**).

The dataset used as input was collected in the Educare project[3], a project that studies the Computer Science students' of Instituto Superior Técnico (IST) from Universidade de Lisboa between 1997 and 2012.

4.1. Data Analysis

To reach the final dataset, we had to pre-process the data. In order to simplify the pre-processing, we decided to use only the students who entered after Bologna Process because we do not need to do an equivalence between the subjects before and after the Bologna Process. Hence, data has two main groups of students: from graduation (LEIC) and master (MEIC) programs. LEIC has a duration of three years (6 semesters) with 30 subjects and MEIC is a specialization of LEIC that has a duration of two years (4 semesters) with 15 subjects.

We obtained a Bologna dataset with 3764 instances. First, we removed those who has less than 20 approved subjects (out of 30). With this we removed students who dropped the course and those who have not made many subjects. We managed to reduce the percentage of missing values (which are the subjects that have not been made by students) from 68.32% to 8.64% obtaining a dataset with 648 out of 3764 instances. Then we removed all students who have not yet started MEIC, i.e., those who still have no subjects approved in MEIC, so we could calculate an average for each student in MEIC. Later we will this MEIC average as a class in the classifiers. In the end we obtained a matrix with 442 students and 30 subjects with only 4,69% of missing values.

[3] Project *Educare* - https://sites.google.com/site/istprojecteducare/

Thus, we built our matrix with data from LEIC (students x subjects matrix) where we have students in the rows and subjects in the columns, and the values in the matrix are the students' grades (see **Table 1**). These values are numbers between 10 and 20.

In **Figure 5,** we present the distribution of the MEIC grade average of students. We can see a symmetric histogram, moving closer to a normal distribution.

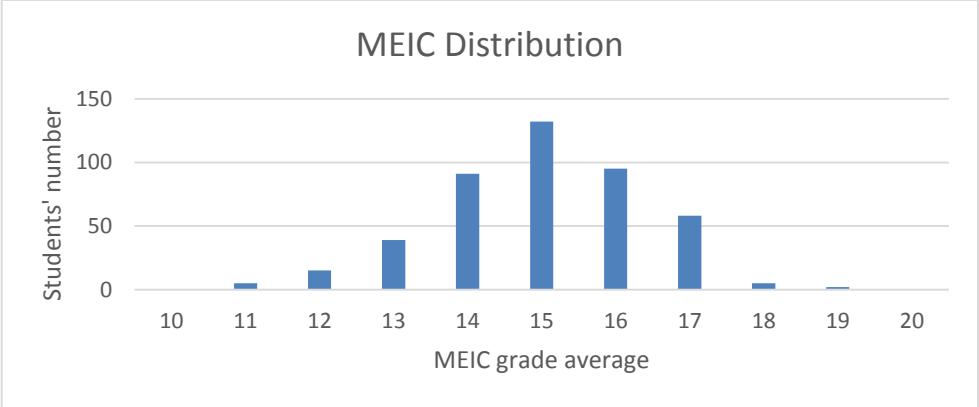

Figure 5. Distribution of the average grade of the MEIC students.

In the case of student's grades, a training dataset can be created using the *n* subjects from LEIC as attributes for the *m* instances of students, with the class being the average grades at MEIC, discretized into 3 bins (Fair, Good and Very Good). The correspondence between the bins and MEIC's averages is in **Table 3**. In **Figure 6** we can see how classes were distributed. As normal, the *Very Good* class is in a smaller number than the other classes.

In this manner is possible to train a model to anticipate the average grades of new students enrolling in MEIC.

Table 3. Correspondence between classes and grades.

Classes	Grades
Fair	[10, 14]
Good	[15, 16]
Very Good	[17, 20]

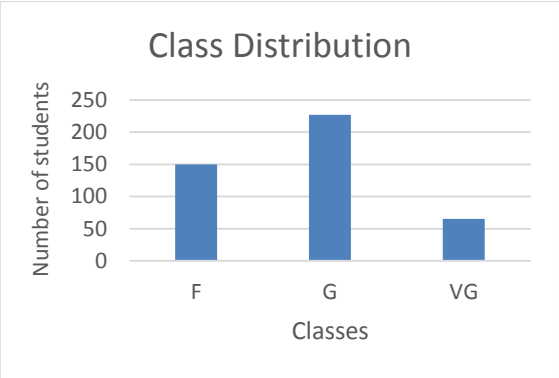

Figure 6. Class distribution.

To obtain more reliable results with biclustering analysis, it was decided to discretize the matrix in order to get a small number of symbols. Instead of having grades between 10 and 20 (corresponding to 11 symbols), we produce a dataset with five symbols, A to E. **Table 4** has the association that was made between symbols and grades.

Table 4. Correspondence between symbols and grades.

Symbols	Grades
A	{20, 19, 18}
B	{17, 16}
C	{15, 14}
D	{13, 12}
E	{11, 10}

Thus, in the end we got a matrix with a configuration equal to the **Table 5** where we have values of $x \epsilon$ {A, B, C, D, E} and $y \epsilon$ {F, G, VG}.

Table 5. Grade matrix having in the last column the corresponding Class for each student.

	Subject 1	…	Subject *j*	…	Subject *m*	Class
Student 1	x_{11}	…	x_{1j}	…	x_{1m}	$y_{1(m+1)}$
Student …	…	…	…	…	…	…
Student *i*	x_{i1}	…	x_{ij}	…	x_{im}	$y_{i(m+1)}$
Student …	…	…	…	…	…	…
Student *n*	x_{n1}	…	x_{nj}	…	x_{nm}	$y_{n(m+1)}$

4.2. Biclustering Analysis

After pre-processing, we applied the four biclustering algorithms mentioned before (Section **2.3.1**) to the matrix and obtained 16 biclusters with *OPSM,* 975 with *xMotifs,* 308 with *ISA* and 39 with *Bimax* (**Table 6**). With the biclusters obtained, our goal is to enrich a training dataset in order to improve the accuracy of the classification that predicts the average grades of students of MEIC.

Table 6. Statistics of biclusters.

Algorithm	Number of biclusters	Average Size	Average number of lines	Average number of columns
OPSM	16	403	97	10
xMotifs	975	27	5	7
ISA	308	50	23	2
Bimax	39	328	82	4

The biclusters were found using the *BicAT* tool (Barkow et al., 2006). After several tests with different values of the parameters of the algorithms, we have reached these best final parameters:

- **OPSM:** l → 999;
- **xMotifs:** Default parameters;
- **ISA: t_g** → 1.8, **t_c** → 1.8, **starting points** → 1000;
- **Bimax: minimum number of genes:** 175 (weak), 41 (medium), 15 (excellent), **minimum number of chips:** 4.

In the **Annex** section are some examples of biclusters that each algorithm found. The green cells correspond to lower grades, whereas cells in red correspond to higher grades. As we will see in the next section, these biclusters will be used as attributes in the dataset to improve classification.

On **Table 7** we show examples of five patterns that have a higher support in each one of the four bicluster algorithms. We can observe that each algorithm has a most predominant number of subjects, for example, *OPSM* find patterns where frequently appears ACED, EO, PO and IPM. On the other hand, in *Bimax* mainly appears CDI-I, CDI-II and PEst subjects with the highest incidence.

Table 7. Patterns with most support found by each bicluster algorithm.

