Università di Firenze, Università di Perugia, INdAM consorziate nel CIAFM

DOTTORATO DI RICERCA
IN MATEMATICA, INFORMATICA, STATISTICA

CURRICULUM IN INFORMATICA
CICLO XXXI

Sede amministrativa Università degli Studi di Firenze
Coordinatore Prof. Graziano Gentili

Community Elicitation from Co-occurrence of Activities

Settore Scientifico Disciplinare INF/01

Dottorando:
Paolo Mengoni

Tutore
Prof. Alfredo Milani

Coordinatore
Prof. Graziano Gentili

Anni 2015/2018

Learn to fly

know how to land

Abstract

A massive amount of data is continuously generated by the activities that people hold on online platforms, mobile systems and in general using and interacting with digital systems. Big data, not directly related to social networks, are generated in large quantities from systems that are not necessarily social systems. In fact, in the information society a whole lot of activities take place on systems that were not developed to support explicit social interactions between users.

In this thesis, starting from the observation of users activities within a system, we want to investigate if it is possible to characterise the existence of social relationships among them. As people interact, individually or in groups, we want to elicit their social communities from the temporal and spatial co-occurrence of their activities. The key assumption of this work is that we suppose that there are multiple, parallel, hidden communication channels and social networks where social interactions take place among users and which determine the observed emergent phenomenon of actions co-occurrences.

The main original contribution of this thesis is the proposal of innovative methodologies for users community discovery from implicit user-system interactions and their experimental evaluation. The *History Based Analysis* approach is a novel approach we have introduced, that exploits the similarity of users' activity histories to discover the hidden social communities. To better characterise the histories *binary correlation measures* we have introduced and experimented original *entropy*

amplification factors that take in account system wide distribution of activities at a given time to contextualise the user activity similarities. The other relevant introduced approach, the *Session Based* method, uses graph based representation of concurrent users' sessions to elicit the hidden social communities. Both proposed approaches have been validated using a real world dataset containing the activity logs of students using a virtual learning environment platform.

A remarkable result of our work has been to confirm that co-occurrence of people activities is an emerging epiphenomenon of hidden, implicit information exchanges through side channel communications. Therefore the observation of co-occurrence of events can be used to elicit social relationships. Interesting extension of this work include the analysis of real world co-occurrences, like in the case of people, personal vehicles or other personal objects occurring in the same physical place at the same time, and in general wherever it co-occurrence can be seen as an emerging epiphenomenon of people's relationships and information exchange.

Potential applications of this thesis work can fall in various areas such as business, marketing, public administration, including intelligence and military sectors. Experimental evaluation of the introduced methodologies through tests held in the domain of eLearning demonstrated the effectiveness of our proposed approaches in retrieving hidden social communities.

Keywords: Community Detection, Graph Analysis, Similarity Measures, Social Networks, Modularity, Graph Modelling, Learning Analytics, ...

Acknowledgements

In first place I would like to record gratitude to my principal supervisor, Prof. Alfredo Milani, for giving me the opportunity to work with his supervision. His valuable feedback gave very insightful perspectives on various aspects of the research. His mentoring extends beyond academic research and I am thankful for his continued guidance and encouragement.

I also want to express sincerely thanks to Prof. Jiming Liu and Dr. William Cheung as their kind guidance, advises, great inspirations and generous help. They took time out of their busy, Hong Kong-style, schedules to provide valuable feedback and constructive suggestions on my research work.

In the course of my doctoral study, I have had the opportunity to collaborate with Dr. Valentina Franzoni whose ideas helped - and stressed - me during this study path. I would like to thank the *sixth floor*'s group fellows Dr. Valentino Santucci, Dr. Andrea Chiancone, Dr. Francesco Santini and Dr. Valentina Poggioni, among the other researcher in DMI, for their support, understanding and the beers. Moreover, I want to thank the researchers and staff in Hong Kong Baptist University. Among the others, this special thank is for Yuanxi Li that supported me thoroughly during my stays in Hong Kong.

Besides, I would like to thank my lifelong friends Marco Mencacci, Maurizio Montarani, Antonello Bianchi, Riccardo Martinelli, their families and all the other friends. Without their support I would not be able to reach this objective.

I would like to express my profound gratitude to my father Francesco and my mother Anna, she's now with God and she would be proud of this. They continuously encouraged me without questions and pushed me to go on. I would like to thank my brother Matteo, my sister in law Lina, and their daughters Teresa and Martina, my preferred source of distraction.

Summary

Starting from the big data generated by the observation of people's activities, this thesis we have investigated the possibility of finding the underlying social communities. In Chapter 1, we have identified and motivated the interest in discovering the social behaviour and interactions from the activities of people. The main objective of this thesis is then set to find if the co-occurrence of peoples activities can convey information about their underlying social relationships and communications.

The aim of the literature review, in Chapter 2, is to examine the main contributions that in the year have been proposed to investigate the social communities. Different approaches have been used to explore the community detection problem. Some of them use a network-based approach that exploits the measures and topology of the networks. Other methodologies use the characterisation of individuals using specific features and clustering them by exploiting their similarity. Information diffusion-based approaches use the timing of information to infer the underlying social networks and hidden diffusion patterns.

The behaviour discovery from different points of view have been explored in the works presented in Chapter 3. The behaviour discovery is analysed starting from explicit and implicit interactions among the individuals. Explicit interactions are the base for Facebook posts and comments analysis. The content of the short text messages have been analysed from the perspectives of determining the contextual information and emotion conveyed by posts and their comments. Explicit interac-

tions are the aim of mass behaviour modelling to solve the classical problem *tragedy of the commons*. Behaviour discovery using implicit interactions is the scope of the further, challenging works. In the first work we propose a novel framework for students activities analysis that take in account quantitative information to present visual interfaces morphing metaphors.

Innovative community discovery approaches from implicit interactions have been presented in Chapter 4. History Based Analysis approach is the first approach we introduced in this thesis. We developed a novel approach that exploits the similarity of users' activity histories to discover the hidden social communities. To better characterise the histories similarities we developed and tested various entropy amplification factors. These factors take in account system wide factors to contextualise the activities similarities. Session based approach use graph based representation of concurrent users' sessions to elicit the hidden social communities. Both the approaches have been tested on a real world dataset where are recorded the activities of students on a virtual learning environment. Tests demonstrated the effectiveness of our approaches in retrieving the hidden social communities.

The experimental results comparison of the proposed approaches have been summarised in Chapter 5.

Finally, we draw conclusions and point out to some future research directions.

Table of Contents

Chapter 1 Scope, Challenges and Objectives **1**

1.1 Discover and model social behaviour from activities 2

1.1.1 Co-occurrence of Co-located Activities 3

1.1.2 Time Co-occurrence 3

1.1.3 Space Co-location 4

1.2 Modelling of Interactions 6

1.2.1 Explicit 6

1.2.2 Implicit 7

1.3 Challenges and Thesis Objectives 8

Chapter 2 Social Community Detection **10**

2.1 Network-based communities 11

2.1.1 Graphs and Measures 11

2.1.2 Centrality Measures 14

2.1.3 Community Detection 16

2.2 Feature-based communities 21

2.2.1 Clustering Algorithms 22

2.3 Behavioural-based communities 25

2.3.1 Vector Similarities 25

2.3.2 Neural Networks 30

2.4 Information Diffusion-based communities 32

Chapter 3 Investigating Behaviour and Communities 34

3.1 Behaviour from Explicit Interactions 35

3.1.1 Qualitative classification of interactions 35

3.1.2 Sentiment polarity and emotion from interactions 37

3.2 Mass Behaviour Modelling 42

3.3 Behaviour Quantitative Analysis Tool 44

Chapter 4 Community Discovery Approaches 50

4.1 Problem Description, Domain and Definitions 51

4.1.1 Dataset 53

4.1.2 Activity Abstraction Level 53

4.1.3 Sessions 55

4.1.4 Granularity 57

4.1.5 Ground Truth 58

4.1.6 Evaluation Criteria 59

4.2 Histories Binary Correlation Analysis 62

4.2.1 Interaction Modelling 62

4.2.2 History Similarity 64

4.2.3 Amplifying Parameters 67

4.2.4 Measures 70

4.2.5 Clustering 71

4.2.6 Experiments and Evaluation Criteria 71

4.2.7 History Binary Correlation Discussion 72

4.3 Session Based Analysis 81

4.3.1 Interaction modelling 82

4.3.2 Activities co-occurrence graph 84

4.3.3 Sessions graph 85
4.3.4 Interactions graphs 86
4.3.5 Community Detection by Graph Analysis 87
4.3.6 Experiments and Evaluation Criteria 88
4.3.7 Session Based Analysis Discussion 88

Chapter 5 Experimental Results Evaluation 94

Conclusions 97

Bibliography 101

Appendices 114

Appendix A Histories Binary Correlation Results 114

Appendix B Session Based Analysis Graph Representations 125

Publications 136

List of Tables

2.1 Contingency table for pairwise vector comparison 27

4.1 Composition of the courses included in dataset. 59
4.2 Contingency table for pairwise students' activities 66
4.3 Experiment's results using different aggregation functions for Session Based Analysis on C1 course. 93
4.4 Experiment's results using different aggregation functions for Session Based Analysis on C2 course. 93

5.1 Community size analysis on course C1 and C2 using the best strategies for each approach. 96
5.2 Sample group assignment where a cluster is formed by elements from two different Ground Truth groups. 98
5.3 Sample group assignment where a Ground Truth group is split between two different clusters. 98

A.1 Complete Experiments Results for History Binary Analysis on C1 course. 114
A.2 Complete Experiments Results for History Binary Analysis on C2 course. 120

List of Figures

2.1 One step of Louvain algorithm for modularity maximization. 19

2.2 Pairwise vectors comparison to compute similarity parameters. 27

3.1 Context tag cloud sample of a Facebook post and its comments. . . . 37

3.2 Ontology based emotion similarity. 39

3.3 Sentiment polarisation tag cloud sample of a Facebook post and its comments using the path-based emotion abstraction approach. 40

3.4 Dimensional Morphing Metaphor for learners' monitoring. 47

3.5 Bar Morphing Metaphor for learners' monitoring. 48

3.6 Tag Cloud Morphing Metaphor for learners' monitoring. 48

4.1 Interactions distribution during course using 1h timeslots. 55

4.2 Number of accesses by user during course. 56

4.3 Representation of simple sessionsand separation example. 57

4.4 Sessions representation and separation example. 57

4.5 *Bucketing* of various activities. 58

4.6 Visual scheme of the Histories Binary Correlation Analysis approach. 63

4.7 History Summarised Vector with binary values representing the presence of interactions. 65

4.8 History Summarised Vectors pairwise comparison to extract measure parameters. 67

4.9 Visual representation of τ_t entropy amplification parameter extraction for a timeslot of History Summarised Vectors. 68

4.10 V-measure computed on cluster assignments without for course C1 using entropy amplification on measures. 72

4.11 V-measure computed on cluster assignments for course C1 using entropy amplification on a parameter. 73

4.12 V-measure computed on cluster assignments for course C1 using entropy amplification on all parameters. 73

4.13 V-measure computed on cluster assignments for course C1 using entropy amplification on all parameters and missing activities for d parameter. 74

4.14 V-measure computed on cluster assignments for course C2 without using entropy amplification on measures. 74

4.15 V-measure computed on cluster assignments for course C2 using entropy amplification on a parameter. 75

4.16 V-measure computed on cluster assignments for course C2 using entropy amplification on all parameters. 75

4.17 V-measure computed on cluster assignments for course C2 using entropy amplification on all parameters and missing activities for d parameter. 76

4.18 Impact of EA strategies on similarity measure calculation and Agglomerative clustering on course C1. 77

4.19 Impact of EA strategies on similarity measure calculation and Agglomerative clustering on course C2. 78

4.20 Homogeneity, Completeness and V-Measure of best performing measures and EA strategy using Agglomerative clustering on course C1. . 79

4.21 Homogeneity, Completeness and V-Measure of best performing measures and EA strategy using Spectral clustering on course C2. 79
4.22 Visual scheme of the Session Based Analysis approach. 83
4.23 Link creation in Activities co-occurrence multigraph. 84
4.24 Link creation in Sessions multigraph. 86
4.25 Link creation in Interactions graph. 87
4.26 Interactions graph of course C1 based on φ'_{sum} aggregation function. . 89
4.27 Interactions graph of course C2 based on φ'_{sum} aggregation function. . 90
4.28 Homogeneity, Completeness and V-Measure of modularity class assignments on course C1. 91
4.29 Homogeneity, Completeness and V-Measure of modularity class assignments on course C2. 92

5.1 Homogeneity, Completeness and V-score differences in percentage between History Based approach best measures and Session Based approach count aggregation on course C1. 95
5.2 Homogeneity, Completeness and V-score differences in percentage between History Based approach best measures and Session Based approach count aggregation on course C2. 96

B.1 Interactions graph of course C1 based on *sum* aggregation function. . 126
B.2 Interactions graph of course C1 based on *count* aggregation function. 127
B.3 Interactions graph of course C1 based on *min* aggregation function. . 128
B.4 Interactions graph of course C1 based on *max* aggregation function. . 129
B.5 Interactions graph of course C1 based on *mean* aggregation function. 130
B.6 Interactions graph of course C2 based on *sum* aggregation function. . 131
B.7 Interactions graph of course C2 based on *count* aggregation function. 132
B.8 Interactions graph of course C2 based on *min* aggregation function. . 133

B.9 Interactions graph of course C2 based on *max* aggregation function. . 134
B.10 Interactions graph of course C2 based on *mean* aggregation function. 135

Chapter 1

Scope, Challenges and Objectives

A massive amount of data, the big data, is continuously generated by the activities that people hold on online platforms, mobile systems and in general using and interacting with digital systems.

Many online platforms have evolved into social systems. In fact their focus on accumulating knowledge about their users' social network. This social information is usually not publicly accessible to everyone, only the owners of the platforms (Facebook, Twitter, Weibo, etc.) can analyse and get insights from the social network connections.

On the other hand, big data not directly related to social networks are generated in large quantities from systems that are not necessarily social systems. In the information society a whole lot of activities take place on systems that were not developed to consider, monitor and represent the social interactions between the people. Starting from such systems, we want to explore and see whether it is possible to derive social communities from the observation of people activities and their implicit interactions.

1.1 Discover and model social behaviour from activities

Starting from the observation of people's behaviour, we want to find if it can characterise the existence of social relationships among them. As people interact, individually or in groups, we want to elicit their social communities from the temporal and spatial co-occurrence of their activities.

Our starting point of view is that the systems we are going to analyse are composed of various components, some explicit/observable and other are implicit/hidden. The observable components are the people (e.g. system users), the environment (e.g. a digital system), and a specific set of actions that people can take on the environment. Examples of implicit components can be communication channels between people and the various social networks that connect them, which are hidden to the observers.

If people visit the same place, physical or virtual, in the same moment we can make the hypothesis that there is a social interaction. Depending on the context, the co-occurrence of two users actions can have different levels of significance as a clue of the existence of an interaction. The actions and activities co-occurring in crowded environments when many other people are interacting (e.g. people in a shopping mall at start of sales season) doesn't bring too much information for our purposes, since it could likely be a random coincidence. On the other hand, activities taken in isolated environments and unlikely times (e.g. students studying together at late night) are important to discover the social interactions as they are significant clues of underlying social communication channels and social connections.

The key component of our work is that we suppose that there are multiple, parallel, hidden communication channels and social networks where social interactions

take place. Our purpose is to elicit and model the properties of the social communities by only observing their effects. In other words co-occurence of actions of different users is considered as an emergent property of the inherent social network.

1.1.1 Co-occurrence of Co-located Activities

Time and space co-occurrence of activities can bring information about the interactions among people. The conveyed information is useful to discover the social connections between individuals and to reconstruct the underlying social communities.

The more frequently people will take activities at the same time (co-occurrence) and in the same place (co-location), the more we can assume and discover about their social interactions. The multiple occurrences of actions at the same time and in the same space will reinforce the assumption that the involved individuals are belonging to the same social communities.

Think about a group photo. People should gather in the same place and at the same time to shoot the picture. An observer looking at the group photo can infer the information that all the people represented in it had some sort of communication to arrange for the shot. By looking at the behaviour, i.e. how closely or loosely they gather, the observer can also infer the people's different social communities. The photo itself is the representation of social communications and interactions among people.

1.1.2 Time Co-occurrence

Event timing is important to analyse the interactions among people. When people's activities occur in the same moment, or within a short time interval from each other, we can infer with an high degree of confidence that the co-occurrence is

the epiphenomenon of the social interaction among them. The more the events are distant in time, the less information they convey about the strength of people relationships.

1.1.3 Space Co-location

Another essential factor to analyse is the location where interactions among people take place. In general, observable user activities can take place in physical or virtual spaces.

Physical spaces Activities in physical spaces are conveying information about the social communities if they take place in the same place or within an observable distance.

Take for example two individuals that attend a concert. They will probably have the same musical taste and we can suppose that they belong to the same group. Multiple occurrences of this kind of events will reinforce our supposition that there is a link between them.

The same behaviour can be observed for people that meet in shopping malls during sales season, students that attend the same school, people that work in the same place, and so on. Some co-location events convey more information than others, but all of them help to deepen the knowledge of the observed phenomenons.

Virtual spaces The other type of interactions that can be analysed are the one that happen in the virtual world. People daily interact by sending private and business emails, visiting social networking websites and have activities that involve the use of dedicated, web and eLearning systems. All the underlying actions can be tracked and the information can be used to elicit the knowledge about the social relationships between the individuals.