Algorithm	Pattern	Support (%)
OPSM	(Ges, IPM)	97,1
	(ACED, SSina, IPM)	81,5
	(ACED, PEst, LP, IPM)	59,0
	(ACED, EO, PO, CGra, IPM)	39,8
	(ACED, EO, PO, SSina, CGra, IPM)	26,0
xMotifs	(SD, AL, Ges, SSina, Com)	4,1
	(SD, CDI-I, CDI-II, MD, Ges)	3,6
	(SD, AL, TCom, MO, Ges)	3,6
	(ACom, IAED, SO, CGra, LP)	3,6
	(AL, TCom, MO, Com, PEst)	3,4
ISA	(CDI-I, IPM)	39,4
	(ACED, CGra)	27,8
	(CDI-II, CGra)	26,9
	(Ges, CGra, ESof)	20,8
	(AL, IPM, SDis)	20,1
Bimax	(CDI-I, CDI-II, ACED, PEst)	43,7
	(CDI-I, CDI-II, MO, PEst)	43,0
	(CDI-II, Ges, ACED, PEst)	42,3
	(CDI-I, CDI-II, MO, ACED)	41,9
	(CDI-II, MO, ACED, PEst)	41,6

4.3. Evaluation

As in (Barracosa & Antunes, 2011), a new dataset can be obtained from the previous one, enlarged *k* Boolean attributes, one for each bicluster. Each bicluster attribute is then filled with the true value whenever the bicluster has the student instance and false otherwise. As such, the new instances take the following format:

Subject1, Subject2, … , SubjectN, Bicluster1, Bicluster2 , … , BiclusterK, Class

4.3.1. Methodology

To help us on feature selection, cross-validation and classification, we used the data mining open source software *Weka 3.6.10*[4] (Hall et al., 2009). The data flow used in *Weka* consists of first loading the data, using a feature selection method to get only the best columns, making cross-validation with 10 folds since we did not have defined training and test sets, and finally applying a classifier for obtaining the results (**Figure 7**).

[4] *Weka 3: Data Mining Software in Java* - http://www.cs.waikato.ac.nz/ml/weka/

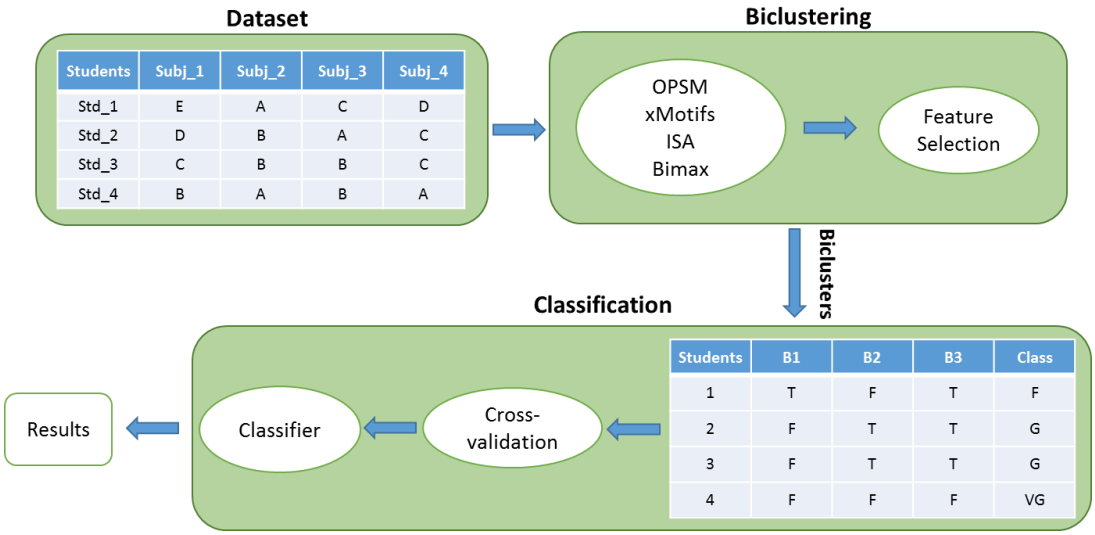

Figure 7. Data flow used.

4.3.2. Feature Selection

Feature Selection (FS), sometimes referred as data selection, discover relevant features in a dataset. This step is of huge importance to simplify the data, and minimize the confusion presented to the classifier. We use a FS method that allowed us to choose the best attributes of our dataset, including the best biclusters. In order to understand what is the best FS method for our problem, all these existing methods of feature selection in *Weka* were tested (Hall et al., 2009):

- ChiSquaredAttributeEval
- ClassifierSubsetEval
- ConsistencySubsetEval
- CfsSubsetEval
- FilteredAttributeEval
- FilteredSubsetEval
- WrapperSubsetEval

These methods were run with every compatible search method from *Weka*. The results that compare the different methods are in Section **4.3.4** since we tested them according to the performance of classifiers.

4.3.3. Cross-validation

K-fold cross-validation partitions the dataset into *k* mutually exclusive subsets, called folds, with approximatively the same size. In cross-validation, each fold contains the same proportion of instances from each class. Training and testing are executed *k* times, using in each iteration, one fold in the last and the others in the older ones. During the learning, each fold is used just once for testing and *k-1* times for training. The accuracy estimate is given by the overall number of correct classifications from the *k* iterations, divided by the amount of tuples in the initial dataset. In all the tests was used cross-validation

with 10 folds and a random seed with value of 1 for the performance evaluation of classification algorithms.

4.3.4. Classification

First, we will make a brief introduction of the four algorithms we used in this case study.

AdaBoost is machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire (Freund & Schapire, 1995). It can be used in conjunction with many other types of learning algorithms to improve their performance. The output of the other learning algorithms, normally weak learners, is combined into a weighted sum that represents the final output of the boosted classifier.

Decision Tree is a tree based model structure where each non-leaf node has a test on an attribute, each branch represents an outcome of the test and each leaf has a class label. This structure has the decisions and their possible consequences, including chance event outcomes, resource costs, and utility. ID3, C4.5 and CART are the most popular decision tree algorithms. We used the C4.5 algorithm for the tests (Quinlan, 1993), *J48* in *Weka*.

Naïve Bayes is a simple statistical classifier that predict class membership probabilities based on the Bayes' Theorem (John & Langley, 1995).

Random Forest is an ensemble learning method that operate by constructing a multitude of decision trees at training time and then output the class that appears most often in the classes that are output by individual trees (Breiman, 2001).

As we explained in Section **4.1**, we have three classes in our dataset {Fair, Good, Very Good} that will serve to train a model to anticipate the average grades of new students enrolling in MEIC. For the Classification, we used AdaBoost (AB), Decision Trees (DT), Naïve Bayes (NB) and Random Forest (RF). We used *Weka* implementation of these methods, along with the parameters described in the following paragraph.

AdaBoost with *NaiveBayes* classifier*,* number of iterations equal to 10 and seed to 1. Decision Tree without binary splits, confidence factor to 0.25 and seed equal to 1. Naïve Bayes without supervised discretization and executed without kernel estimator. Random Forest with 10 generated trees, unlimited maximum depth and seed equal to 1.

4.3.5. Results

Figure 8 has the results of classifiers accuracy for each method of feature selection. It should be noted that these results do not contain the biclusters attributes. We can observe that the FS method that obtains better accuracy in the four classification algorithms is *WrapperSubsetEval*.