Web Web pages are the basic virtual environment for user activities. People view pages, read content, click on images and links, and in general interact with system and the objects contained in it. By doing so, the users leave tracks of their actions that are recorded and stored in the system logs.

Virtual Learning Environments Virtual Learning Environments (VLE) are used by teachers and instructors to present to students the content of their courses using web technologies. The courses can have various and different styles of presentation to students. Interaction with VLEs depend on course style:

- *MOOC/SPOC*: the course content is fully presented online. Students, using the VLE, access all the study material, follow lessons and maintain online the most part of their communications with instructors and peers. Only exceptionally the students know each other and communicate using other channels. The platform is their main meeting place.

- *Blended*: the blended learning courses mix face-to-face lessons with material that can be accessed online. Usually students-instructor communication is taking place in forums and mailing lists. On the other hand, student-student communication are likely taking place on hidden side channels. The communications through the platform are very limited and usually are directed from the instructor to the students.

In all the cases, the actions that take place online are tracked and stored in the system logs.

1.2 Modelling of Interactions

The interactions between people can be modelled in different ways, depending on the information that is available. In general, interactions can be categorised as explicit or implicit.

Modelling of explicit interactions require a deeper knowledge on the social links between people. The social network that is connecting the individual should be exposed to be examined. On the other hand, the modelling of implicit interactions is required when the social connections are unknown. This poses new challenges as the solution of many problems cannot be easily solved without determining the underlying social network.

1.2.1 Explicit

The social links between people can be derived by the simple observation of explicit interactions. In this case the environment exposes all the information about the social network that is connecting the different individuals.

The problems connected to the modelling of explicit interactions have been already widely explored in literature. Facebook posts and comments have been examined to understand the role of the social network in information diffusion [9], to elicit the semantic relatedness [17, 40], to determine the emotions conveyed by the posts [59, 39], and many other. Twitter social connections have been used to find the sentiment polarity of posts [96], to discriminate from true and fake followers [23], and so on. The common trait of all this works is that both the activities of the individuals and their social interactions are evident and observable.

1.2.2 Implicit

The modelling of implicit interactions is an emergent, and more challenging, research field. The domain where the activities take place is partially observable. In fact, the activities and actions of the individuals are known and observable but the social connections and interactions between people are hidden to the observer.

We can find different cases where the social interactions are hidden. In some case the system owners does not give access to it. For example Facebook, Twitter, and all the other social network platforms give partial or no information about the users social networks. In other cases the systems, physical or virtual, where these activities take place cannot record the social interactions between the users. This can be for the nature of the problem, e.g. a surveillance camera that records unknown people passing, or because the systems were not developed to take in account the social interaction information, e.g. web servers and applications. In this latter cases the only information available is the activity and action type, timing and sometimes the semantic associated to it.

1.3 Challenges and Thesis Objectives

The objective of this work is to find whether the co-occurrence of people's activities can convey information about their underlying social relationships and communications.

To characterise the social communities we use indirect information about the social links between people and we want to understand if it is possible to elicit the social communities by observing the activities of the individuals within system.

With this work we want to find an answer to the following main research questions:

- Is it possible to discover social communities basing only on the observation of people behaviours?
- Shall an external observer, not involved in the communications and looking only at the activities of the individuals, discover the underlying social communities?
- Are the social communities an emerging phenomenon of side channel communications and activities co-occurrence?

Potential applications of this work can fall in various areas. Business, marketing, public administration, including intelligence and military sectors, can benefit from the approaches introduced in this thesis. The social communities knowledge is elicited from people activities that are an emerging epiphenomenons of hidden, implicit information about side channel communications.

As a testbed for developing models and methods of community discovery as well as for experimental evaluation, we have focused on eLearning data. This kind of data is readily available and we can refer to real social communities are known.

The thesis is organised as follows. In Chapter 2 we review the social communities detection problem by introducing related works. In Chapter 3 we present various novel works whose common trait is the representation and analysis of behaviour. Chapter 4 we present the innovative social community detection approaches, their analysis in a specific domain and the approach-specific results. In Chapter 5 the experimental results from the various approaches are compared and evaluated. Finally, in Chapter 5 we summarise the thesis work and draw conclusions.

Chapter 2

Social Community Detection

Social communities elicitation is a particular aspect of the community detection problem. The latter has been explored from various points of view. The applications of community detection include the most different fields. From chemistry and pharmacology, where typical problems consists in finding groups of proteins and chemical compounds, to social networks, where the social relationships between the individual are analysed and exploited. As apparent from the previous examples, the definition of community in each field can vary.

Different approaches take in account the various characteristics of the considered domain. Some approaches exploit the network characteristics to elicit the communities, while others analyse the features of the elements to find the common characteristics in order to group them together.

2.1 Network-based communities

A great extent of the research efforts in community detection is devoted to the problems involving a network structure. Networks are, in general, dynamic and evolving and graphs are their direct mathematical representation. Graph theory measures and analysis methodologies hence can be used to study social, biological and information networks.

In social network analysis (SNA) these graph theory tools are used in conjunction with other visualisation and statistical tools. SNA focus is on the relationships between the individuals more than the single individual characteristics. This helps to find explanation of social behaviour using the network of relationships.

2.1.1 Graphs and Measures

Graph theory [16] provides a set of concepts and methodologies to analyse graphs.

A graph $G = (V, E)$ is defined as a collection of *vertices* V connected to each other by the *edges* E. If the edges have directionality, e.g. and connect a starting vertex to an end one, the the resulting graph is a *directed* graph. If the edges can be traversed in both ways the graph is *undirected*.

Ego Networks The ego network is defined as the subgraph of a social network formed by a vertex (*ego*), the other vertices directly connected to it (*alters*) and their connecting edges. Ego networks have been extensively studied in sociology and anthropology literature, that has defined their properties and characteristics.

Tie Strength One of the most studied properties is the *tie strength* [85] that defines the connection strength between two vertices and helps to categorise them vertices in *circles* [64]. In SNA the measures for tie strength can be based on var-

ious features. The frequency of interaction between the individuals, the individual perception of the strength of relationship, the cost of communication (e.g. distance) between the individuals, or a combination of the previous are all weighting functions helpful to evaluate the information diffusion on the network [4].

Homophily The homophily is the tendency to relate with people with similar characteristics. This leads to the formation of homogeneous groups (*clusters*) where it is easier to form connections and relationships. The ties within clusters are strong.

Heterophily There is heterophily when there are connections with characteristics that are not so similar as the ones in homogeneous groups. The heterophily is desirable in some contexts as it can bring innovation and idea generation. The ties are weak as the connections are between clusters.

Transitivity The transitivity property of ties. It is defined as in set theory:

$$if\,(aTb) \wedge (bTc) \Rightarrow (aTc) \tag{2.1.1}$$

with $a, b, c \in V$.

Bridging Bridges are vertices and edges that connect between groups. The bridge links facilitate communications between groups. The nodes connected by bridge are the contact points between communities and are fundamental for the social network cohesion.

Cliques In SNA groups of people that interact and share similar characteristics are defined as cliques. So the graph vertices in cliques verify the properties of homophily and transitivity. The ties within the cliques are usually strong.

Degree The degree of a vertex is the number of its connections with neighbours. In directed graph we can distinguish between *in-degree*, i.e. the number of links that lead into the vertex, and *out-degree*, i.e. the number of links the lead out if the vertex.

Paths A *path* between two nodes is a sequence of nodes that should be traversed to reach one another. The *shortest path* is the path that connects two nodes with minimum number of edges traversals. Shortest paths are desirable when looking for quick information spread on a network.

Assortativity The assortativity express the correlation between two vertices. This is usually done by computing Pearson's correlation coefficient (or Pearson's r) [78] using *degree-degree* correlation. This can be computed through the joint degree distribution function $P(k_1, k_2)$ that is the probability of finding an edge whose end points have degree k_1 and k_2 respectively. It has been found that social networks are in general assortative (positive correlation) since vertices of degree k tend to be connected to vertices of similar degree. In contrast, "technological networks" like the Internet are in general disassortative (i.e. node degrees are negatively correlated).

Modularity Modularity value Q is, up to a multiplicative constant, the number of edges falling within groups minus the expected number in an equivalent network with edges placed at random [74]. The value that measures the strength of division of a network into *modules*.

Given:

- $n = |V|$ number of vertices
- $m = |E|$ number of edges
- $u, v \in V$ vertices

- A adjacency matrix
- A_{uv} number of edges connecting u and v
- $k_v = deg(v)$ degree of node v
- $\delta(C_u, C_v)$ a function that yields one if $C_u = C_v$ (the two vertices are in the same community), zero otherwise.

we can define the modularity Q as:

$$Q = \frac{1}{2m} \sum_{uv} \left[A_{uv} - \frac{k_u k_v}{2m} \right] \delta(C_u, C_v) \tag{2.1.2}$$

The Q value range is in $[-1, 1]$. The higher is this value, the more distant the group assignment is from a random one. In general, the higher the value of Q of a partition, the crisper the community structure is.

2.1.2 Centrality Measures

Degree Centrality The degree centrality measures how much is connected the vertex. In SNA is used to find the influence and popularity of people. It helps to identify also which vertex is central with respect to spreading information and influence their immediate neighbours. Degree centrality for vertex v is defined as:

$$C_D(v) = deg(v) \tag{2.1.3}$$

Betweenness Centrality Betweenness centrality [42] measure gives information on which vertices are more likely to appear on communication paths within the graph. It is useful to identify cut points where the network can be partitioned. It is defined as

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2.1.4}$$

where σ_{st} is total number of shortest paths from vertex s to vertex t and $\sigma_{st}(v)$ is the number of those paths that pass through v vertex.

Closeness Centrality Closeness centrality [13] is a measure of the speed of information spread starting from one vertex to reach the others. In SNA it helps to identify which people can reach anyone else in the network. It is defined as follows:

$$C_C(v) = \frac{1}{\sum_u d(u,v)} \tag{2.1.5}$$

where $d(u,v)$ is the distance, in number of edge traversals, between u vertex and v vertex.

Eigenvector Centrality The eigenvector centrality value of a vertex is proportional to sum of the eigenvector centralities of its neighbours. This brings information that a vertex with high eigenvector centrality is connected with other nodes with high eigenvector centrality. This is similar to Google PageRank algorithm [77] where links from highly linked pages count more in searches. In SNA is useful to identify how well a person is connected to other well-connected people. The eigenvector centrality is defined as:

$$C_E(v) = x_v = \frac{1}{\lambda} \sum_u a_{u,v} x_u \tag{2.1.6}$$

where u is a neighbour of v, $A = (a_{v,t})$ is the adjacency matrix of the graph, and $\lambda \neq 0$ is a constant. Th eigenvector formula, in vector form, is:

$$Ax = \lambda x$$

The centrality vector x is the left-hand eigenvector of the adjacency matrix A associated with the eigenvalue λ. A good choice for λ is the largest eigenvalue in absolute value of matrix A as this choice will guarantee that if matrix A is irreducible, or equivalently if the graph is (strongly) connected, then the eigenvector solution x is both unique and positive.

2.1.3 Community Detection

Communities in graphs can be defined as groups of vertices that have strong ties between themselves and weak ties with the rest of the graph. Also called *clusters* or *modules*, they contain groups of vertices that exhibit common properties and roles within the graph structure.

Real networks are in general not homogeneous. The vertex degree distribution is broad with many low degree vertices and some with large degree. A great number of edges is found within some particular groups of vertices, and low numbers between these groups. This property of real networks is the so-called *community structure* [44].

Several methods have been proposed to uncover the communities present in a graph [37]. Community detection problem is a special case of the graph partitioning problems. These class of problems is NP-hard as there is no known algorithm that solves the problem in polynomial time [43]. Several heuristic approximated solutions have been proposed.

In community detection an added difficulty is that there is not a single accepted rigorous definition of measure for the quality of communities. One of the most commonly used measure is modularity as defined in paragraph 2.1.1.

Community detection algorithms that use graph information and measures can be grouped in several different classes. In literature various solutions include graph partitioning, divisive algorithms, modularity maximization, statistical inference and

clique-based methods.

Graph Partitioning This is one of the traditional methods for community detection. It generally consists in dividing the vertices in G in a predefined number of communities k. The problem is, in general, NP-Hard but various approximated solution have been developed.

The *Minimum Cut* algorithm is a partitioning algorithm that finds the k communities with the minimal number of edges connecting the the communities. The number of edges falling between communities is the *cut size*. Usually the approximated algorithms proceed by bisecting the graph using a various heuristics and repeating the procedure until the number k of communities is reached [55, 95]. Other methods make use of the Ford and Fulkerson max-flow min-cut theorem [36] to find efficient approaches to determine minimal cuts in graphs with capacity on the edges [22].

The main feature, and main drawback of this method, is that the number k of communities to be found must be specified. In fact, the minimum cut of a graph is always the one with a single community e.g. there are zero edges between communities.

Divisive Algorithms Divisive algorithms work by identifying and removing the edges that connect communities. By doing so the communities will become disconnected from each other. The divisive algorithms usually differ by the measure that is computed to discriminate the edges to be removed.

One popular divisive algorithm is the one developed by Girvan and Newman [44, 73] in their works exploring social and biological networks. The edge to be removed are selected by using their *edge betweenness centrality*. This measure is an extension to edges of the definition of betweenness centrality for vertices reported in 2.1.4.

The algorithm steps can are :

1. compute betweenness centrality for all edges in graph and modularity Q
2. remove one of the edge with largest centrality
3. recalculation betweenness centrality on the resulting graph and modularity Q
4. repeat from step 2 until there are no edges to remove or modularity Q reach the desired threshold

The result of the algorithm is a dendogram, with individual nodes as leaves. The dendogram represents the top-down split of the graph obtained by the edges removal. This algorithm is quite slow and its runtime is acceptable for graphs with a number of vertices $n = |V| \approx 10000$.

Modularity Maximization Modularity value Q, as defined in equation 2.1.2, has originally been introduced as stopping criterion for the divisive algorithm by Girvan and Newman [74]. This value has been widely accepted as the best to qualitatively evaluate communities in graphs.

Modularity is value that measures the strength of division of a network into modules. Having high modularity groupings in a network is represented by having dense connections among the nodes within modules and sparse connections between nodes in different modules.

Modularity maximization methods try to find the grouping that has the highest modularity value Q within a network. The naive method is to start with one node and proceed by adding a connected node to the group until the modularity score increases. This problem has been proved to be NP-Hard [18] but heuristic algorithms have been developed. Greedy [15], simulated annealing [46], evolutionary computation and other heuristic methods [22] have been used to solve the modularity maximization optimisation problem.

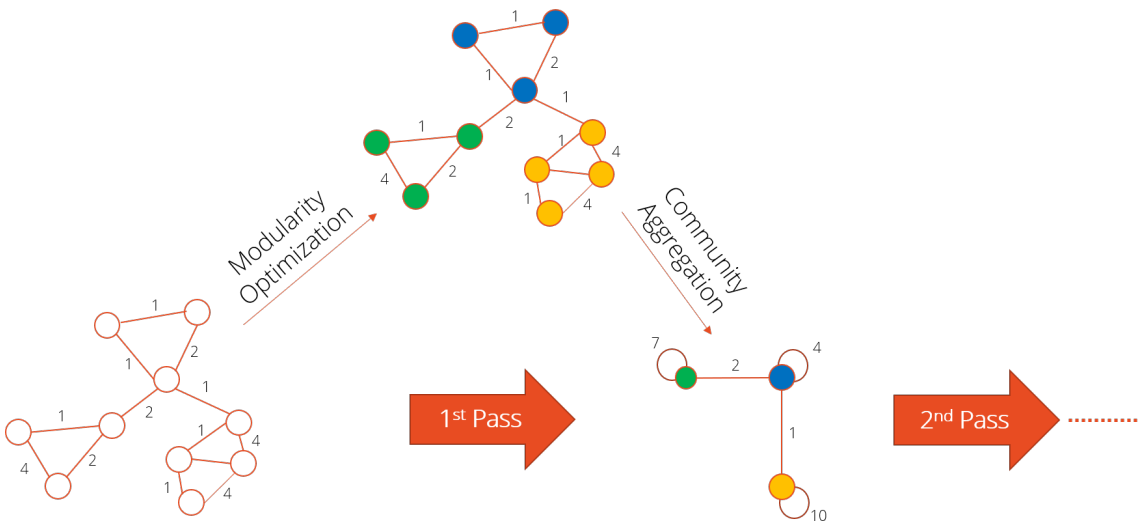

Figure 2.1: One step of Louvain algorithm for modularity maximization.

One of the most efficient approaches to modularity maximization is the so-called *Louvain Method*. The algorithm introduced by Blondel et al. in "Fast unfolding of communities in large networks" [15] is particularly efficient even for very large size networks, with nodes in the order of hundreds of millions.

The Louvain method examines a weighted graph. At start each vertex is assigned to a different community. In first phase a community aggregation takes place. For each vertex, the considered vertex will join one of its neighbours' community if there is a gain in modularity. This procedure is repeated until there is no gain in modularity. The second phase consist in building a new graph composed of super-vertices that are the communities found in the first phase. The edges between super-vertices are weighted as sum of the weight of the links between nodes in the corresponding two communities. The process is repeated until there are no changes in the network structure and the maximum modularity is achieved. In Fig. 2.1 is shown one step of the modularity maximization algorithm. The result of the algorithm is the assignment to each vertex of a modularity class value.

Statistical Inference Methods using statistical inference [67] try to generate the network structure using a model that exploits the features of the input data.

The general approach is to use a stochastic blockmodel [50] and its variants to produce graphs that contain communities of nodes. The original graph is decomposed in communities of vertices that share common properties. Edges within the communities are more dense than the edges connecting one community to the other.