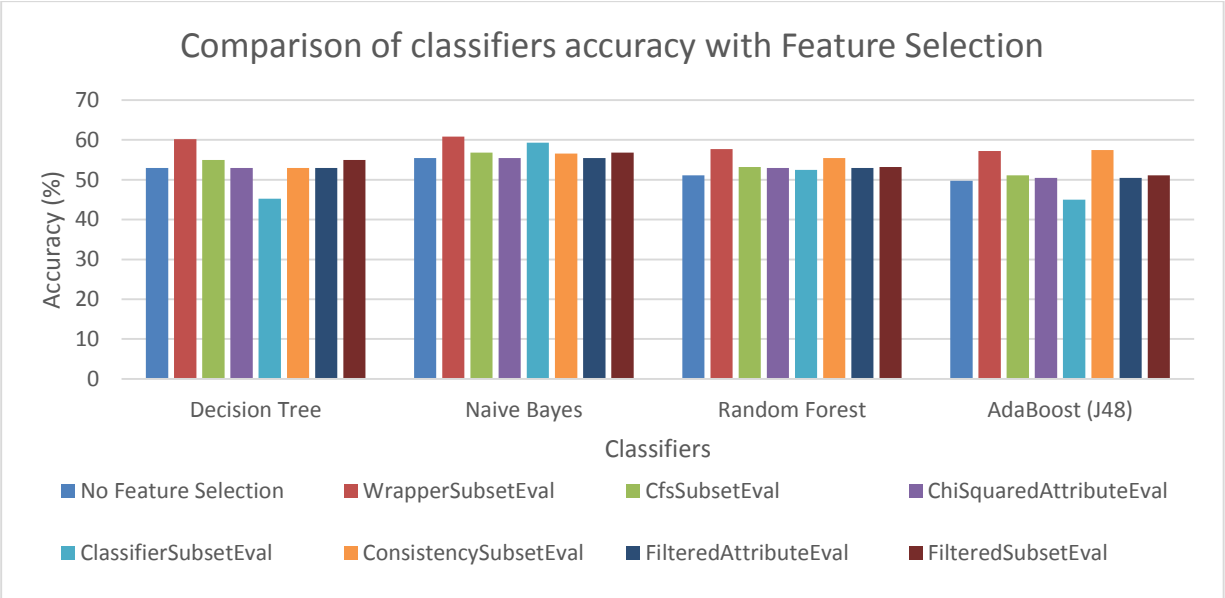

Figure 8. Comparison of classifiers accuracy with feature selection.

As the *WrapperSubsetEval* method was the one that had the best results, we did experiments putting the biclusters as attributes (as described at Section **4.3**) and compare the two classification algorithms who obtained better results with the FS, Naive Bayes and Decision Tree (see **Figure 9**). As we can verify the Naive Bayes always obtained better results compared to the Decision Tree, with or without biclusters.

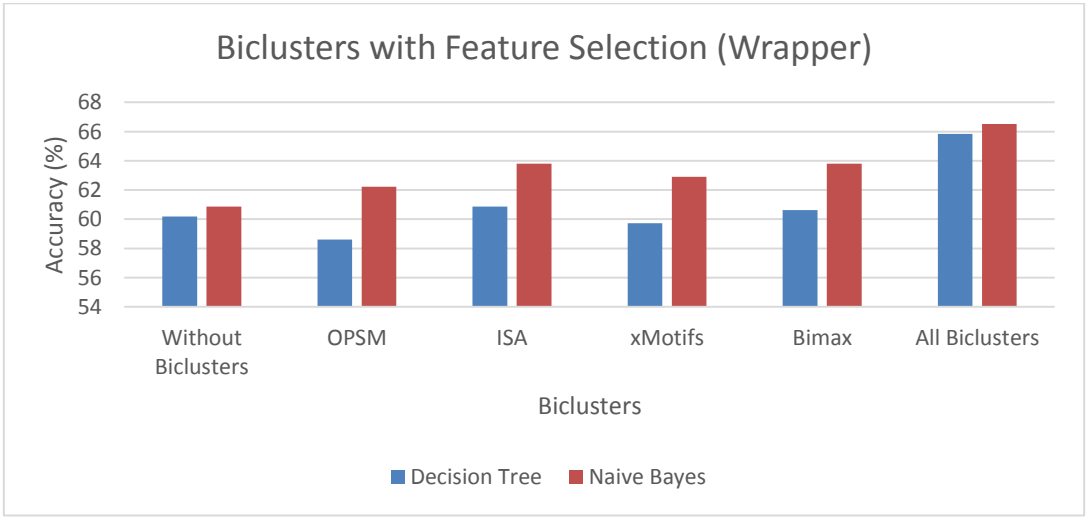

Figure 9. Comparison between Decision Tree and Naive Bayes using WrapperSubsetEval.

Figure 10 shows the result of applying the classification without biclusters and with biclusters. We can confirm that there are improvements in classifiers with biclusters compared with the results without biclusters. The accuracy of Naive Bayes got an increase of more than 5% (from 60.9% to 66.5%).

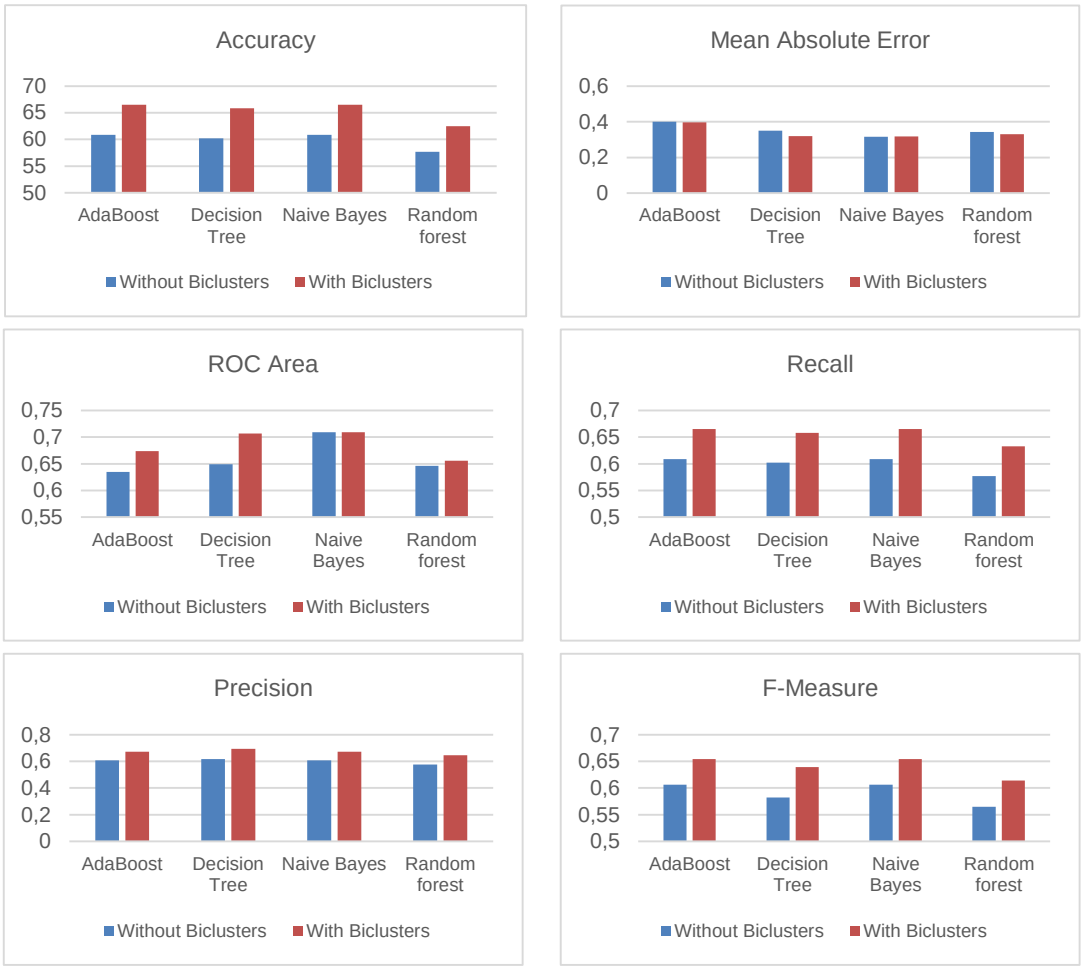

Figure 10. Overall results.