Various statistical approaches use belief propagation [8, 47, 83] that make use of Bayes theorem to infer the communities in graphs. In fact, the community detection problem can be seen as a specific example of statistical inference. The graph structure, represented by the adjacency matrix, is the evidence of the problem. The missing information to be inferred by the model is represented by the model's parameters and the vertex grouping information.

Furthermore, Markov chain Monte Carlo (MCMC) statistical inference approaches [79] have been introduced to produce greedy agglomerative heuristic for community detection in graphs.

Clique-based Methods Cliques are groups of two or more individuals who share similar characteristics and are connected one to each other forming a network. Cliques can overlap each other which is a desirable characteristic for many social network tasks. Detection methods can find cliques of fixed size or with the maximal number of elements. The former can use percolation methods to determine the node cliques [61], while the classical algorithm for the latter is the Bron–Kerbosch algorithm [19].

2.2 Feature-based communities

In social communities each individual is characterised by unique features. The habits and profiles of users, but also their behaviour and interactions, can be recorded and recognised to define the user's profiles. The *clustering* algorithms take in account the different features of the individuals and find the ones that have more in common to group them together. Each *cluster* will contain the individuals that are more similar in the specific feature space given a specific comparison measure. Clustering of elements that share similar features and interests is widely used in different fields. From network optimisation to recommender systems and marketing the clustering algorithms are used to find groups of similar elements. The clustering of geographic location of web requests help to improve the performance of services on the Internet, by exploiting these features the clusters of web clients can be served by a dedicated regional server [60]. Identifying clusters of customers with similar interests is essential for online retailers. Recommendation systems [58], help to guide the customer choices when browsing online catalogues. This has the double effect of directing the customers' choices and increase the sales for business owners. In customer relationship management (CRM) systems the goal is to find users to submit marketing campaigns and evaluate their effects. The customers for marketing campaigns should share common features and their communities are based on similarity of features and interactions. This helps business to refine the marketing strategies to match customer expectations [103] as well as to evaluate the customer satisfaction [6].

Using the clustering approach the result is a class assignment for each of the elements. All the information to reconstruct a network is already inside the clustering assignments. As a matter of fact, it's easy to add links that connect, with different weights, all the elements intra and extra cluster. The weight will be representative

of the measure used by the clustering algorithm to compare the elements.

2.2.1 Clustering Algorithms

Various approaches and clustering techniques have been investigated in order to best adapt to the given domains. The various methodologies can be broadly classified in four categories, i.e. hierarchical, statistical, partitional and spectral clustering. All the clustering algorithms, except the statistical ones, use similarity or dissimilarity measures to compare the elements.

Agglomerative Agglomerative clustering is a technique that falls into the hierarchical clustering algorithms' category [53]. This technique uses a bottom-up approach to build nested clusters by repeated merges of elements. Each element starts in its own cluster and then is linked to the clusters that minimises the sum of squared differences between all clusters. The elements are usually compared using a similarity measure.

Divisive Opposite from the approach of agglomerative clustering, the divisive algorithms are in the hierarchical clustering family [53] and use a top-down approach. At start all the elements are in the same cluster. Iteratively the clusters are separated until every object is in its own cluster. The elements are usually compared using a dissimilarity measure.

Statistical Statistical clustering algorithms are statistical generative models that represent the process of generating information. These models include a set of statistical assumptions about the generation of sample data that is representative of a larger population.

The EM (Expectation-Maximization) [29] is an iterative algorithm for finding the maximum likelihood of estimated parameters of a statistical model. The model

depends on latent variables, e.g. from equations which cannot be resolved directly, or from data which were not observed, where the existence of such data can be assumed true. EM iteration rotates an expectation step (E), which iteratively calculates the expected likelihood on the current estimate of parameters, and a maximization step (M), that estimates which parameters maximise the expected likelihood, calculated in the E step. The algorithm terminates when updating the parameters does not increase the likelihood.

Partitional Partitional clustering methods try to find a specific number k of clusters in a set of elements. These elements are immersed in a feature space where a distance measure can be defined between pairs of elements. The distance is a measure of dissimilarity between elements basing on the feature's value. The goal is to separate the elements in k clusters in a way that maximise (minimise) a given cost function based on distances. The distances can take in account the differences between existing elements and/or from *virtual* elements such as the cluster *centroids*.

K-Means [68] is a clustering algorithm to partition n elements into k clusters containing the closest mean differences from *centroids*. The problem is computationally difficult (NP-hard), but algorithm exist to solve efficiently the problem by using heuristics to converge quickly towards a local optimum starting from random *centroids* [65].

Spectral Spectral clustering techniques makes use of the *spectrum*, i.e. the set of *eigenvalues*, of a similarity matrix to partition the initial set of elements in different clusters. The similarity matrix is constructed using a pairwise similarity function. The eigenvalues are necessary to perform a dimensionality reduction before clustering in lower dimensions. The techniques are general as the low dimension similarity matrix can be used in combination with almost all the other clustering methods. Usually *K-Means* is used to detect and assign the cluster labelling to elements.

The clustering assignment using the spectral information renders more evident the cluster properties of the initial elements.

Different spectral clustering make use of the eigenvectors of different matrices. Initial works used the eigenvectors of the adjacency matrix [33] or the eigenvectors of the Laplacian matrix [35]. The latter approach is the most used matrix representation for spectral clustering techniques. Various optimisation have been proposed for the computation of the Laplacian matrix and its eigenvectors [92, 75]

2.3 Behavioural-based communities

The discovery of behaviour-based communities is based on the assumption that if two people have the same dynamic behaviour they are similar. The behaviour of a person can be described by the sequences of actions they take during time. These *user histories* represent dynamic features that can be used to model the users' behaviour.

The user modelling is similar to what introduced in *feature-based communities* 2.2, where clustering is applied using features that can also summarise action sequences (e.g. frequency rates, connection time, etc.) but very seldom the action histories are used as sequences.

In our approach the histories are features that describe dynamic timed sequences expressing the user's behaviour. Behaviour similarity (i.e. history similarity) is computed by looking at temporal correlation of actions.

Social communities can be elicited by clustering the elements using similarities extracted from user histories. The elements' similarities can be found by:

- using vector similarity measures;
- training a neural network in order to obtain user behaviour embedding.

Recommender systems are a typical application of history similarity. The histories of purchases are modelled and compared to find similarities between users and elicit social communities, such as customer segments, for marketing purposes.

2.3.1 Vector Similarities

The user histories can be represented using vectors, so we can find the correlations between the elements using vector correlation based similarity measures. Communities are elicited by using history similarity as clustering similarity measure.

Each vector element will represent a time slot containing a boolean value representing the presence or absence of a given feature F. The vectors can be compared using binary similarity measures, that can emphasise different aspect of the relationship.

Measure Parameters To apply the various binary similarity measures, the vectors are compared pairwise to compute the measure's parameters. The parameters' contingency table is shown in Table 2.1 and exemplified in Figure 2.2. According to this table the value of each parameter is computed counting all the occurrences of the corresponding condition on the boolean vector pair. The parameters are computed according to the following formulas:

$$a = \sum_{F \in Features} (Vector_1[F] * Vector_2[F]) \tag{2.3.7}$$

$$b = \sum_{F \in Features} (Vector_1[F] + Vector_2[F]) \tag{2.3.8}$$

$$c = \sum_{F \in Features} (Vector_1[F] + Vector_2[F]) \tag{2.3.9}$$

$$d = \sum_{F \in Features} (1 - Vector_1[F]) * (1 - Vector_2[F]) \tag{2.3.10}$$

Cooccurence This simple statistic measure takes in account only of the presence of features in both sets.

$$Cooccurrence = a \tag{2.3.11}$$

Table 2.1: Contingency table for pairwise vector comparison

	Vector 1 Element Feature F present	Vector 2 Element Feature F absent
Vector 1 Element Feature F present	a	b
Vector 2 Element Feature F absent	c	d

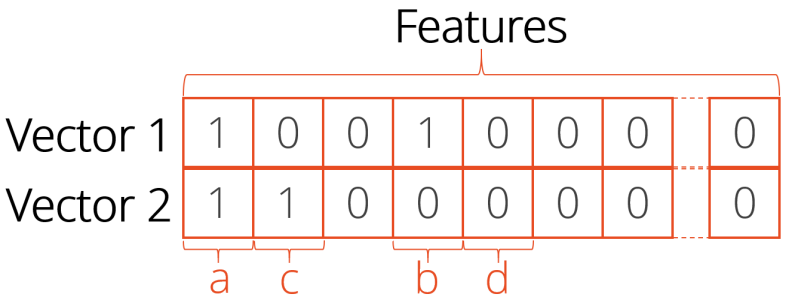

Figure 2.2: Pairwise vectors comparison to compute similarity parameters.

Jaccard similarity measure Introduced by Paul Jaccard at the start of 20^{th} century [52], this measure is also known as the similarity ratio. It does not use the features' absence parameter in the computation.

$$Jaccard = \frac{a}{a+b+c} \tag{2.3.12}$$

Russell and Rao (RR) similarity measure This similarity measure, introduced in 1940 by Russell and Rao [88], considers at the denominator the information about the absence of a features in the two sets.

$$RR = \frac{a}{a+b+c+d} \tag{2.3.13}$$

Simple Matching (SM) similarity measure Known also as the Rand Index [82], this measure is the ratio of the number of matches to the total number of characteristics.

$$SM = \frac{a+d}{a+b+c+d} \tag{2.3.14}$$

Dice similarity measure Dice similarity measure [30], also known as Czekanowski or Sorenson, is similar to the Jaccard index, but give more emphasis to the presence of common features between the compared sets.

$$Dice = \frac{2a}{2a+b+c} \tag{2.3.15}$$

Sokal and Sneath (SS1) similarity measure n. 1 In their work, Sokal and Sneath [93], presented various similarity measures. This measure enhances the concurrent presence and absence of features in the two sets.

$$SS1 = \frac{2\,(a+d)}{2\,(a+d)+b+c+d} \tag{2.3.16}$$

Sokal and Sneath (SS2) similarity measure n. 2 This measure enhances the disjoint presence and absence of features in the two sets.

$$SS2 = \frac{a}{a+2\,(b+c)} \tag{2.3.17}$$

Sokal and Sneath (SS3) similarity measure n. 3 This measure balances the concurrent presence and absence of features in the two sets with the disjoint feature's parameters.

$$SS3 = \frac{a+d}{b+c} \tag{2.3.18}$$

Sokal and Sneath (SS4) similarity measure n. 4 This measure yields the conditional probability that a characteristic of one vector is in the same state (presence or absence) as the characteristic of the other vector.

$$SS4 = \frac{\frac{a}{a+b}+\frac{a}{a+c}+\frac{d}{b+d}+\frac{d}{c+d}}{4} \tag{2.3.19}$$

Ochiai similarity measure Also known as Fowlkes-Mallows index [38], the Ochiai similarity measure [76] is the binary form of the cosine vector similarity.

$$Ochiai = \sqrt{\frac{a}{a+b}\cdot\frac{a}{a+c}} \tag{2.3.20}$$

Yule Y coefficient of colligation Yules Y coefficient of colligation [106] is a measure of association between two binary variables.

$$YuleY = \frac{\sqrt{ad} - \sqrt{bc}}{\sqrt{ad} + \sqrt{bc}} \tag{2.3.21}$$

2.3.2 Neural Networks

Neural Networks have been successfully applied to various fields of research. The ability to classify and find correlations between elements makes them suitable for various tasks. A classical and successful field of application is to natural language processing (NLP) tasks. A very popular model architecture for estimating neural network language model (NNLM) have been proposed in [14] using a feedforward neural network to learn together the representation of word vectors and the statistical language model. Later investigation brought to the separation of the NNLM tasks from the word embedding learning [71]. This latter task has been found to be useful to simplify the NLP tasks and have been trained using various corpora [97, 51]. Although these word representations brought improvements in the accuracy, the architectures were computationally expensive to train.

The state of the art for word embeddings has been reached with the continuous bag of words (*CBOW*) and *skip-gram* models introduced by Mikolov et al. [70]. This approaches start by the observation that two words are semantically similar if a correlation can be found in the same context. These model overcome the training limitations of the previous architectures, achieve excellent semantic and syntactic accuracy while using a simple architecture.

The two models have been implemented in the *Word2vec* software. When using the CBOW approach, the model predicts the current word from a window of surrounding context words. The order of context words does not affect the prediction. When using the skip-gram approach, the model uses the current word to predict the surrounding window of context words. The nearby context words are given more importance than distant context words.

The word embeddings produced by Word2vec have been later used for many different tasks. This process can be seen as an alternative way to find features and similarities to feed convolutional neutral networks (CNN), clustering and other classification algorithms. Some examples include text classification using CNNs [62], recommender systems using collaborative filtering [12], opinion mining using deep CNN [80], healthcare analytics using deep models [26], and many more.

2.4 Information Diffusion-based communities

Information diffusion is a phenomenon that take place in networks. Under this definition we can include various and different processes. The word of mouth effects in marketing [54], the spread of information of opinion and news [2], the spread of infectious diseases [3] are some of the application fields were information diffusion has been studied.

To study the information diffusion first we need to define what is the contagion (i.e. the message, idea, disease, virus, etc.) and find a way to track it when it diffuses through the network. Moreover, the network itself is usually unknown and we can only observe it when a node gets infected. There is no information from where the infection came from.

One successful approach, using generative probabilistic model, has been developed by Gomez-Rodriguez, Leskovec and Krause [45, 86] and implemented in the *NetInf* program. In their approach they trace the paths of diffusion through networks and simultaneously infer the network structure. This approach needs sequences of timed events (*cascades*) as input, where the nodes of the network and the time of infection are known. Using this approach the role in spreading the information and the range of influence of the nodes can be inferred.

This approach has a double nature:

- starting from sequences of timed events it can elicit the underlying network;
- starting from a network can bring the expected diffusion of information.

The *NetInf* implementation is an efficient approximation of the general problem, that scales well on very large real datasets.

Social behaviour dynamics, using information diffusion and probabilistic framework, have been further explored using maximum likelihood estimation [105]. Infor-

mation diffusion have been also applied to detect communities without prior knowledge of the underlying network [81]. Expectation maximization based approaches have been explored to infer the spread of information on social networks [10, 11]. Other approaches, different from probabilistic models, have been explored to model the information diffusion using recurrent neural networks (RNN) [100].

Chapter 3

Investigating Behaviour and Communities

The investigation for this thesis has started by focusing on the analysis of explicit, direct, interactions among people, in particular on modelling the emerging behaviour, i.e. the actions that individuals take on the system, in a domain where the communities of users are known. First we have investigated the discovery of the contextual importance and emotional information conveyed by short text, next we have explored models of collective behaviour of agents in the *tragedy of the commons* problem. In addition we have developed a tool to investigate the behaviour of users from the quantitative point of view.

In this chapter we will review the before mentioned works, while in next chapter we will introduce innovative approaches to model the social behaviour of individuals and discover social communities where the social relationships and communications channels among the individuals are hidden to the observer. Session Based Analysis and History Binary Correlation Analysis approaches, that fall in the Network and Behaviour community detection techniques introduced respectively in Sections 2.1 and 2.3, exploit different aspects of the activities that people take within systems.

3.1 Behaviour from Explicit Interactions

In the works we present in this section we analyse the explicit interactions among the individuals.The assumption is that the social connections and relationships among the individuals are completely known. The general focus of all these works is on processing the interactions' content assess the properties related to explicit behaviour.

The aim of the two main directions we have explored is on assessing the quality of social interactions and evaluating their emotional content in the context of Facebook domain.

These works are based on path-based similarity measures, where the similarity among words can be measured by exploring a hierarchical taxonomy (e.g. WordNet [72], SnowMed, NANDA, etc.), thus measuring the path length that connects them in such a taxonomy, or collaborative web-based such as Wikipedia and Linked Data. WordNet distance [20], Leacock-Chodorow similarity [63] and Wu-Palmer similarity [104] are examples of measures that exploits the hierarchy of an ontological knowledge base. Examples of web-based semantic measures based Wikipedia are Wikipedia Link Vector Model (WLVM) [102], Heuristic Semantic Walk [41] and Linked Data [7].

3.1.1 Qualitative classification of interactions

The quality of interactions between users can be determined by the content of posts and comments on a social network. Automatic categorisation of the contextual information between posts and its comments brings to a classification of users' behaviour [40]. The goal of this work is to find the correlation and contextualisation of comments in respect to the original Facebook posts.

Facebook comments and shared posts often convey human biases, which play a pivotal role in information spreading and content consumption, where short infor-

mation can be quickly consumed, and later ruminated. Such bias is nevertheless at the basis of human-generated content, and being able to extract contexts that does not amplify but represent such a bias can be relevant to data mining and artificial intelligence, because it is what shapes the opinion of users through social media.

Starting from the observation that a separation in topic clusters, i.e. sub-contexts, spontaneously occur if evaluated by human common sense, especially in particular domains e.g. politics, technology, this work introduces a process for automated context extraction by means of a class of path-based semantic similarity measures which, using third party knowledge e.g. WordNet, Wikipedia, can create a bag of words relating to relevant concepts present in Facebook comments to topic-related posts, thus reflecting the collective knowledge of a community of users.

Experimental evidence shows that clustering on frequency-based bag of words (see Figure 3.1) can identify the main context words contained in Facebook comments identifiable by human common sense. Group similarity measures are also of great interest for many application domains, since they can be used to evaluate similarity of objects in term of the similarity of the associated sets, can then be calculated on the extracted context words to reflect the collective notion of semantic similarity, providing additional insights on which to reason, e.g. in terms of cognitive factors and behavioural patterns.

In this research work we introduced a method to investigate and identify the main context words obtained from Facebook posts and related user comments. The method is based on Natural Language Processing Part of Speech nouns extraction from sentences, similarity measurement using WordNet ontology, and clustering techniques. Results show that clustering on frequency-based bag of words gives interesting results in the identification of topic contained in Facebook and it is more similar to human judgement than low level features comparison.

Figure 3.1: Context tag cloud sample of a Facebook post and its comments.

3.1.2 Sentiment polarity and emotion from interactions

Another qualitative element of the relationship between users includes the sentiment and emotion conveyed by the posts. The purpose of this research is to analyse the content of Facebook posts and their comments to classify the users behaviour in respect to sentiment and emotion [39].

Each word in a short text can potentially convey emotional meaning [94]. Every word, even those apparently neutral, can evoke pleasant or painful experiences, more or less abstract, due to their semantic relation with emotional concepts or categories. Some of such terms or concepts have also an emotional meaning, with respect to an individual story, while for many others the affective power is part of the collective imagination. For instance, words such as “war”, “attack”, “bomb” can convey meanings related to basic negative emotions, e.g. fear and anger.

In a communication environment such as Facebook, where people like to express strong concepts and opinions with less filtering than in other social media and everybody feels free to say anything with weak communication filters, articulated comments can be expressed by users aggregated in homophily communities [9].

Starting from the observation that users can become polarised comment after comment, where they comment the same post expressing similar concepts or with respect to a similar level of abstraction, besides preferential attachment users often comment the main topic using similar use cases. For example, in the domain of information security, where a previous comment search for a solution on how to solve a problem, other users will probably seek help and ask questions about the same problem, because they trust the source and they think they are facing a similar problem.

In this research we introduced a novel methodology to separate clusters of emotion abstraction in Facebook comments, where concepts underlying the content of comments are grouped, by similarity, with the emotional concepts related to basic emotions models and then linked to the concept of abstraction in an external ontology annotated by experts. Figure 3.2 is represented the ontology based emotion similarity computation using the Ekman model for exemplification. The similarity is computed as the shortest path between the examined word and the emotion terms included in the emotion model.

In this process, Natural Language Processing techniques are used to extract meaningful terms from the short and articulated text of Facebook comments, path-based similarity measures are calculated on a taxonomy of meanings. WordNet [72] is used in our priming experimentation as universal knowledge base, but we provide a description of some other useful state of the art methods on different knowledge bases and measures.

In this work, preprocessing of Facebook posts and comments, extracted from

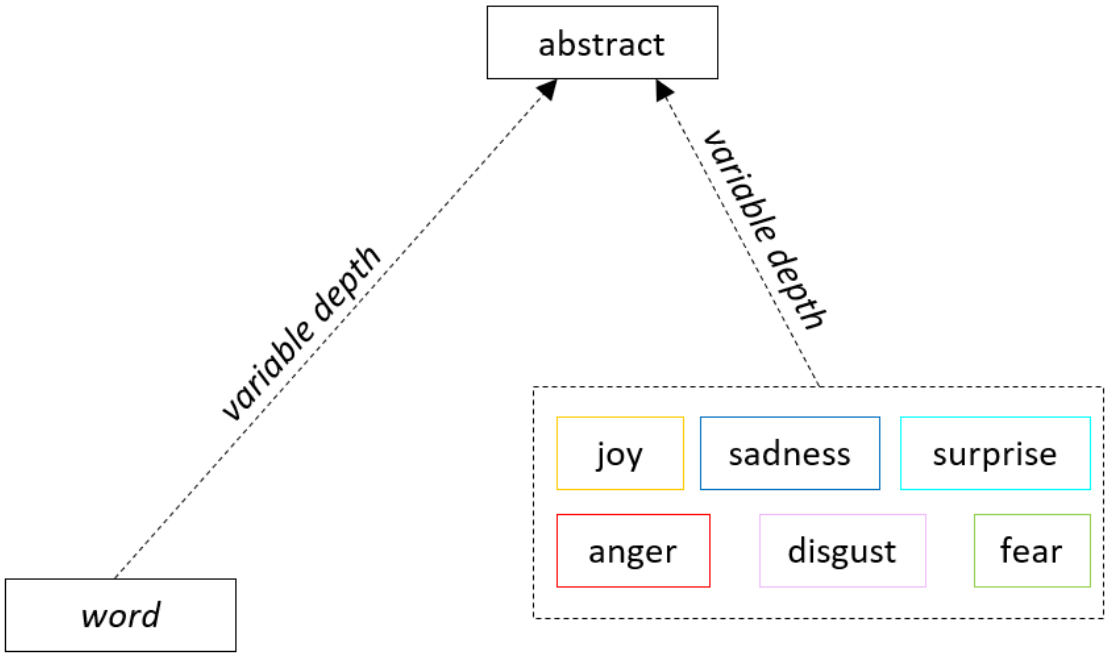

Figure 3.2: Ontology based emotion similarity.

Facebook graph, is applied using basic Natural Language Processing techniques [101]. The resulting bag of words is processed using simple techniques, where advanced methods of sentiment analysis are not needed, to obtain a set of candidate topics that are used to find the level of emotion abstraction. Measures of semantic similarity path-based measure such WordNet distance, Leacock-Chodorow similarity [63], and Wu-Palmer similarity [104] are calculated, by means of the hierarchy of WordNet as an ontological knowledge base. Experiments have been implemented using path-based distances between Facebook comment-extracted terms and basic emotions, e.g. in the Ekman model. Sentiment polarity analysis of words and comments is then added to the features to study. Preprocessed information, augmented with the similarity and polarity values, is then submitted to a clustering algorithm to obtain clusters of words, conveying the emotion valence. A representation of the approach applied to a Facebook post and its comments is shown in Figure 3.3.

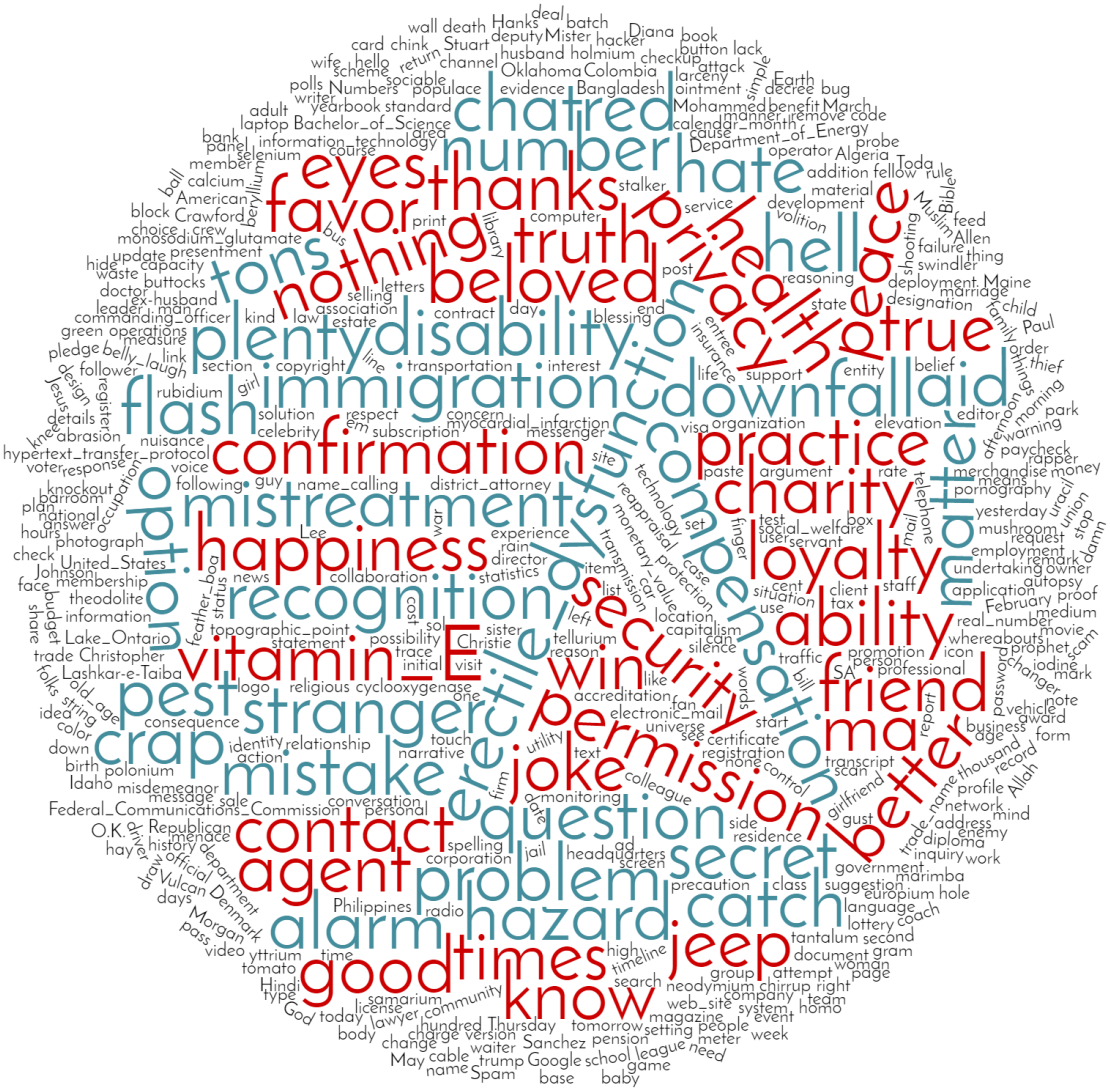

Figure 3.3: Sentiment polarisation tag cloud sample of a Facebook post and its comments using the path-based emotion abstraction approach.

Experimental results show that the proposed approach gives interesting results in the identification of emotional abstraction of Facebook comments, thus allowing to discriminate the users' behaviour in respect to sentiment and emotion.

3.2 Mass Behaviour Modelling

Emerging behaviour from the interactions of a great number of agents. To solve a classical problem, the *"Tragedy of the Commons"* (TOC) [66, 48, 31], we modelled the social interaction among agents to access a shared resource. Analysis of the impact of the social interactions helps to understand the influence on the problem solution.

The tragedy of the commons is a problem in which the sustainability of the society (a group of agents) reduces due to self-interested individual agents using a shared resource (a commons). This problem first appeared in the seminal paper of Hardin in 1968 [48]. Many areas of interest to society like climate change, fisheries management, and preservation of rain forests exhibit this phenomenon [98].

Researchers in the area of Distributed Artificial Intelligence (DAI) and Multi-agent systems [89, 32, 56, 21, 91] have also addressed the TOC problem. Different aspects of the optimisation problem have been explored; from the optimal resource utilisation by the individuals of the society [89], where the agents have only local information, to the performance of the society is studied when aspiration levels are associated with an individual [91]. An aspiration level corresponds to the satisfying return for an individual. Such an aspiration level is adjusted based on past experience.

Any attempt to avoid the tragedy of the commons should incorporate in to the decision making process of an agent the following: the individual gains as well as the social welfare. However these two aspects often conflict. This issue has been addressed in the work of Hogg and Jennings [49] in the context of designing socially intelligent agents, where a framework is proposed for making socially acceptable decisions.

Consider a society where a public good is available for free (or very little cost)

to the members of the society. If there is no law associated for the utilisation of the public good, an individual of the society would like to act in a manner that maximises its utility of the public good. From an individual perspective this is the best decision. However if all individuals act in the like manner, the public good would soon get depleted due to the synergistic behaviour and so the society collapses. Thus laws are necessary for the proper functioning of a society. When there is a law in effect it entails a member to abide by it. TOC is concerned with the situation when there is no such formal law or rule. This is where the behaviour of an individual comes in to effect that should consider its utility from the public good and the depletion rate of the public good.

In this work we considered socially motivated agents. The agents make decisions that consider the welfare of the society. This helps the society to survive for a longer period of time compared to the situation when the agents would have acted for their individual gains only.

The approach we introduced is based on the study of different agent behaviours. First self-interested agents, i.e. the agents interested in their individual gains only, have been considered. Then we assumed that the agents make decisions based on the resource availability, individual gains or combination of both. The second class of agent's behaviour takes into account the welfare of the society, then the society survives for a longer time compared to that in previous case. In the third type of behaviour considered, we defined a measure of greediness and the agents make decision based on this parameter. A social cooperative behaviour (SCB) algorithm has been proposed, that converges to equilibrium quickly and with no potential danger of over utilisation of common resources.

Experimental results shows that the algorithm outperforms state of the art existing approach.

3.3 Behaviour Quantitative Analysis Tool

The actions and activities that the students take on the Virtual Learning Environments (VLE) have been modelled and analysed to discover the quantitative effects of monitored activities. The analysis is useful for the instructors to evaluate their learning designs as well as for the students to have better understanding of their learning style.

To model the behaviour of students in VLE we developed a tool for learning analytic. The tool displays quantitative information about users' online activities and usage of learning objects. The resulting analysis is then presented to the teacher by using various static and dynamic VLE's interface morphing metaphors .

This helps instructors to verify the effectiveness of their learning designs (LD). Acceptance of content and students' engagement are interesting indicators for instructors. These factors can be represented by analysing the usage over the time of the learning objects with respect to single student, group of students and/or entire classes. The data to be analysed may include the temporal and quantitative factors of activities such as the time and number of accesses of learning objects, assignment or projects submissions, etc. Usually these tasks are accomplished through the visualisation of log reports in textual, tabular or graphical formats.

Direct consultation of activity logs is usually included in the VLE's interface. This enables teachers to visualise, sort and filter the information contained in the general logs. The parameters that the educators can adjust to filter and aggregate the information are usually very limited. Role aggregation and time window selection are desired variables needed to fine-tune the analysis task. When those advanced options are present, the selection of choices is typically very limited. Resulting information is presented to the user as simple textual or tabular visualisations, difficult to read and to analyse.

Learning dashboards [90] help to present Key Performance Indicators (KPI) [99] through visual elements in the VLE's control panel. Data is visualised in graphical, synthetic, form using charts to display the KPIs to the instructor [57]. An extensive review and evaluation of impact of the state of the art learning dashboards can be found in the work of Verbert et al. [99].

Tools like Moodle Activity Viewer (MAV) [27], a browser plugin, and Block Heatmap Moodle plugin [28] provide basic functionality to analyse the learning objects' accesses. The information is visualised using color grading overimposed to the VLE interface with static point of view.

The main drawback of the currently available tools based on log report analysis is that skilled instructors/administrators are required to understand and use KPIs to assess the learning process effectiveness. In order to use typical analytic functions, such as predicting student outcome [5], evaluating the risk of students to drop out their study track [34], determining courses utilisation levels and quality for informing the LD process, the teachers usually need specific statistics and data mining knowledge to give a proper interpretation to textual data and dashboard charts.

From the educator point of view, all of these tools are often difficult to use or require an effort to learn new concepts and methodologies. As noted by Chatti et al. "currently many of the systems are data rich, but information poor" [25] as the methods previously introduced lack of information for educators whose background is from fields that does not require knowledge of statistical and data mining concepts needed to analyse the information presented in log reports and KPIs.

The aim of the innovative tool we introduced is to allow instructors to carry out learners' continuous monitoring by exploiting the richness of activity logs and by reaching the goals of integrating informativeness, immediacy and ease-of-use for unskilled instructors. We introduced a tool that allows them to fine-tune the

information visualisation (i.e. using aggregation functions, temporal dynamics, etc.) to obtain indirect feedback on their teaching material and LD.

The accent of this approach is on the learning objects usage analysis to model the mass behaviour in respect to the learning objects. In this approach to interaction analysis we take in account the students mass as a whole. While it's possible to fine grain the detail of analysis, the individuals and their social relationships remain in the background.

We introduced a class of dynamic visual interfaces whose characteristic is to act on a learning management system's interface to modify the learning objects appearance. The general idea is to modify elements of the interface in order to express some key dimension of the data and show them through an in-place dynamic dimensional morphing visualisation. This form of visualisation combines the completeness of information with the efficacy of representation largely improving the usability of the eLearning platform monitoring.

In order to facilitate the visual monitoring of the overall impact of the course resources on students, we compute quantities extracted from the users logs of the available learning objects and we use them as visualisation parameters. These quantities are used to render visually the relative differences and relationships of the examined elements among the users.

Temporal information is expression of the chronological access history to the elements, within the interval of analysis. To render properly this kind of information we need two time-related values associated with the learning objects: the date of access and a reference date. While the former is extracted by access logs, the latter is provided by the user of the monitoring system to set the point of view or temporal landscape.

Temporal proximity is then rendered with the hot/cold metaphor, where hot topic refers to resources more recently last accessed which are presented in warm

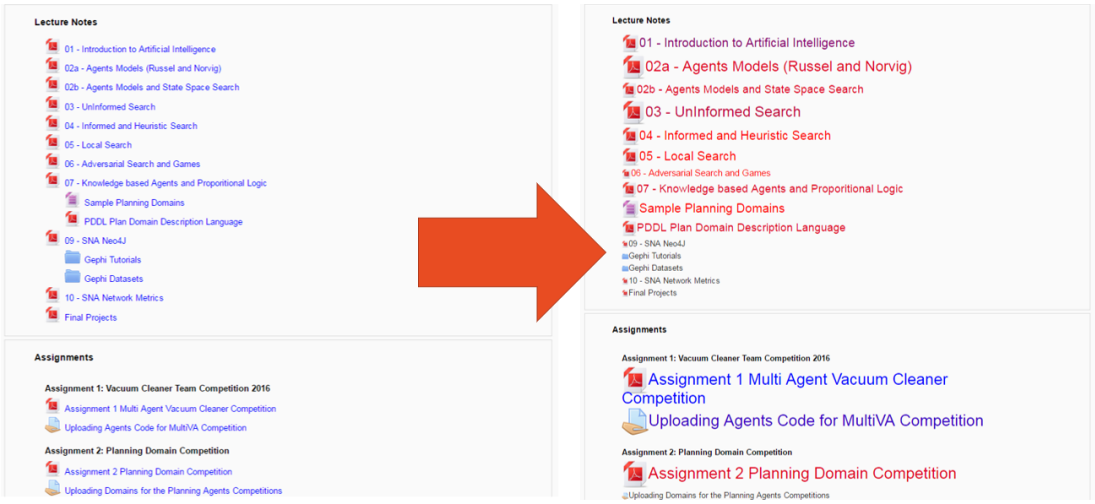

Figure 3.4: Dimensional Morphing Metaphor for learners' monitoring.

colours, i.e. toward the red frequency of the spectrum, while less recently accessed objects are rendered in cold colours, i.e. closer to the blue/magenta end of the spectrum. The resulting colour will then be proportional to the time difference between the object's access date and the reference date. To point out the peculiar status of objects that have had no significant access by the users we use a special colour, either white or black depending on the contrast with the current background.