4.4. Biclustering and Frequent Item-Set Mining

One important topic of research in data mining is Pattern Mining. Its main purpose consists of extracting relevant patterns that occur frequently in data. These patterns can be item-sets, subsequences or other structures: graphs, trees, etc. Algorithms for frequent pattern mining can be classified into three categories, according to the types of patterns that they aim to obtain: transactional, sequential or structured pattern mining. We will only focus on the transactional algorithms since it approaches the type of biclustering patterns obtained in section **4.2**.

Transactional Pattern Mining can be defined as an item-set of a non-empty set of items that appear together in a transactional dataset. The patterns can be expressed in the form of association rules, as proposed originally by (R. Agrawal & Srikant, 1994).

Let $L = \{i_1, i_2, \ldots, i_n\}$ be a set of literals representing *items.* The problem of discovering frequent item-sets can be formally stated as follows: let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq L$ and has a *TID* (transaction id), which is a unique identifier. X is a set of some

items in *L*, or a *k-item-set* if it contains *k* items. A transaction is defined as a tuple *T = (TID, X).* A transaction is said to contain *X* if $X \subseteq T$.

The first method to be developed was the Apriori algorithm (R. Agrawal & Srikant, 1994), which employs an iterative approach known as level-wise search, where the database is scanned to find the frequent 1-item-sets, then using these to generate frequent 2-itemsets, until no more frequent *k*-item-sets can be generated for some *k*. An alternative frequent-pattern growth (or FP-growth) methods were proposed (Han, Pei, & Yin, 2000), which mine the complete set of frequent item-sets without the need to repeatedly scan the database and check the candidates by performing pattern matching, like Apriori algorithm does.

So, in order to have a base for comparison with the results obtained, we decided to compare with the Frequent Item-Set Mining (FIM) using the same methodology we used for the biclusters. We applied the discovered patterns as features to use in classification.

The algorithm used to obtain the patterns was FP-Growth (Han et al., 2000) with a minimum support of 2%. The value of 2% was chosen since we only have biclusters with more than ten students and 10/442 (size dataset) = ~ 2%. We initially got 26067 patterns, which are distributed as shown in **Table 8**, we removed all patterns that have only 1 or 2 subjects because they are not relevant, so we finished with 22323 patterns.

Table 8. Statistics of patterns of FIM.

Size of Patterns	Number of Patterns
1 subject	133
2 subjects	3611
3 subjects	15840
4 subjects	6223
5 subjects	260

On **Table 9** we can see examples of some patterns that FIM found. As we expected, the patterns found are very similar to the patterns found by *ISA* and *Bimax* algorithm because they can find the same type of patterns.

Table 9. Patterns with the most support found by FIM algorithm.

Algorithm	Pattern	Support (%)
FIM	(CDI-II, ACED, PEst)	18,3
	(CDI-II, ACED, CDI-I)	17,4
	(CDI-II, ACED, MO)	16,5
	(ACED, CDI-II, SSina)	16,3
	(CDI-II, PEst, CDI-I)	16,1
	(CDI-II, MO, PEst)	15,4
	(CDI-II, ACED, EO)	15,2
	(CDI-II, MO, PEst, CDI-I)	10
	(CDI-II, ACED, MO, CDI-I)	10
	(CDI-II, ACED, MO, PEst)	9,7
	(CDI-II, ACED, CDI-I, EO)	8,8
	(CDI-II, ACED, PEst, CDI-I)	8,8
	(ACED, CDI-II, IPM, SSina)	8,6
	(ACED, CDI-II, CDI-I, EO, MD)	5,7
	(CDI-II, MO, ACED, PEst, CDI-I)	5,7
	(CDI-II, ACED, EO, PEst, MD)	5,2
	(CDI-II, ACED, CDI-I, MO, LP)	5,2
	(CDI-II, CDI-I, ACED, MO, EO)	5,2
	(CDI-II, MO, PEst, CDI-I, CGra)	5
	(CDI-II, ACED, PEst, EO, Com)	5

We then placed the patterns obtained with FIM on a matrix, like we did with the biclusters, and run a classifier (*Naive Bayes*, in this case) and finally compared the results obtained with the results obtained with the biclusters (**Figure 11**). As expected, we obtained better results with the biclusters since we can find more types of patterns than with Frequent Item-Set Mining.

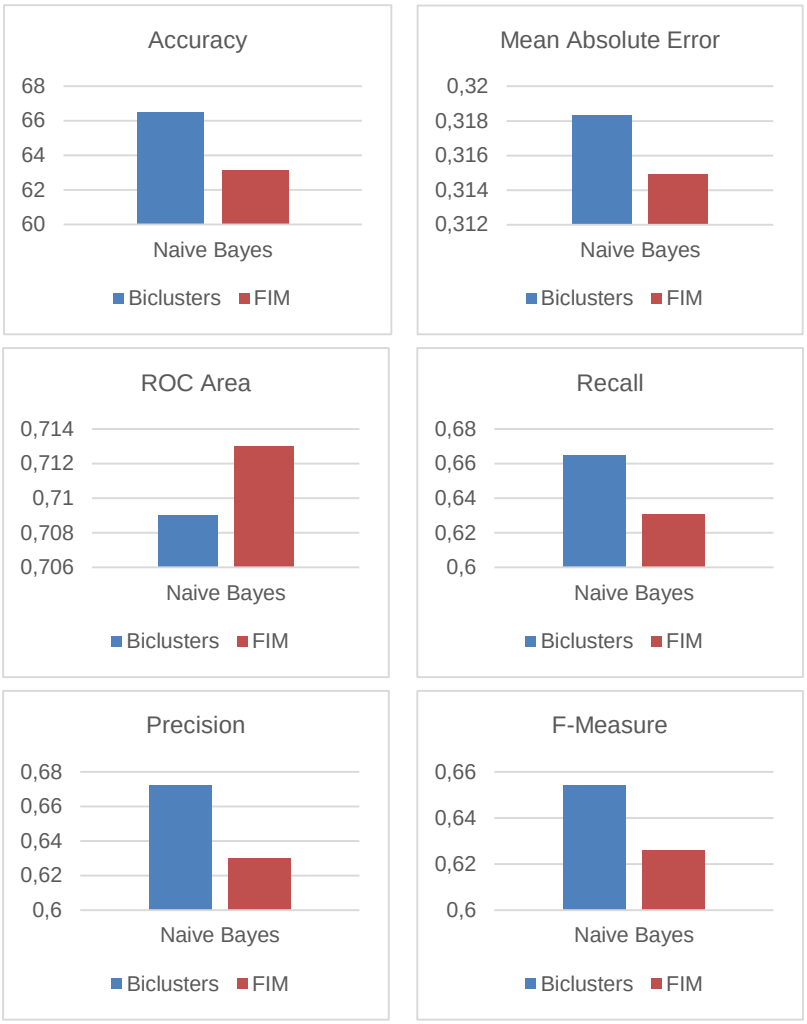

Figure 11. Comparison between classification with biclusters and classification with FIM patterns.