After analysing the main requirements and choosing the characteristics that should be dynamically presented to the instructors, we designed different monitoring metaphors to represent in-place the quantitative data gathered from the LMS. Dimensional, Bar and Tag Cloud interface morphing effects define the transition from one interface representation to another. In *dimensional morphing*, shown in Figure 3.4, the LMS course content appearance is modified by changing the size (and the colour) of learning objects' icons and labels. While retaining the ease of use of the familiar interface structure, this change in dimensions affects the instructor's perception of the importance that each activity had during the course. The dimensions of the icons and labels are then rendered proportionally to the value.

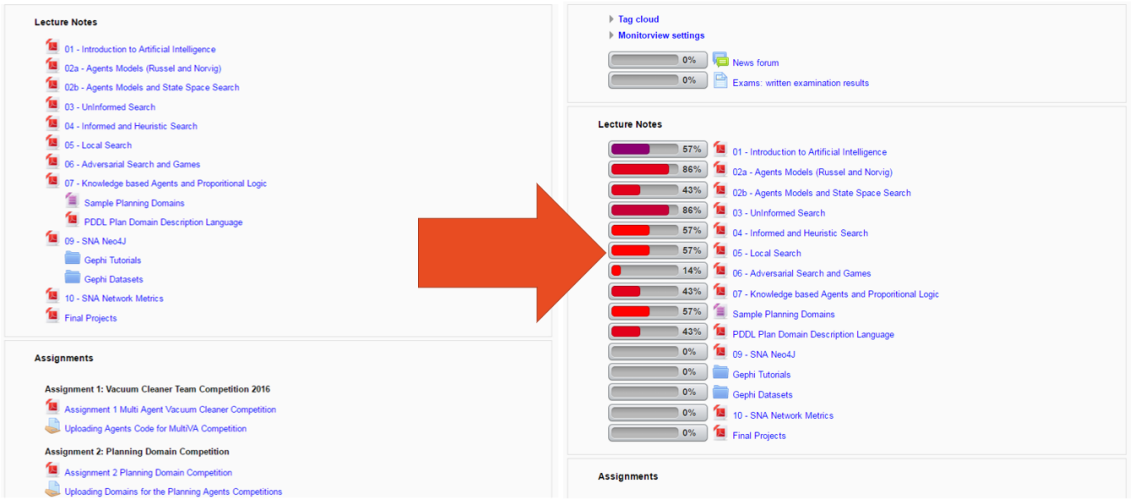

Figure 3.5: Bar Morphing Metaphor for learners' monitoring.

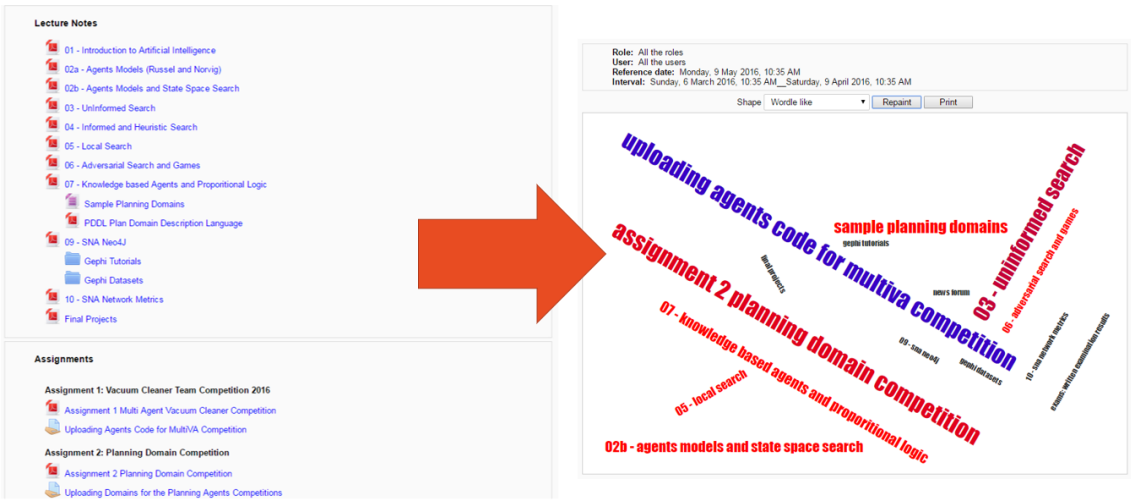

Figure 3.6: Tag Cloud Morphing Metaphor for learners' monitoring.

The *tag cloud morphing* use similar concepts to dimensional morphing, but using a representation with a visually stronger impact as shown in 3.6. In *bar morphing*, the usage quantities are represented using the bar metaphor. Shown in Figure 3.5 the bars near the learning object's description are filled according to the learning objects usage and coloured according to timing of access of the resource.

The approach integrates static and temporal dynamic controls. It can be easily integrated in existing LMS that provide learner activity logs and appearance customisation. Quantitative and qualitative evaluations of experiments show that the proposed approach outperforms traditional log reporting for real-time students' engagement monitoring using cohort and temporal dynamics. The educators reported in interviews that using the proposed tool they acquired valuable feedback on their teaching material and learning designs.

Chapter 4

Community Discovery Approaches

This chapter represents the core contribution of the thesis. We define the problem of modelling, discovering and analysing the user's social communities induced by the observation of user's actions and activities within a system. Moreover, the application domain and the dataset used are described. In the second section we introduce the different approaches that we have explored in our research. The introduced methodology and models have been experimented on the domain of students' activities within Virtual Learning Environments (VLE). Results are reported at the end of the corresponding section, while experiments results evaluation is presented and discussed in the next chapter.

4.1 Problem Description, Domain and Definitions

The purpose of the research is to verify the validity of the assumption that co-occurrence of actions and synchronisation of activities within a system are emergent properties of the underlying social network and that the observation of co-occurrence of user-system interaction can be used to elicit the social bonds which ultimately caused them.

The research is focused in the domain of VLEs to discover the communities of students, basing on the information of their activities within VLEs.

When a student accesses the course material or start an activity in VLE his actions are recorded in the VLE system *logs*. The basic information that can be retrieved comprises the *course* that the student accessed, the *starting time* of student-system activities, the *action type* (e.g. read announcements and forum posts, view and download resources, submit assignments, etc.) and the *learning objects* he interacted with.

We consider the blended learning environment (BLE) scenario. In BLE students interact with the VLE to access learning material and perform online activities supporting the actual course while is held in person in physical classes. Since the environment is blended, the students interact and share information using a number of media and communication channels: in addition to the online tools provided by the VLE, students interact directly in person in the class as well as using social network platforms and instant messaging, apps, etc. The side channel communications are out of VLE and instructors' control scope and their digital traces are then not available to be analysed.

The behaviour of students, when and how they access the VLE can vary under various circumstances. For example, a lecturer asks the students to check on VLE for additional learning material to be used for assessment, in this case nearly all the

students will immediately connect and access the platform to check the examples. In another case, two students studying together decide to revise a certain lesson topic or work more on some difficult concepts found during their study. They will arrange their study schedule, or to check the lesson material, by talking face to face to each other or by exchanging messages through instant messaging apps (e.g. Telegram, Whatsapp, etc.). In this case they could login in the VLE together, or within a short time interval, and access the same learning objects with their own laptop or mobile devices.

From the system point of view, the social interactions, i.e. the teacher *in-class* advice or the students message exchanges on side communication channels, are unknown. We assume that the system can observe the time and type of activities on the VLE and this data can be analysed in order to understand if they are a manifestation of an underlying social channel communication. That is to say that our aim is *reverse engineering* the social communities and communication channels that caused the monitored activities. We focused on a higher abstraction level looking for *social communities of interacting students* instead of single communicating pairs. In fact, co-occurrences take place in a noisy environment, i.e. the students actions can randomly co-occur with no previous communication between them. This random noise makes difficult to identify a *single information exchange* between students.

This framework can give insights on students' interactions, by checking the presence of similar activities within the VLE and their timings. Using this information, we can model the students' social communities by analysing the activities they take on the VLE and use it as the foundation for eliciting the students' social communities.

Community information can be studied at various levels of time and activity granularity, e.g. one or more courses in a time interval that can span from a few weeks to several years, by monitoring the students' activities along the complete

academic career.

4.1.1 Dataset

The VLE dataset that will be used in the experimental tests of our methodology, consists of the system logs of two computer science courses at university master level. The courses have been taught in blended mode with online and offline activities. The online activities include the *access* to weekly release of *learning material*, *project submissions* and *assessment quizzes* on the eLearning platform, while the offline part of the courses includes in-class face-to-face lessons, group project development and final examination. The group projects were finally submitted to the instructor through the online VLE platform for assessment. The students' group composition is known and recorded in the system, group members can access the VLE freely and independently from each other. The dataset consists of all the system log records of students' online activities that take place within the learning platform (e.g. access to teaching material, forum, quiz, feedback, project submission, etc.).

The actual VLE supporting those blended learning courses was the widely known Moodle, a de facto open source standard eLearning platform. Moodle system logs contain various information for each student's event. The general log format include *time*, *user id*, *event context*, *component*, *event name*, *description*, *origin*, and *IP address*.

For the purpose of this research data have been anonymised and all unnecessary information has been removed.

4.1.2 Activity Abstraction Level

In general, given a VLE, we consider different activities hierarchical abstraction levels at which we can conduct the analysis to expose different information and

detail on the students' interactions. For example we can consider user activities at platform level, i.e. co-occurrences of user activities within the whole platform, at course level, i.e. activities co-occurring in the same course, and learning object level, i.e. co-occurrence of activities and actions taken on the same resource/activity in the same course.

Analysing the activities at learning object level gives the most fine level of detail, while the system activity abstraction level is the most coarse one. The risk of analysing the activities at platform level is that it will bring too much noisy information as too many concurrent actions will be elicited. On the opposite, the risk of analysing data at learning object level is that the co-occurrence of information will be too much sparse.

The level of activity abstraction analysis we selected for our approaches is at course level. This choice is motivated from the observation that instructors usually encourage the interaction between students by assigning group projects during a single course. Students should collaborate to develop the projects and this will have an impact on their study activities. We can assume that students working on a project together will study and frequently access the course learning material together.

A preliminary analysis has been conducted to determine if the landscape of the gathered data can convey useful information for our task.

Fig. 4.1 shows the students' activities distribution during the course, grouped by hour. As shown in the image, the online activities of the students mainly happen around the lesson days, to retrieve the lesson materials and assess the lecture notes, and at the end of the course, to revise the course content before the exam. The most interesting information shown in the graph is that the students use the eLearning system almost continuously during the course delivery. The students actions, alone or in small groups, can be found even in unlike hours. This behaviour

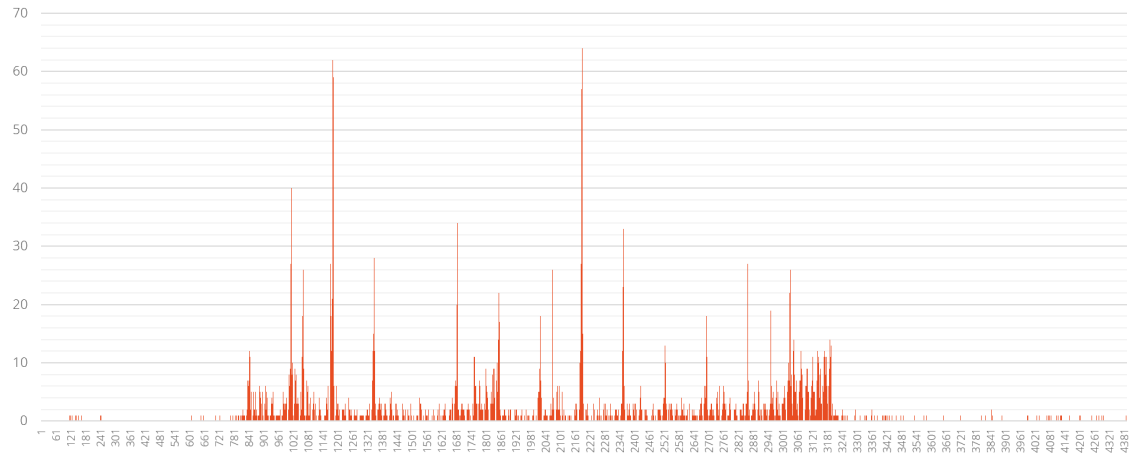

Figure 4.1: Interactions distribution during course using 1h timeslots.

brings important information for our research.

Fig. 4.2 represents, for each student, the total number of activities they had within the VLE during the course enactment. As shown all the students interacted with the VLE, except one. The average of about 53 access for each student, with some peaks over 100. Given that the examined course lasted for 90 days, it results that each student had one access every two days.

4.1.3 Sessions

The session time that each student spend while interacting with the VLE can not be clearly determined. Similarly to visits of web pages, the start of each session can be easily identified by looking at the login or web servers' data delivery actions. On the opposite, logout from the system is difficult to be determined. The users of web-based systems, such as the VLE, tend to just close the web browser at the end of their session instead of using the logout system function. The former behaviour cannot be recorded in the VLE system log. Different methodologies have been proposed to define the session duration.

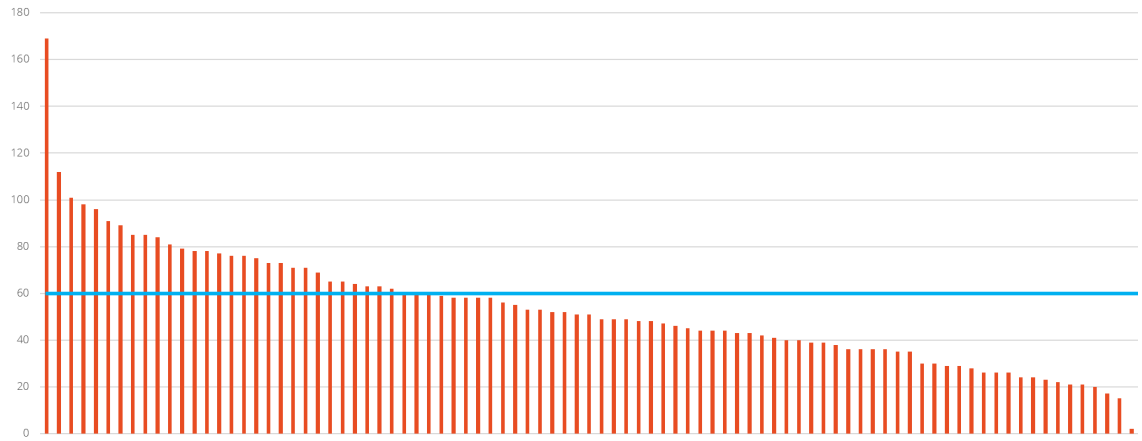

Figure 4.2: Number of accesses by user during course.

Time-oriented approaches base the session duration by setting an *inactivity threshold*. After this threshold is reached the session expires. In literature there is no agreement on how long it should be the threshold for session expiry. Early works proposed that 30 minutes is the average session duration while browsing the web [24]. This assumption has been first accepted and used as industry standard, but rejected by later works where context and user based thresholds have been suggested [69]. In *navigation-based* approaches the session duration is based on the continuity of users' activities within the system. The session is active until the users navigates inside the system.

When needed, we base our approaches using a session threshold σ, expressed in seconds, from the last recorded activity. In Figure 4.3 is represented the typical case of sessions identified for a single user. Sessions $S1$ and $S2$ start with an activity recorded in the system and end after the σ threshold expires. The two sessions are separated as the distance betweed $S1_{end}$ and $S1_{start}$ is greater than σ. In Figure 4.4, Session $S1$ starts and ends like in previous example and the two sessions separation follows the same rules. Session $S2$ starts with the first interaction recorded after the gap, and ends after the expiration of the session threshold after the last interaction.

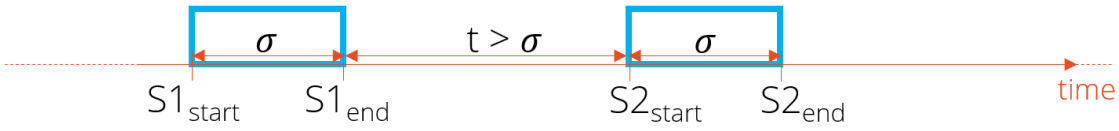

Figure 4.3: Representation of simple sessionsand separation example.