Summary

In this chapter we applied four biclustering algorithms, the OPSM, xMotifs, ISA and Bimax. The biclusters obtained were then chosen by a method of feature selection for future use as a feature in classifier, which serves to predict students' grades in the course specialization. With the inclusion of biclusters in classifier we could increase accuracy in more than 5%. Finally, we compared the results obtained with the patterns of Frequent Item-Set Mining (FIM) using the same methodology, getting a better result with biclusters than with FIM.

Chapter 5

Case Study - Subjects' Approval Rate

In this chapter, we introduce a new type of patterns that can be discovered with biclustering and create a concrete case study to obtain these patterns in educational data.

We decided to use a different type of matrix where in this case we use in the column a time variable to realize if we can get some interesting results with the biclustering patterns applied to the EDM. First we will make a brief introduction to biclustering with time series and its related work.

5.1. Biclustering Over Time

Biclustering with time series are typically used for expression time series in gene expression. The great difference of biclusters with time series for biclusters that we have mentioned previously is that, in this case, we are always interested to have biclusters that have continuous columns, which correspond to coherent expression patterns shared by a group of rows in consecutive time points.

CCC-Biclustering algorithm (S. C. Madeira et al., 2010) was developed to work with genes and uses a generalized suffix tree to identify, in time linear on the size of the expression matrix, all maximal biclusters with contiguous columns that exhibit coherent expression evolutions over time.

Then, we will see how this algorithm behaves in educational data on a specific matrix and understand whether these patterns bring somewhat important information for education.

5.2. Data Analysis

In this case study we also used data from the *Educare* project[5]. But we made a matrix that has the subjects in rows and a time variable in columns (semesters of the subjects). The values of the matrix corresponds to the number of approved students, in percentage, in the subject of the respective semester.

In **Table 10** we have the matrix that we used for this case study (we are already showing the discretized matrix), with 30 subjects of graduation program (LEIC) as rows and six semesters of these subjects as columns, which is related to six academic years (annual subjects). Unfortunately it is a matrix that has

[5] Project *Educare* - https://sites.google.com/site/istprojecteducare/

a small number of instances, but this is one of the major problems of EDM, the obtaining of data with many instances. Though, with the globalization of technology in most schools, we are starting to get more complete and accurate data for research.

Table 10. Data Matrix used to obtain the bicluster patterns.

Subjects	S1	S2	S3	S4	S5	S6
AL	B	C	D	D	D	B
ACED	B	D	D	C	D	D
ASA	C	C	B	C	B	E
AC	C	D	D	D	C	C
BD	C	B	B	A	A	A
CDI1	C	E	D	E	D	D
CDI2	B	D	C	D	C	D
Comp	C	C	C	C	B	E
CG	C	C	B	D	B	E
EO	D	D	D	C	D	D
ES	B	B	B	A	A	NA
FP	C	D	C	C	C	D
Ges	C	C	C	C	B	C
IA	D	B	B	B	B	C
IPM	A	A	A	A	A	A
IAED	C	D	D	C	B	C
LP	C	B	C	C	B	B
MD	B	C	D	C	B	C
MO	D	D	E	C	C	C
Mod	A	A	B	A	A	E
PP1	A	B	A	B	B	B
PP2	A	A	C	A	B	E
PE	A	C	E	E	D	D
PO	B	C	C	C	C	C
Redes	B	B	B	B	A	A
SD	C	D	D	C	C	C
SDist	B	B	B	A	B	NA
SS	B	C	C	C	C	C
SO	B	C	D	C	C	C
TC	C	C	C	C	B	B

The data of rate approvals was in percentage with a range between 0% and 100% (0% corresponds to saying that zero students were approved and 100% that all enrolled students were approved), however, the matrix was discretized to CCC-biclustering algorithm present the patterns properly. In **Table 11** is the correspondence between the intervals of approval rates percentages that we defined and symbols (five symbols).

Table 11. Correspondence between symbols and approval rate.

Symbols	Approval Rate
A	80% < A ≤ 100%
B	60% < B ≤ 80%
C	40% < C ≤ 60%
D	20% < D ≤ 40%
E	0% ≤ E ≤ 20%

5.3. Biclustering Analysis

To obtain the bicluster patterns we used a tool that has the CCC-Biclustering algorithm implemented, the *BiGGEsTS* (Gonçalves, Madeira, & Oliveira, 2009).

In **Table 12** we put a statistical summary of the biclusters obtained with the matrix above (**Table 10**). We obtained 39 patterns, which for the size of the matrix are not few. Generally the patterns are small, with an average of 3 rows by 3 columns.

Table 12. Statistics of biclusters obtained.

Algorithm	Number of biclusters	Average Size	Average number of lines	Average number of columns
CCC-Biclustering	39	8	3	3

In **Figure 12** we show some of the biclusters obtained by applying the CCC-Biclustering algorithm. Biclusters were chosen taking into account their support and their evident pattern. In the following section we will refer to some conclusions that we can get just by looking at the pattern of biclusters.

B1	S1	S2	S3	S4	S5	S6
PO	B	C	C	C	C	C
SS	B	C	C	C	C	C

B2	S2	S3	S4	S5
IA	B	B	B	B
IPM	A	A	A	A

B3	S4	S5	S6
BD	A	A	A
IPM	A	A	A
MO	C	C	C
SD	C	C	C

B4	S1	S2	S3	S4	S5
Comp	C	C	C	C	B
Ges	C	C	C	C	B

B5	S3	S4	S5
FP	C	C	C
IA	B	B	B
IPM	A	A	A

B8	S2	S3	S4
Comp	C	C	C
Ges	C	C	C
PO	C	C	C
SS	C	C	C

B6	S1	S2	S3	S4
Mod	A	A	B	A
PP2	A	A	C	A

B7	S3	S4	S5	S6
AC	D	D	C	C
LP	C	C	B	B

B9	S1	S2	S3	S4	S5	S6
Redes	B	B	B	B	A	A
TC	C	C	C	C	B	B

B10	S1	S2
IPM	A	A
Mod	A	A
PP2	A	A

Figure 12. Examples of biclusters obtained with CCC-Biclustering algorithm.

5.4. Evaluation

We can recognize that the patterns of biclusters have some interesting behaviours (**Figure 12**). For example, in bicluster *B1* we can realize that the subjects PO and SS have almost always the same behaviour (and the same happens in the bicluster *B4*). Or, in bicluster *B2*, we can perceive that despite having different approval rates they have always been constant. And the same can be assumed to bicluster *B3* and *B5* for different periods. In bicluster *B9,* we can realize that in semester S5 occurred changes in TC and Redes subjects who did improved the approval rate. The *B7* bicluster has also a similar behaviour.