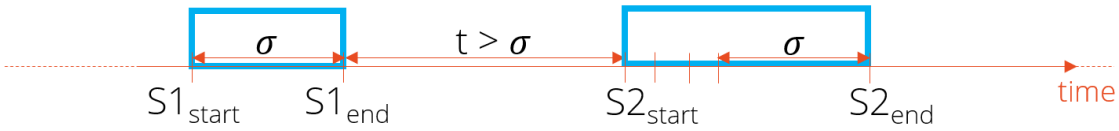

Figure 4.4: Sessions representation and separation example.

The other interactions extend the duration of the session as their distance in time from the previous interaction is smaller than the σ threshold.

4.1.4 Granularity

The granularity of activities refers to the continuity of activities' observation. Depending from the environment we can classify the granularity types in:

- **Continuous**: CCTV cameras are an example of continuous granularity. The observation of activities is continuous until the people is in the camera field of view;
- **Periodic**: Cellphone polling is an example of periodic granularity. The observation of the cellphone position (i.e. to which cell it is connected) is recorded only when a central system request it. Between two polling intervals the position of the cellphone is unknown;
- **Event**: Logs are a typical example of event granularity. An activity is recorded in system logs only when a certain event is verified, e.g. a user login, a light is switched on, etc.

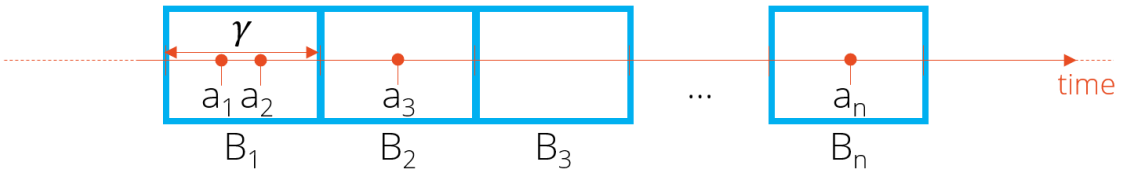

Figure 4.5: *Bucketing* of various activities.

The granularity of activities can be transformed from event to periodic using *bucketing* technique. Event activities are grouped together in single buckets when they are within a certain γ time threshold. In Figure 4.5 the bucketing technique is represented. The timeline is divided in n buckets, each B_i bucket has size γ and contains information about all the activities a_i whose timestamps is inside the bucket interval.

4.1.5 Ground Truth

In order to analyse and compare the results of the students' clustering algorithms, the reference *ground truth* which has been used is the student's groups composition. Groups are a non overlapping partition of the students, which is established at the beginning of the courses. Many assignments and lab activities are group oriented and a relevant number of the final work is represented by the group projects, which require a close interaction of the group members along the course period for project development, report writing and presentation. It is quite reasonable to assume that the group is a relevant social group of people studying and holding course activities with similar paces.

The aim of our experiments is to explore the correlation between the grouping found using the social community detection techniques with the actual course groups of students. The students attending the courses included in our dataset were required to develop a group project and submit to the instructor their grouping information.

Table 4.1: Composition of the courses included in dataset.

	Number of Students	Number of Groups
Course C1	89	19
Course C2	61	12

Group projects cover multiple topics taught during the course. Hence, the students are required to have an overall view on all the course content.

The number of students enrolled in each course and the number of groups that have been formed is summarized in Table 4.1.

The students are encouraged to have social interactions, relationships and collaborate with peers to reach a common goal. Without losing generality, we assume that students that work together to complete the assignment will study together for all the tasks required during the course.

4.1.6 Evaluation Criteria

To compare with the ground truth, we fixed the number of clusters that the different clustering algorithms should produce to be the same as the number of students' groupings submitted to the instructor.

To evaluate the quality of the clustering labels assignment, different measures highlight different qualities of the detected clusters. Homogeneity, Completeness and V-measure score [87] are some of the most common measures to evaluate the quality of clustering results. These measures are completely independent of the number of classes, the number of clusters, the size of dataset and the clustering algorithm used, thus can be applied to and compared across any clustering solution.

Definitions. Given a dataset comprising N data points, and two partitions of these: a set of classes $C = \{c_i \mid i = 1, \ldots, n\}$, a set of clusters $K = \{k_j \mid j = 1, \ldots, m\}$. Let $A = \{a_{ij}\}$ be the contingency table produced by the clustering algorithm, representing the clustering solution, where a_{ij} is the number of data points member of the class $c_i \in C$ and elements of the cluster $k_j \in K$. In our case, the classes are the subdivision of the students in the ground truth grouping assignment.

Homogeneity. Measures the quality of the cluster assignment by checking if the clusters contain only elements of a single class in the ground truth assignment.

In order to satisfy the homogeneity h criteria, a clustering must assign only the data points that are members of a single class to a single cluster. The value $H(C|K)$ represents the conditional entropy of the class distribution given the proposed clustering, i.e. how close the given clustering to the ideal. The normalisation factor $H(C)$, needed to render the evaluation not dependent on the size of the dataset and the distribution of classes sizes, represents the maximum reduction in entropy that the clustering information could provide.

$$h = \begin{cases} 1 & \text{if } H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{otherwise} \end{cases} \tag{4.1.1}$$

where

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{a_{ck}}{N}\log\frac{a_{ck}}{\sum_{c=1}^{|C|}a_{ck}} \tag{4.1.2}$$

$$H(C) = -\sum_{c=1}^{|C|}\frac{\sum_{k=1}^{|K|}a_{ck}}{n}\log\frac{\sum_{k=1}^{|K|}a_{ck}}{n} \tag{4.1.3}$$

Completeness. Measures the quality of the clustering by checking if all the elements of a single class are assigned to the same cluster.

Completeness c is symmetrical to homogeneity. In order to satisfy the completeness criteria, a clustering must assign all the data points that are members of a single class to a single cluster. The value $H(K|C)$ represents the conditional entropy of the cluster distribution given the class of the data points. The value is normalised using the $H(K)$ factor that represents the maximum distribution of cluster sizes.

$$c = \begin{cases} 1 & \text{if } H(K,C) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{otherwise} \end{cases} \tag{4.1.4}$$

where

$$H(K|C) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{a_{ck}}{N}\log\frac{a_{ck}}{\sum_{k=1}^{|K|}a_{ck}} \tag{4.1.5}$$

$$H(K) = -\sum_{k=1}^{|K|}\frac{\sum_{c=1}^{|C|}a_{ck}}{n}\log\frac{\sum_{c=1}^{|C|}a_{ck}}{n} \tag{4.1.6}$$

V-measure. Measures how successfully the criteria of homogeneity and completeness have been satisfied. It is computed as the harmonic mean of distinct homogeneity and completeness scores, like precision and recall are combined to compute the F-measure score [84].

$$V_\beta = \frac{(1+\beta) * h * c}{(\beta * h) + c} \tag{4.1.7}$$

V-measure score V_β is weighted by factor β. If β is greater than 1 completeness is weighted more strongly in calculation, if β is less than 1, homogeneity is weighted more strongly.

To evaluate our experiments we choose to use $\beta = 1$ to have a balanced score between homogeneity and completeness.

4.2 Histories Binary Correlation Analysis

The *Histories Binary Correlation* analysis approach focuses on the concept of user's activity history vector, i.e. a vector which represents the activities held by the student along a fixed set of predefined time slots. This approach falls into the behavioural-based social community detection category as introduced in Section 2.3. The co-occurrence of activities gathered from students' logs conveys information about the underlying students' social relationships and communications. The *student-system activity histories* are used in our framework to discover and analyse the latent students' social network.

This method exploits the similarity of student-system activity histories to determine students' social proximity and characterise group memberships.

This approach works without using any information about the students' social network, as it relies only on student history similarities to elicit the underlying social communities.

In Figure 4.6 is represented the workflow of the Histories Binary Correlation approach. Starting from the logs of students activities generated by the VLE (phase a), an History Binary Summarised Vector (HSV) is obtained for each student (phase b). The HSVs are compared (phase c) using a binary similarity measure φ and then this information is used to extract the social communities information using clustering algorithms (phase d). The obtained social communities are finally evaluated (phase e) by comparing them with the ground truth.

4.2.1 Interaction Modelling

The information needed to model the students-system activity histories is extracted from the VLE system logs. Each history is representative of the activities of a student within the VLE at system level (see Section 4.1.2). From each log record the

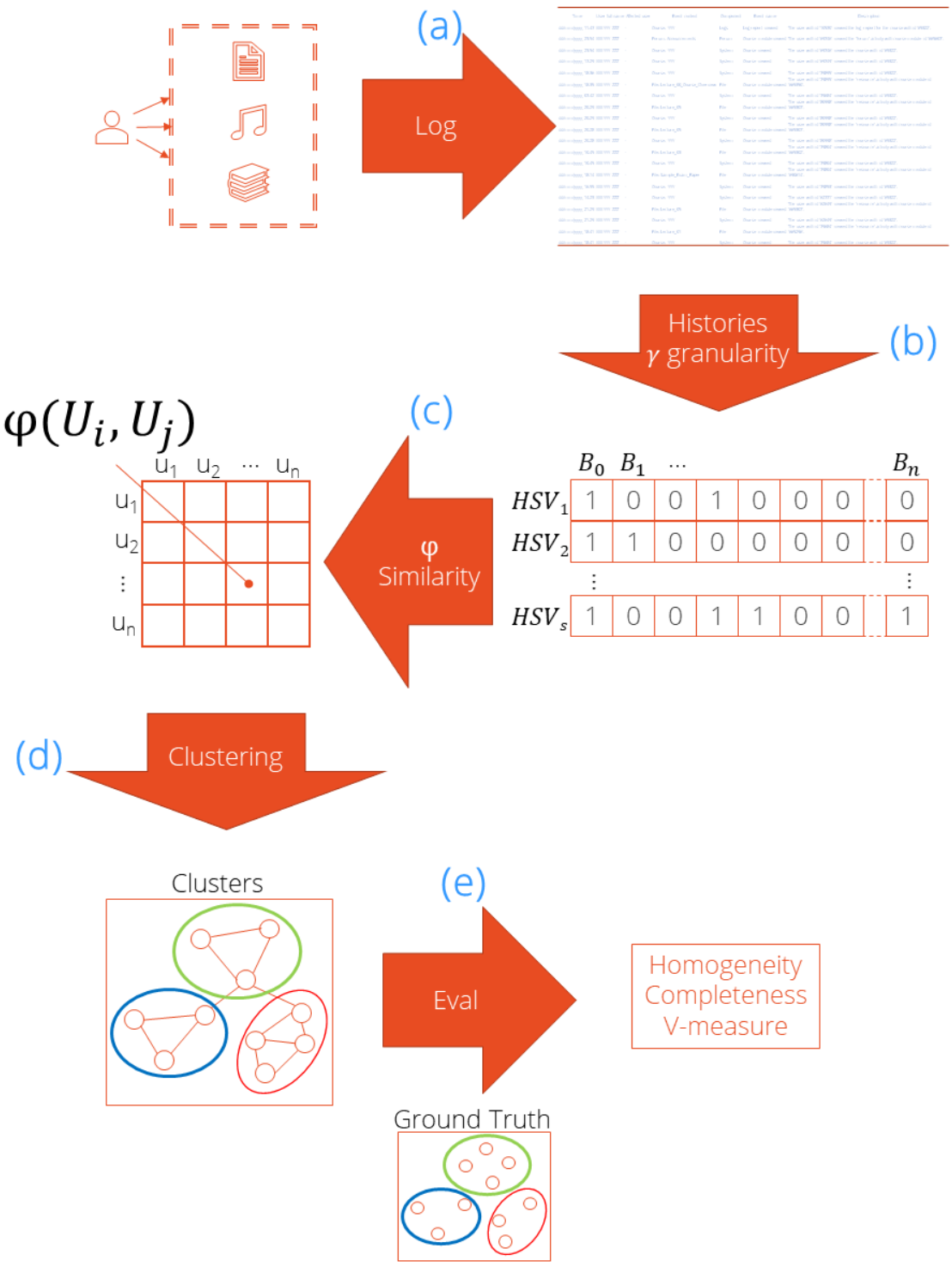

Figure 4.6: Visual scheme of the Histories Binary Correlation Analysis approach.

information about the timing of actions is extracted and recorded in the appropriate activity history.

As introduced in Section 4.1.4 data in system logs is at *event* granularity. We transformed this data to *periodic* using the bucketing technique. The activities are considered in time slots of varying duration, using a γ granularity parameter. This is used in our approach to highlight the *student-student interactions* considering the time for communications to diffuse among the students. The course is divided in many time slots as needed as in Equation 4.2.8.

$$History\ Length = \frac{Period\ Length}{\gamma} \tag{4.2.8}$$

$History\ Length$ is expressed in seconds and the γ granularity parameter is set at time threshold of 1 hour (3600 seconds). The latter approximation is consistent with the assumption introduced in Section 4.1.3.

Definition *History-Summarised Vector (HSV)* of VLE activities of a student i is a boolean vector of length $HistoryLength$ where each boolean entry j record with the value $HSV_i[j]$ for the presence or absence of activities in the appropriate time slot j.

In Figure 4.7 is represented the HSV creation starting from one user interactions with the VLE. The binary value 1 in the first HSV slot B_0 is representing the summary of both the activities a_0 and a_1. In fact, the first two activities timing is inside B_0 time slot and the 1 binary value is representing the presence of *at least* one interaction in that time slot.

4.2.2 History Similarity

A pairwise comparison of student-system activities history vectors is computed to build a matrix of student-student similarities. Various proximity measures can be

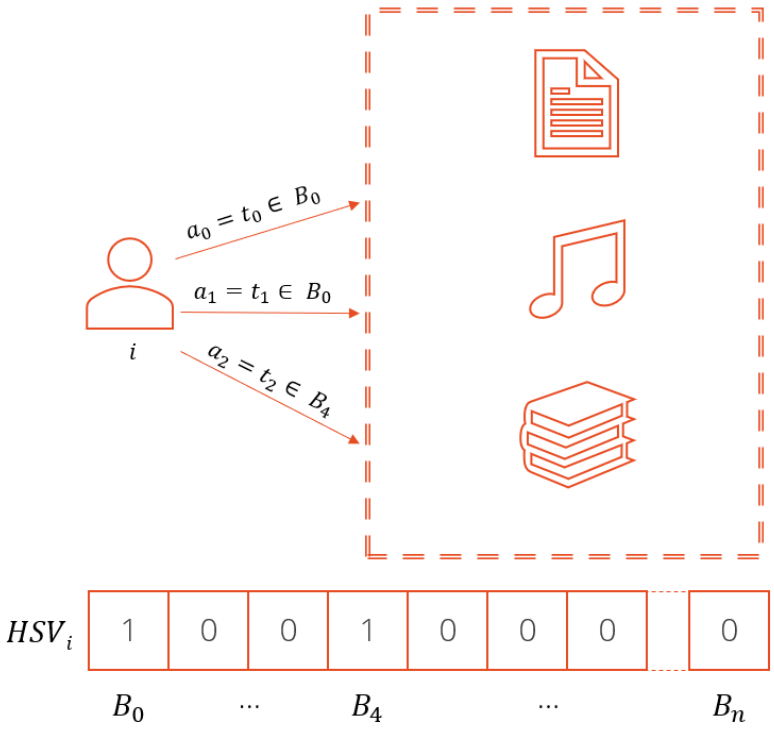

Figure 4.7: History Summarised Vector with binary values representing the presence of interactions.

Table 4.2: Contingency table for pairwise students' activities

	Student 1 Activity present	Student 2 Activity absent
Student 1 Activity present	a	b
Student 2 Activity absent	c	d

calculated by checking the co-occurrence of activities at the predetermined time granularity.

The different binary vector similarity measures, as introduced in Section 2.3, emphasize different aspects of the students-student co-occurrence of activities. Students' HSVs are compared pairwise to compute the parameters by checking the co-occurrence of activities in the same time slot. The parameters' contingency table, derived from Table 2.1, is shown in Table 4.2.

The parameters are computed according to the following formulas, derived from the ones introduced in Section 2.3.1:

$$a = \sum_{t=0}^{n} \left(HSV_1 \left[B_t \right] * HSV_2 \left[B_t \right] \right) \tag{4.2.9}$$

$$b = \sum_{t=0}^{n} \left(HSV_1 \left[B_t \right] + HSV_2 \left[B_t \right] \right) \tag{4.2.10}$$

$$c = \sum_{t=0}^{n} \left(HSV_1 \left[B_t \right] + HSV_2 \left[B_t \right] \right) \tag{4.2.11}$$

$$d = \sum_{t=0}^{n} \left(1 - HSV_1 \left[B_t \right] \right) * \left(1 - HSV_2 \left[B_t \right] \right) \tag{4.2.12}$$

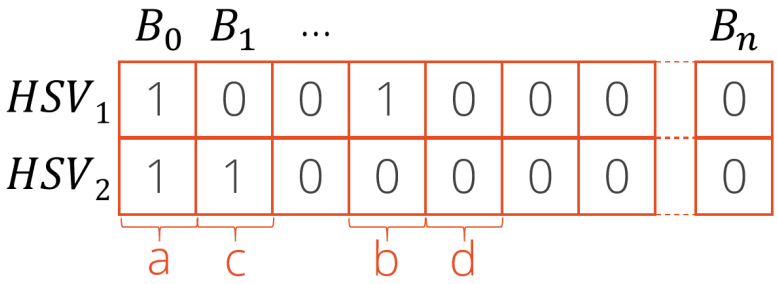

Figure 4.8: History Summarised Vectors pairwise comparison to extract measure parameters.

According to the contingency table 4.2, the value of the corresponding parameter is increased by one for each time slot that verifies one of the four conditions. In Figure 4.8 is represented the comparison of two HSV to extract the measure parameters.