The *B6* bicluster has an interesting behaviour because in semester S3 the rates decreased, which may indicate some evidence of some negative event that happened in that period.

Summary

In this chapter we used as a case study a matrix with temporal data. For get the biclusters we used an algorithm prepared for these cases, the CCC-Biclustering algorithm. Although the matrix it is small, we can extrapolate some interesting results obtained with these biclusters, such as some subjects' behaviours.

Chapter 6

Conclusion

In this chapter, we present the main conclusions of this work and we point to some future work.

6.1. Conclusions

In this dissertation, we have proposed to explore biclustering to discover new patterns in educational data and make use of these patterns to enrich training data in order to improve the prediction of students' performance.

With this work we covered two very challenging areas, Education Data Mining and Biclustering, connecting them, to best of our knowledge, for the first time.

Something that proved difficult was the pre-processing, because the data had very few instances and had many missing values. We also had to test various parameters of biclusters algorithms to reach to more consistent results.

Overall, we tested two different approaches and showed what is possible with the biclustering in EDM.

The classification results were obtained via feature selection, k-fold cross validation, and expressed in terms of accuracy, mean absolute error, ROC area, recall, precision and f-measure. To compare on the best combination of these techniques, we analyzed their overall results through a series of charts. The results we had with the biclusters were also interpreted taking into account the context of data matrix.

Results were satisfactory and the objectives achieved were successful. The different studies have enabled a new perspective on the application of biclustering in education context. With these case studies we fulfilled one objective of this dissertation that was to introduce and study the suitability and adaptability of biclustering algorithms to the problem of analysis of educational data.

In conclusion, we sincerely hope that this work be a starting point to new collaborations about biclustering in EDM.

6.2. Future Work

Despite our contribution with biclustering in EDM, we feel that we only did a small part of what is possible with this technique in this area. There are multiple possible problems with different type of educational data that can bring new interesting results and that should be explored.

Many other approaches can be tested but one that we think would be interesting to experiment are three-dimensional (3D) datasets. For example, we can create a matrix that has a dimension with subjects, a second dimension with the various types of evaluations of the subjects (approval rates, satisfaction of students, average grades, etc.) and a third dimension will be the semesters (time variable). With this we can realize new interesting behaviors that subjects had over the years.

There are already algorithms that address three-dimensional matrices that can be adapt to the context of education. For example, we have the *triCluster* (Zhao & Zaki, 2005) developed by Zhao and Zaki, that can mine arbitrarily positioned and overlapping clusters, and depending on different parameter values, it can mine different types of clusters, including those with constant or similar values along each dimension, as well as scaling and shifting expression patterns.

Another interesting future work would be to develop more efficient techniques using domain knowledge to choose the biclusters that have more relevance for the problem at hand, as it is widely used in gene expression.

References

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD 98*. ACM Press.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, *1215*, 487–499. doi:10.1.1.40.6757

Amershi, S. (2009). Combining unsupervised and supervised classification to build user models for exploratory learning environments. *Journal of Educational Data Mining*.

Antunes, C. (2008). Acquiring Background Knowledge for Intelligent Tutoring Systems. *Educational Data Mining 2008*.

Ayers, E., Nugent, R., & Dean, N. (2009). A comparison of student skill knowledge estimates. *Proc. Int. Conf. Educ. Data Mining*.

Barker-Plummer, D., Cox, R., & Dale, R. (2009). Dimensions of difficulty in translating natural language into first order logic. *Educational Data Mining*.

Barkow, S., Bleuler, S., Prelic, A., Zimmermann, P., & Zitzler, E. (2006). BicAT: a biclustering analysis toolbox. *Bioinformatics (Oxford, England)*. doi:10.1093/bioinformatics/btl099

Barracosa, J., & Antunes, C. (2011). Anticipating teachers' performance. *KDD 2011 Workshop*.

Ben-Dor, A., & Chor, B. (2003). Discovering local structure in gene expression data: the order-preserving submatrix problem. *Journal of Computational*. doi:10.1089/10665270360688075

Bergmann, S., Ihmels, J., & Barkai, N. (2003). Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical Review E*.

Bozdağ, D., Kumar, A. S., & Catalyurek, U. V. (2010). Comparative analysis of biclustering algorithms. *BCB '10*. doi:10.1145/1854776.1854814

Breiman, L. (2001). Random forests. *Machine Learning*, 5–32. doi:10.1023/A:1010933404324

Chen, C.-M., Chen, M.-C., & Li, Y.-L. (2007). Mining Key Formative Assessment Rules based on Learner Profiles for Web-based Learning Systems. *ICALT*. doi:10.1109/ICALT.2007.189

Cheng, Y., & Church, G. M. (2000). Biclustering of expression data. *International Conference on Intelligent Systems for Molecular Biology*.

Cobo, G., & García-Solórzano, D. (2011). Modeling Students' Activity in Online Discussion Forums: A Strategy based on Time Series and Agglomerative Hierarchical Clustering. *Educational Data Mining*.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. *KDD '01*.

Eren, K., Deveci, M., Küçüktunç, O., & Çatalyürek, Ü. V. (2013). A comparative analysis of biclustering algorithms for gene expression data. *Briefings in Bioinformatics*.

Ezen-Can, A., & Boyer, K. E. (2013). Unsupervised Classification of Student Dialogue Acts With Query-Likelihood Clustering. *Educational Data Mining*.

Freund, Y., & Schapire, R. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. *Computational Learning Theory*, *55*(1).

Gonçalves, J. P., Madeira, S. C., & Oliveira, A. L. (2009). BiGGEsTS: integrated environment for biclustering analysis of time series gene expression data. *BMC Research Notes*, *2*, 124.

Gu, J., & Liu, J. S. (2008). Bayesian biclustering of gene expression data. *BMC Genomics*.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software. *ACM SIGKDD Explorations*, *11*(1), 10–18. doi:10.1145/1656274.1656278

Hämäläinen, W. (2004). Data mining in personalizing distance education courses. *Proceedings of the 21st ICDE*.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*. *Soft Computing* (Vol. 54, p. 703). Morgan Kaufmann.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *ACM SIGMOD Record*. doi:10.1145/335191.335372

Hardof-jaffe, S., Hershkovitz, A., Abu-kishk, H., Bergman, O., & Nachmias, R. (2009). How do Students Organize Personal Information Spaces? *Educational Data Mining*.

Hartigan, J. (1972). Direct clustering of a data matrix. *Journal of the American Statistical Association*. doi:10.1080/01621459.1972.10481214

Hernandez-del-Olmo, F., Montero, M., Gaudioso, E., & Talavera, L. (2009). Supporting teachers in collaborative student modeling: A framework and an implementation. *Expert Systems with Applications*.

Hershkovitz, A., & Nachmias, R. (2009). Developing a log-based motivation measuring tool. In *Proceedings of 1st International Conference on Educational Data Mining*.

Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., Kasim, A., … Clevert, D.-A. (2010). FABIA: factor analysis for bicluster acquisition. *Bioinformatics (Oxford, England)*.

Huttenhower, C., Mutungu, K., Indik, N., & Yang, W. (2009). Detailing regulatory networks through large scale data integration. *Bioinformatics*.

John, G. H. G., & Langley, P. (1995). Estimating Continuous Distributions in Bayesian Classifiers. In *CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE. Montreal, Quebec, Canada* (Vol. 1, pp. 338–345). Morgan Kaufmann.

Kaser, T., Busetto, A. G., Solenthaler, B., Kohn, J., Aster, M. von, & Gross, M. (2013). Cluster-Based Prediction of Mathematical Learning Patterns. *AIED*.

Lazzeroni, L., & Owen, A. (2002). Plaid Models for Gene Expression Data. *Statistica Sinica*.

Li, G., Ma, Q., Tang, H., Paterson, A. H., & Xu, Y. (2009). QUBIC: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic Acids Research*.

Li, N., Cohen, W. W., & Koedinger, K. R. (2013). Discovering Student Models with a Clustering Algorithm Using Problem Content. In *Educational Data Mining*.

Liu, J., & Wang, W. (2003). OP-cluster: clustering by tendency in high dimensional space. *Third IEEE International Conference on Data Mining*.

Madeira, S. C., Teixeira, M. C., Sá-Correia, I., & Oliveira, A. L. (2010). Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.

Madeira, S., & Oliveira, A. (2004). Biclustering algorithms for biological data analysis: a survey. *Biology and Bioinformatics, IEEE*.

Mirkin, B. (1996). *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers.

Murali, T., & Kasif, S. (2003). Extracting conserved gene expression motifs from gene expression data. *Pacific Symposium on Biocomputing*.

Myller, N., Suhonen, J., & Sutinen, E. (2002). Using data mining for improving web-based course design. *International Conference on Computers in Education, 2002. Proceedings.* doi:10.1109/CIE.2002.1186125

Nugent, R., Ayers, E., & Dean, N. (2009). Conditional subspace clustering of skill mastery: identifying skills that separate students. In *Educational Data Mining*.

Nugent, R., Dean, N., & Ayers, E. (2010). Skill set profile clustering: the empty K-means algorithm with automatic specification of starting cluster centers. In *Educational Data Mining*.

Perera, D., Kay, J., Koprinska, I., Yacef, K., & Zaiane, O. R. (2009). Clustering and Sequential Pattern Mining of Online Collaborative Learning Data. *IEEE Transactions on Knowledge and Data Engineering*.

Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., … Zitzler, E. (2006). A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics (Oxford, England)*, *22*(9), 1122–9.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. (M. Kaufmann, Ed.)*Morgan Kaufmann San Mateo California* (Vol. 1, p. 302). Morgan Kaufmann.

Ritter, S., Harris, T. K., Nixon, T., Dickison, D., Murray, R. C., & Towle, B. (2009). Reducing the knowledge tracing space. In *Proceedings of International Conference on Educational Data Mining*.

Romero, C., & Ventura, S. (2010). Educational Data Mining: A Review of the State of the Art. *Trans. Sys. Man Cyber Part C*. doi:Doi 10.1109/Tsmcc.2010.2053532

Romero, C., Ventura, S., Pechenizkiy, M., & Baker, R. (2010). *Handbook of Educational Data Mining*. *lavoisierfr* (p. 535). CRC Press.

Sharan, R., Maron-Katz, A., & Shamir, R. (2003). CLICK and EXPANDER: a system for clustering and visualizing gene expression data. *Bioinformatics (Oxford, England)*. doi:10.1093/bioinformatics/btg232

Shih, B., Koedinger, K., & Scheines, R. (2010). Unsupervised discovery of student learning tactics. In *Educational Data Mining*.

Talavera, L., & Gaudioso, E. (2004). Mining student data to characterize similar behavior groups in unstructured collaboration spaces. In *Proceedings of Workshop on Artificial intelligence in CSCL*.

Tanay, A., Sharan, R., & Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*. doi:10.1093/bioinformatics/18.suppl_1.S136

Tang, T. Y., & McCalla, G. (2002). Student modeling for a web-based learning environment: a data mining approach. Retrieved from http://dl.acm.org/citation.cfm?id=777092.777246

Tian, F. T. F., Wang, S. W. S., Zheng, C. Z. C., & Zheng, Q. Z. Q. (2008). Research on e-learner personality grouping based on fuzzy clustering analysis. *2008 12th International Conference on Computer Supported Cooperative Work in Design*. doi:10.1109/CSCWD.2008.4537122

Trivedi, S., Pardos, Z. A., & Heffernan, N. T. (2011). Clustering Students to Generate an Ensemble to Improve Standard Test Score Predictions. In A. Biswas, G and Bull, S and Kay, J and Mitrovic (Ed.), *AIED* (Vol. 6738).

Trivedi, S., Pardos, Z. A., Sarkozy, G. N., & Heffernan, N. T. (2012). Co-Clustering by Bipartite Spectral Graph Partitioning for Out-of-Tutor Prediction. *International Educational Data Mining Society*. Retrieved from http://eric.ed.gov/?id=ED537191

Trivedi, S., Pardos, Z. A., Sárközy, G. N., & Heffernan, N. T. (2011). Spectral Clustering in Educational Data Mining. In *Educational Data Mining*.

Vale, A., Madeira, S. C., & Antunes, C. (2014). Mining Coherent Evolution Patterns in Education through Biclustering. In *Educational Data Mining*.

Ventura, S., Bra, P. De, & Romero, C. (2004). Knowledge Discovery with Genetic Programming for Providing Feedback to Courseware Authors. *User Modeling and User-Adapted Interaction*. doi:10.1007/s11257-004-7961-2

Ventura, S., & Romero, C. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*. doi:10.1016/j.eswa.2006.04.005

Verma, N. K., Singh, A., & Cui, Y. (2010). A comparison of biclustering algorithms. *Systems in Medicine*.

Zakrzewska, D. (2008). *Cluster analysis for user's modeling in intelligent elearning systems*. (N. T. Nguyen, L. Borzemski, A. Grzech, & M. Ali, Eds.) (Vol. 5027). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-69052-8

Zhang, K. Z. K., Cui, L. C. L., Wang, H. W. H., & Sui, Q. S. Q. (2007). An Improvement of Matrix-based Clustering Method for Grouping Learners in E-Learning. *2007 11th International Conference on Computer Supported Cooperative Work in Design*.

Zhao, L., & Zaki, M. J. (2005). triCluster : An Effective Algorithm for Mining Coherent Clusters in 3D Microarray Data. *Sigmod*, 694–705.

Zukhri, Z., & Omar, K. (2007). Solving new student allocation problem with genetic algorithms: A hard problem for partition based approach. *J. Zhejiang Univ.*

Annexes

Annex 1 – Examples of biclusters

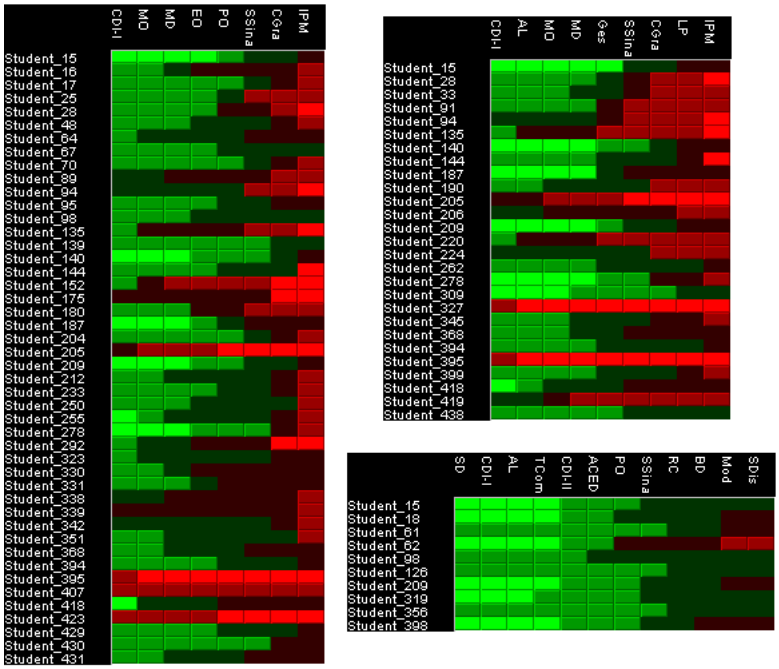

Figure 13. Examples of OPSM biclusters.