4.2.3 Amplifying Parameters

Parameters as shown in Table 4.2 are the basic form to compute binary similarity measures. This representation of student-student activities similarity does not take in account the context in which their activities are immersed.

Adamic and Adar [1] introduced a concept in link prediction where, when predicting the connection between two nodes on a network, the very large neighbourhoods are less significant than the small ones. The value of Adamic-Adar index A for node pair is defined by:

$$A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log|N(u)|} \tag{4.2.13}$$

where x, y, and u are nodes of the network and $N(u)$ is the set of neighbours of node u. The formula makes use of the inverse log frequency of the number of common neighbours, thus amplifying the value of common neighbours that are

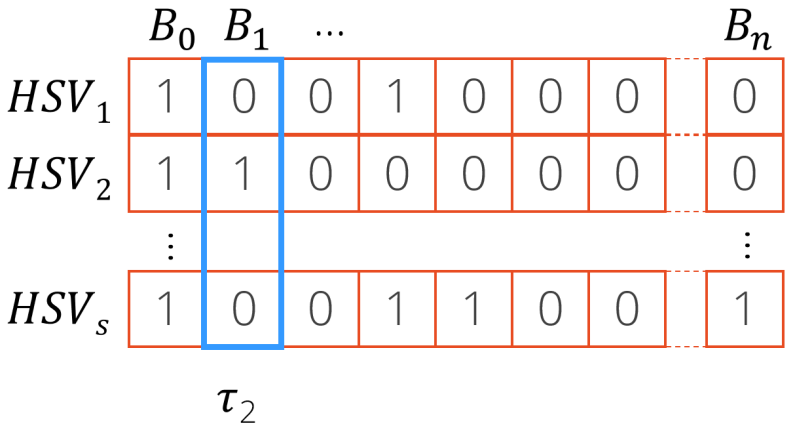

Figure 4.9: Visual representation of τ_t entropy amplification parameter extraction for a timeslot of History Summarised Vectors.

uniquely connected to a few users.

In analogy to this interpretation, the activities that happen when everyone is connected to the system are less significant than the ones that we record when only a small number of people is present. For example, when two students are connected to the VLE at unlikely hours without anyone else online brings more information about their relationship than when the same two students are connected before the lesson and everyone is consulting the learning material.

To reflect this concept we introduced an innovative Entropy Amplification (EA) factor τ_t for the measures' parameters. In the EA parameter we take in account the number of total activities in the considered timeslot t, as shown in Equation 4.2.14.

$$\tau_t = \sum_{i=0}^{s} HSV_i\,[t] \tag{4.2.14}$$

where s is the total number of students enrolled in the course. The τ_t computation for one time slot is represented in Figure 4.9.

We introduce various different strategies to apply the EA to the measures parameters. The first strategy is to correct only the co-occurrence parameter (i.e. the

a parameter), the second is to apply EA to all the parameters and the third strategy is to apply an inverse entropy amplification for the absence of activities parameter (i.e. the d parameter).

Amplifying co-occurrence parameter (*EA-a*). This EA strategy consists in computing the a parameter of the contingency table 4.2 for Student 1 and Student 2 by a *weighted counting*. Each co-occurence of activity is weighted by a factor that is inversely proportional the total number of activities on the considered time slot.

$$a = \sum_{t=0}^{n} \frac{2}{\tau_t} \left(HSV_1 \left[B_t \right] * HSV_2 \left[B_t \right] \right) \tag{4.2.15}$$

Notice that $\tau_t = 1$ when only two people are using the VLE in that time slot and $\tau_t \simeq 0$ when the number of people connected tends to $+\infty$.

All the other parameters b, c, and d are computed using a weight of 1 as in the standard parameters counting.

Amplifying all parameters (*EA-abcd*). In this EA strategy, the a parameter is computed as in Equation 4.2.15. The EA applied to the other parameters is done according the following equations:

$$b = \sum_{t=0}^{n} \frac{1}{\tau_t} \left(HSV_1 \left[B_t \right] + HSV_2 \left[B_t \right] \right) \tag{4.2.16}$$

$$c = \sum_{t=0}^{n} \frac{1}{\tau_t} \left(HSV_1 \left[B_t \right] + HSV_2 \left[B_t \right] \right) \tag{4.2.17}$$

$$d = \sum_{t=0}^{n} \frac{1}{\overline{\tau_t}} \left(1 - HSV_1 \left[B_t \right] \right) * \left(1 - HSV_2 \left[t \right] \right) \tag{4.2.18}$$

$$\text{where} \quad \overline{\tau_t} = \begin{cases} \frac{1}{\tau_t} & \text{if } \tau_t \neq 0 \\ 1 & \text{otherwise} \end{cases} \tag{4.2.19}$$

Amplifying absence parameter (*EA-ad*). In this EA strategy a, b, and c parameters are modified respectively as in Equations 4.2.15, 4.2.16, and 4.2.17. Parameter d is modified using a strategy similar to the strategy a.

The d parameter is computed by a weighted counting that is inversely proportional to the number of *missing* activities in the considered time slot. The less people is absent in the considered time slot, the more informative the information is.

$$\mu_t = s - \tau_t \tag{4.2.20}$$

$$d = \sum_{t=0}^{n} \frac{2}{\mu_t} \left(1 - HSV_1\left[B_t\right]\right) * \left(1 - HSV_2\left[B_t\right]\right) \tag{4.2.21}$$

4.2.4 Measures

Different measures emphasize different aspects of the student-student co-occurrence of activities. Various binary similarity measures have been introduced in Section 2.3.1. The most relevant that we considered to compare pairwise the HSVs are:

- Cooccurence (Equation 2.3.11)
- Jaccard similarity measure (Equation 2.3.12)
- Russell and Rao (RR) similarity measure (Equation 2.3.13)
- Simple Matching (SM) similarity measure (Equation 2.3.14)
- Dice similarity measure (Equation 2.3.15)
- Sokal and Sneath ($SS1$) similarity measure n. 1 (Equation 2.3.16)
- Ochiai similarity measure (Equation 2.3.20)

Each measure, tested with each of the different EA strategies, has been used to build the *student-student similarity matrix*.

4.2.5 Clustering

After the student-student similarity matrix is built, this is used to elicit the students' social communities using clustering techniques.

Since our methodology is general, we selected various clustering algorithms to elicit the clusters of students hence labelling each student with cluster assignment labels. Each label is regarded as the social community group assignment of the student.

The various student-student similarity matrices, each representing a different of combination of EA strategy and measure, have been used in combination with various and different clustering techniques. Each matrix is used to compute the student-student similarities needed by the clustering algorithms.

The clustering algorithms that we considered to elicit the students social communities are:

- Agglomerative
- Expectation Maximization
- K-Means
- Spectral

The characteristics of each algorithm have been discussed in Section 2.2.

4.2.6 Experiments and Evaluation Criteria

Experiments have been held using the dataset described in Section 4.1.1, using all the measures, EA strategies and clustering algorithms previously introduced. In

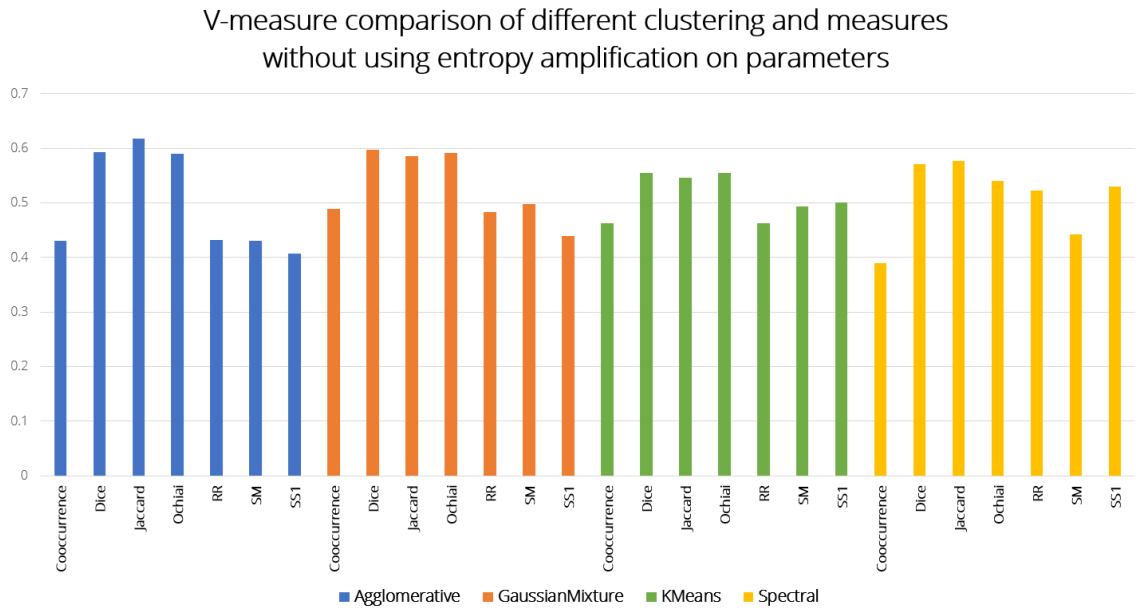

Figure 4.10: V-measure computed on cluster assignments without for course C1 using entropy amplification on measures.

total 112 combinations of measure, parameter and clustering techniques were tested for each course.

The evaluation criteria for cluster labelling assignments has been introduced in Section 4.1.6. Homogeneity, Completeness and V-measure scores are computed to evaluate the quality of the retrieved social communities when compared to the ground truth communities of students.

4.2.7 History Binary Correlation Discussion

The full results of cluster assignment evaluation in terms of homogeneity, completeness and V-measure can be found on Appendix 1, Table A.1 and Table A.2.

In Figures 4.10, 4.11, 4.13 and 4.12 the results of the evaluation of histories binary correlation are presented in terms of V-measure scores. The different clustering techniques are used with different similarity measures and different entropy amplifi-

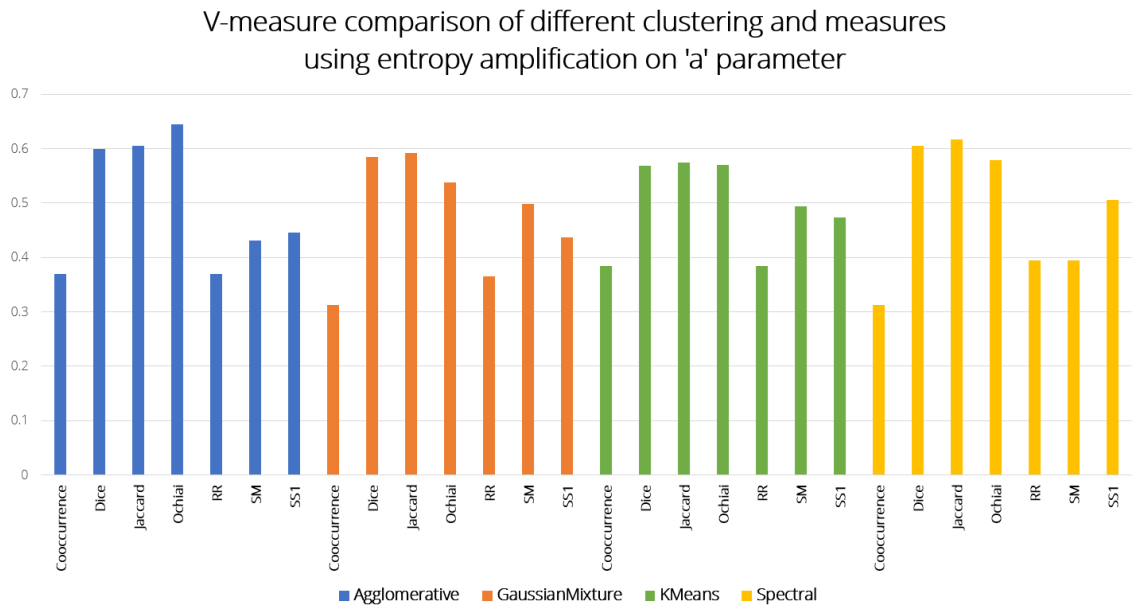

Figure 4.11: V-measure computed on cluster assignments for course C1 using entropy amplification on a parameter.

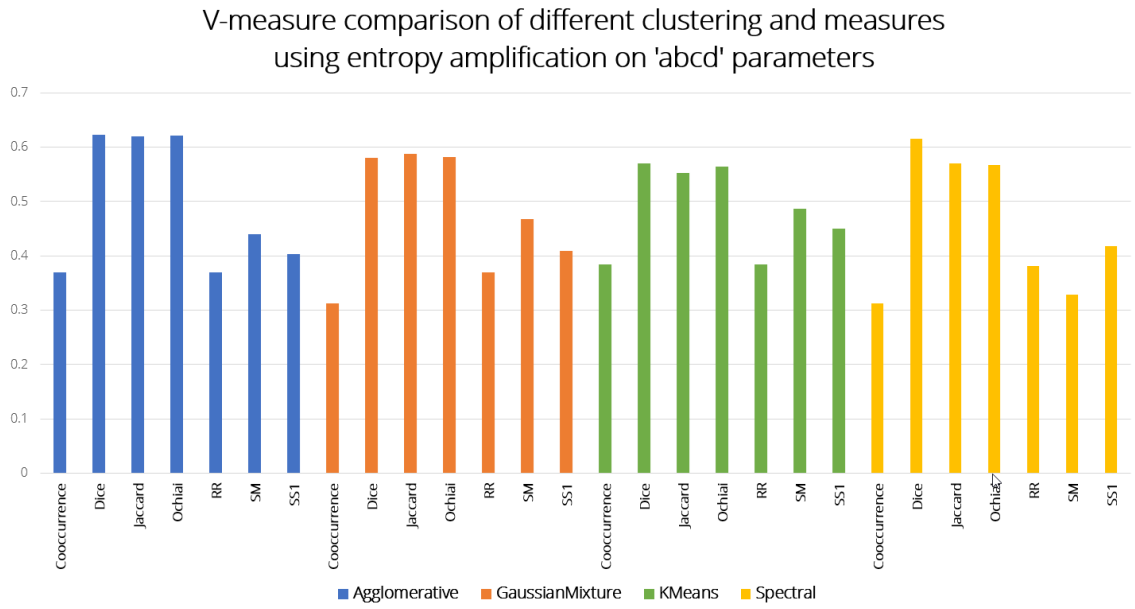

Figure 4.12: V-measure computed on cluster assignments for course C1 using entropy amplification on all parameters.

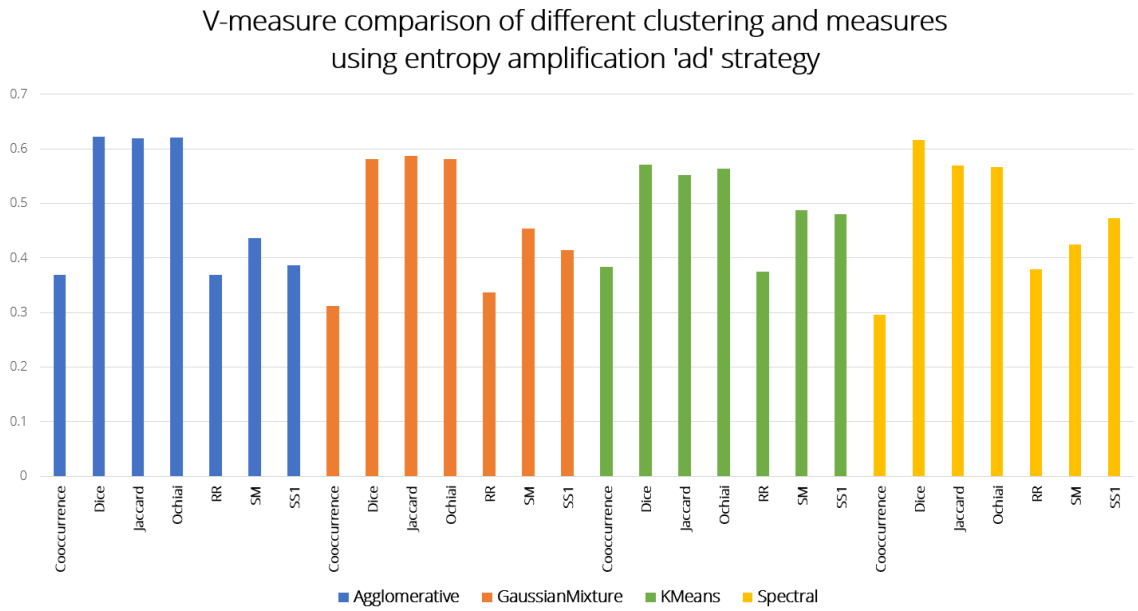

Figure 4.13: V-measure computed on cluster assignments for course C1 using entropy amplification on all parameters and missing activities for d parameter.