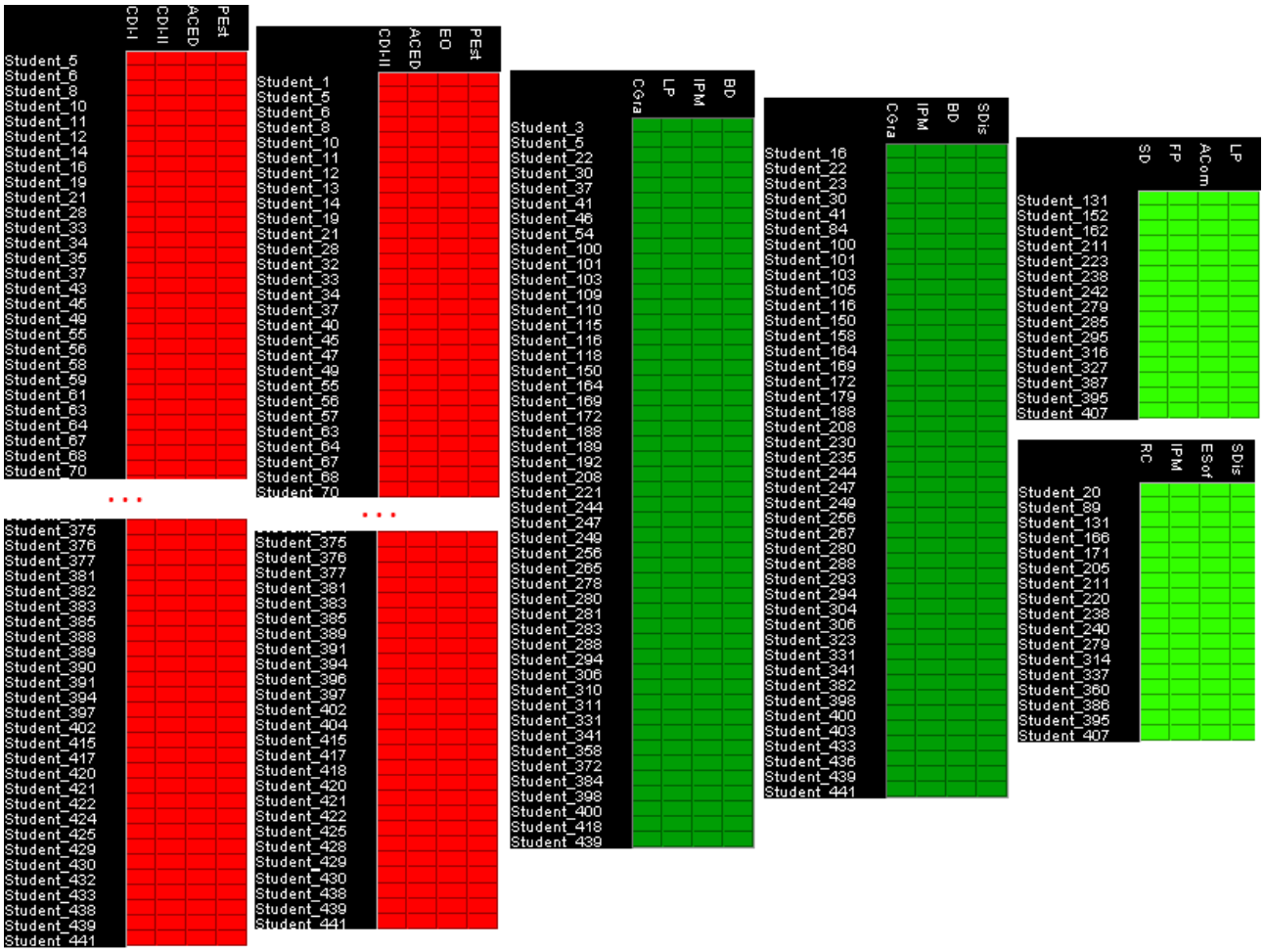

Figure 14. Examples of Bimax biclusters.

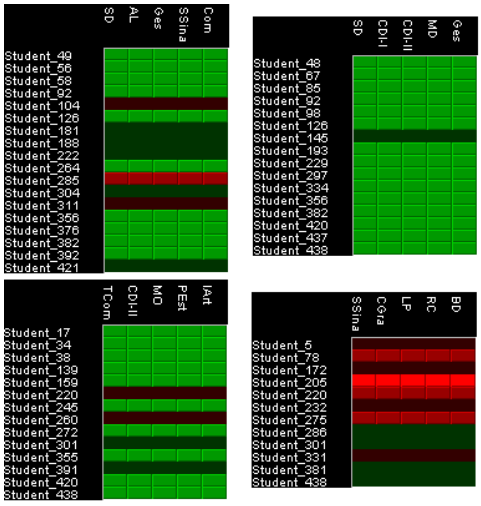

Figure 15. Examples of xMotifs biclusters.

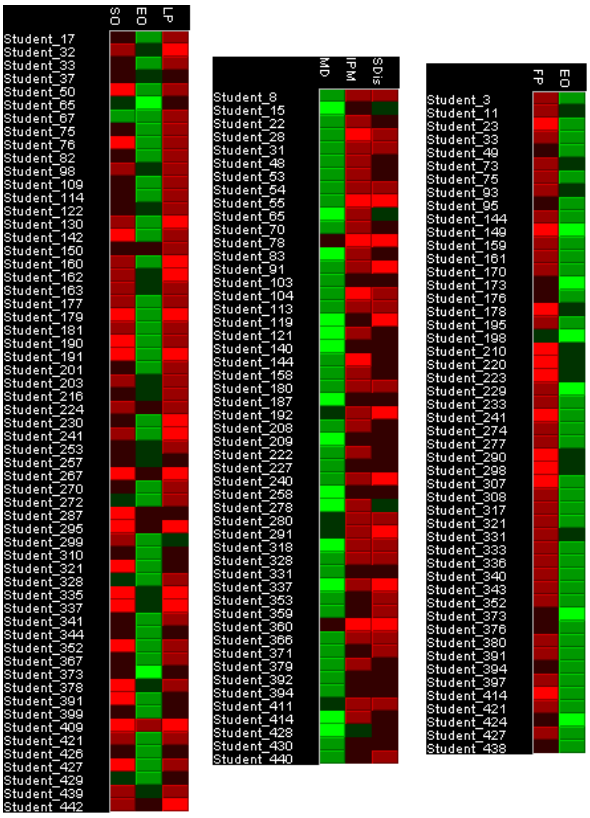

Figure 16. Examples of ISA biclusters.