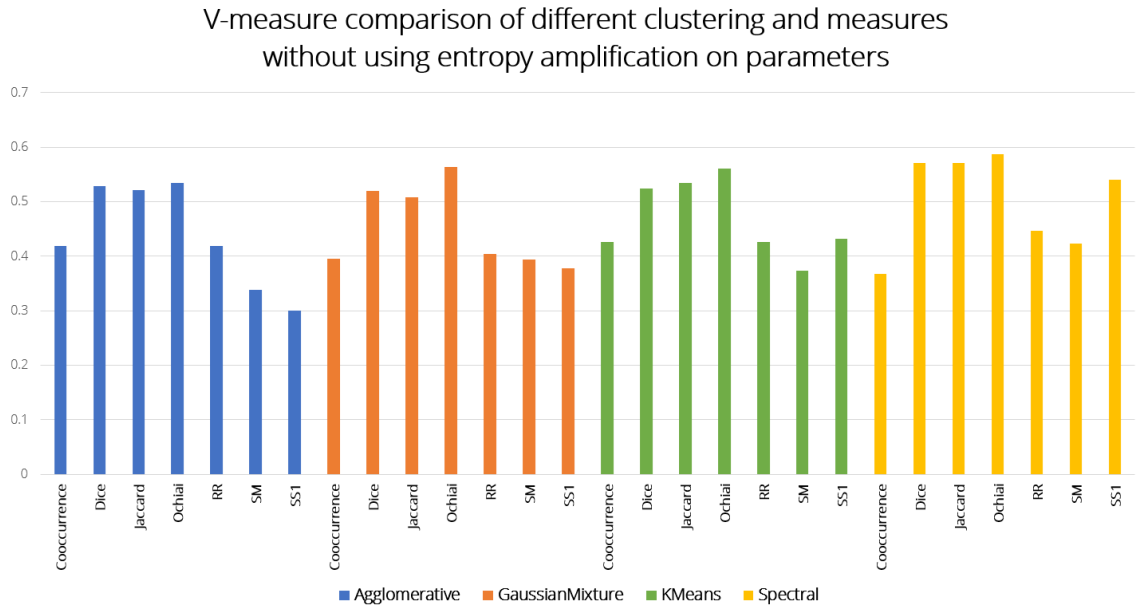

Figure 4.14: V-measure computed on cluster assignments for course C2 without using entropy amplification on measures.

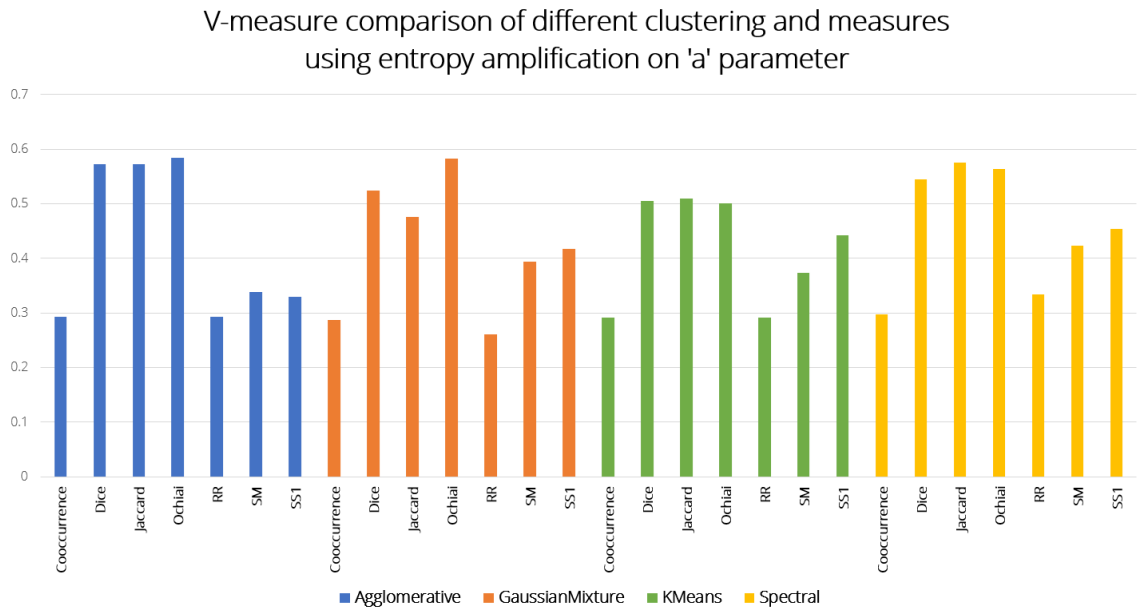

Figure 4.15: V-measure computed on cluster assignments for course C2 using entropy amplification on a parameter.

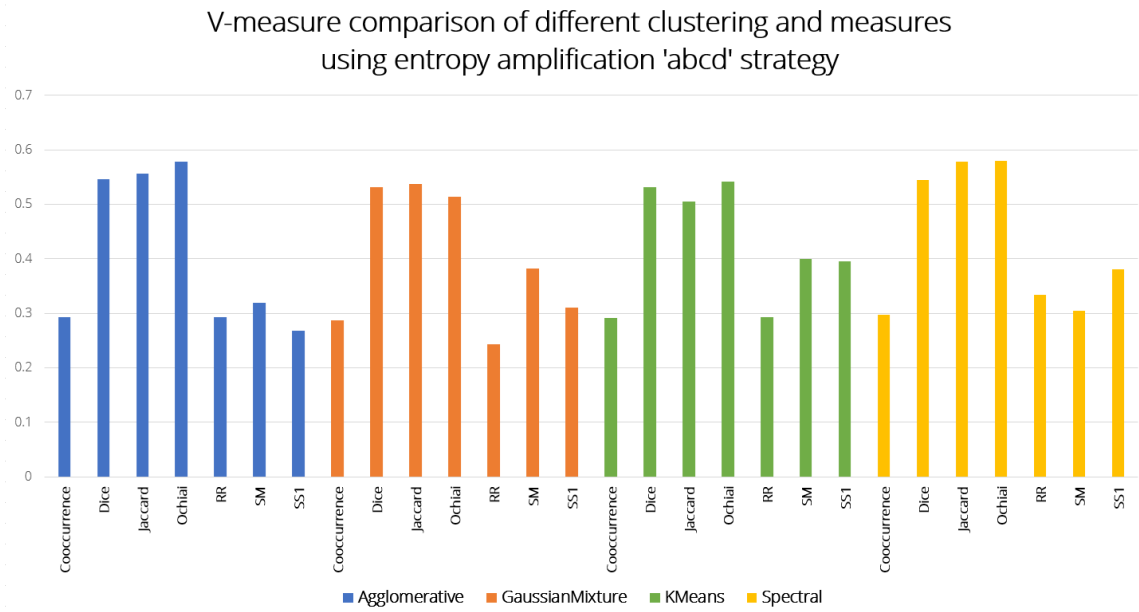

Figure 4.16: V-measure computed on cluster assignments for course C2 using entropy amplification on all parameters.

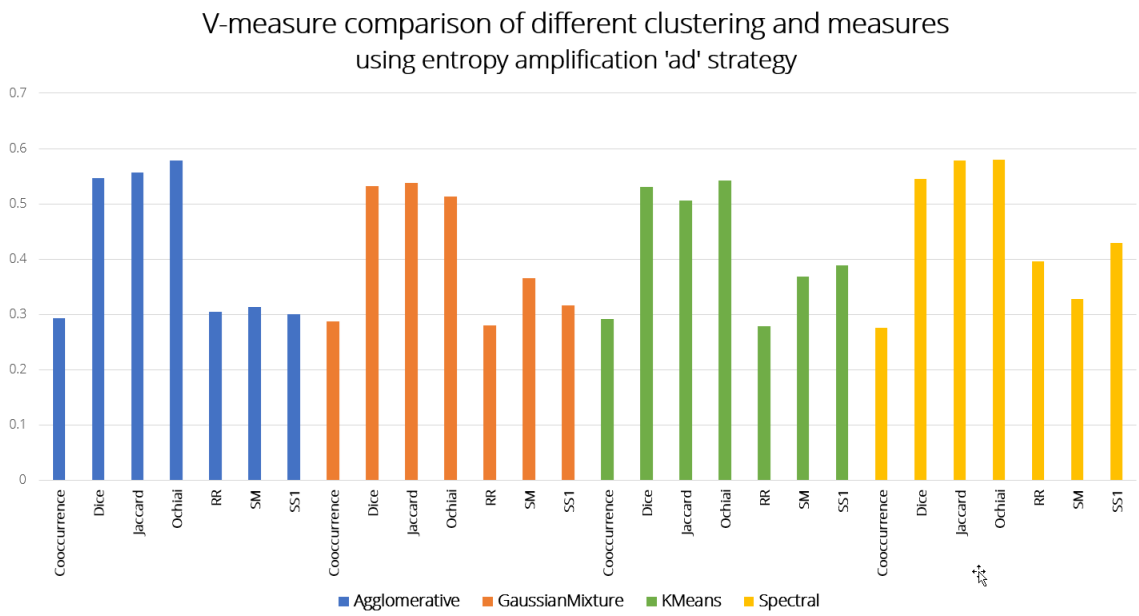

Figure 4.17: V-measure computed on cluster assignments for course C2 using entropy amplification on all parameters and missing activities for d parameter.

cation (EA) strategies on data from course C1. In Figures 4.14, 4.15, 4.17 and 4.16 the V-measure scores are displayed for each similarity measure, EA strategy and clustering algorithm combination on course C2.

As shown, for course C1 the best performance in community detection is achieved by the Agglomerative clustering algorithm when combined with Dice, Jaccard and Ochiai similarity measures. Only Dice measure in combination with Spectral clustering can get close to the other results. These combinations outperform all the other clustering-measure-EA matchings on average by more than 2%.

The different EA strategies help to improve the results of the best three measures. In Figure 4.18 the impact of different EA strategies is displayed, showing the results of the different measures computed on course C1 using Agglomerative clustering. The comparison is with the measures computed without using entropy amplification. As shown, the EA strategy *EA-abcd* applied to the different measures has the best

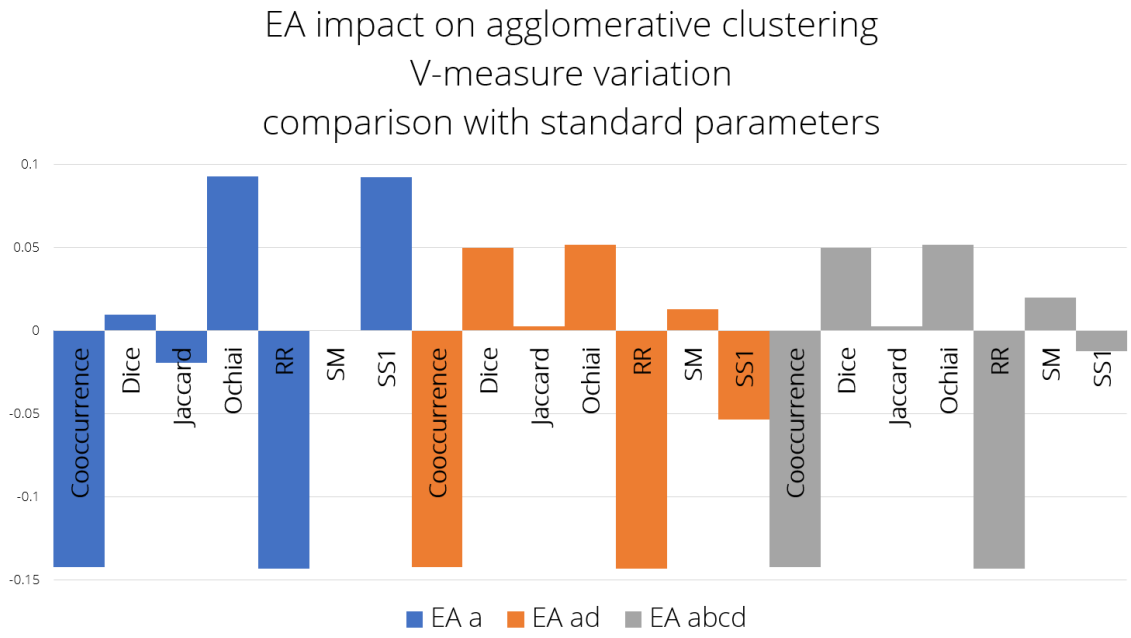

Figure 4.18: Impact of EA strategies on similarity measure calculation and Agglomerative clustering on course C1.

general performance. Even if the *EA-a* strategy is achieving very good results combined with Ochiai and *SS1*, the results of remaining measures are not improving as much as with the other strategies. The results of *EA-ad* strategy are close to the *EA-a* strategy on the best performing measures, with slight differences on Dice, *SM* and *SS1*.

For course C2 the best results are achieved by Agglomerative and Spectral clustering algorithms, with very slight differences between each other. In combination with Dice, Jaccard and Ochiai measures the community detection approach achieve comparable results to the ones on course C1.

The entropy amplification strategies have a stronger impact on the different measures when compared with course C1. As shown in Figure 4.19, in this case the Agglomerative clustering community detection performance is improving the most when used in combination with the *EA-a* strategy. The *EA-abcd* strategy is not

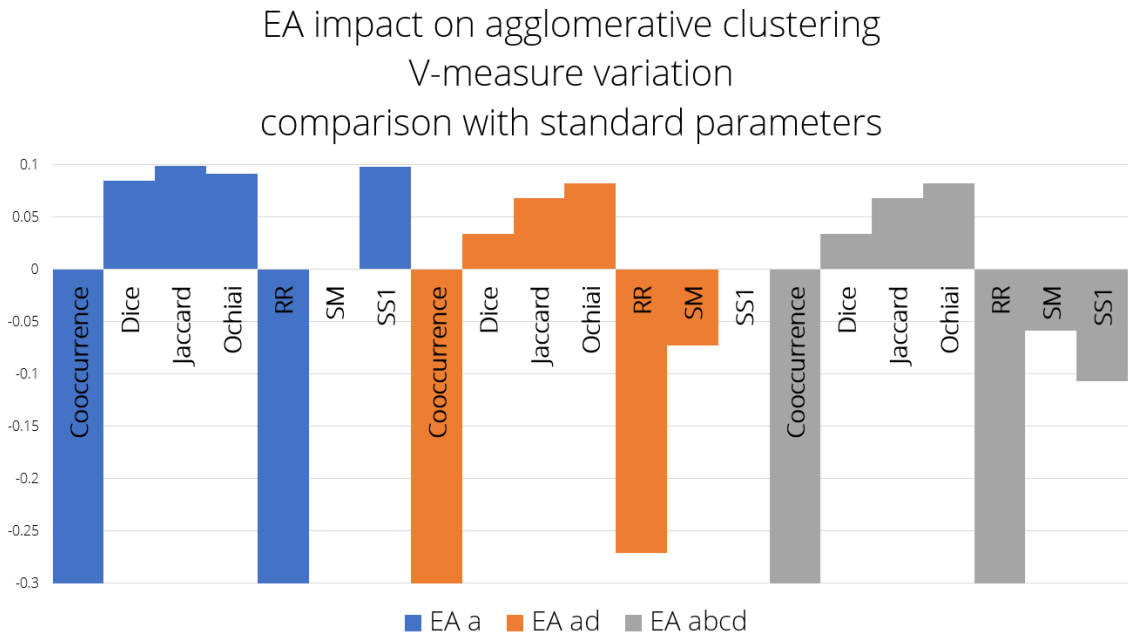

Figure 4.19: Impact of EA strategies on similarity measure calculation and Agglomerative clustering on course C2.

achieving good results, especially with *SM* and *SS1* measures, even when compared with the *EA-ad* strategy.

Figure 4.20 shows detailed results for course C1. The best clustering algorithm is Agglomerative in this case and the best performing Entropy Amplification (EA) is on all the parameters. From the chart, we can find that Dice, Jaccard and Ochiai measures achieve better results in terms of completeness than homogeneity. This means that using this measures the communities elicited from our approach are maintaining together most of the students in a same ground truth group.

For course C2, in Figure 4.21 results of the best performing measures are shown. In this case the best clustering algorithm is Spectral and the EA strategy on the co-occurrence parameter only. Dice, Jaccard and Ochiai measures are the best performers also on data from this course. Most of the students from the same group are clustered together, as reported by the completeness score.

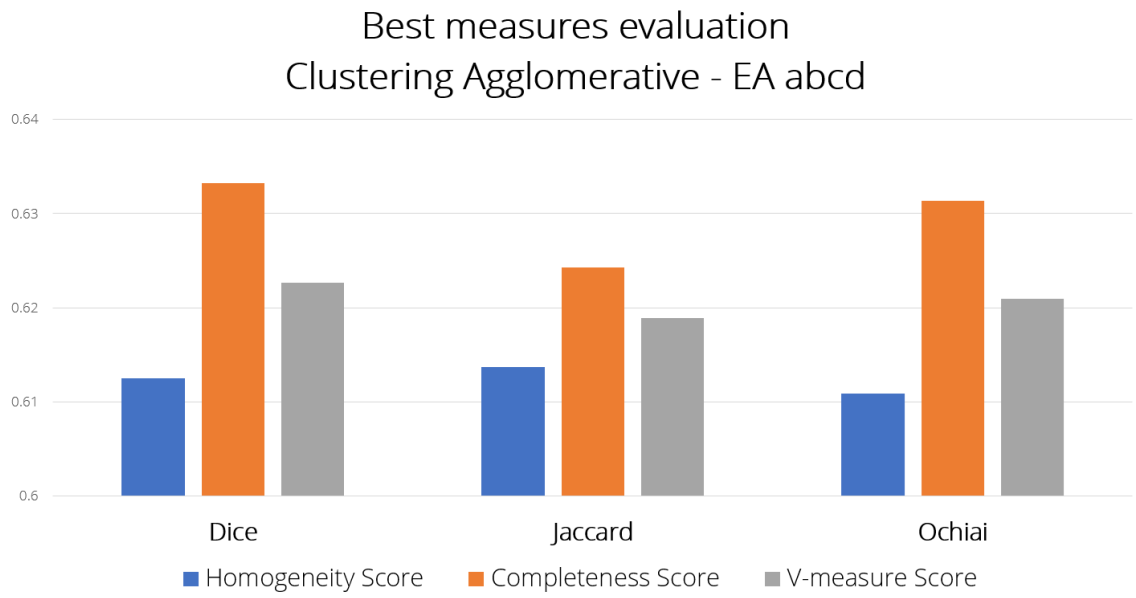

Figure 4.20: Homogeneity, Completeness and V-Measure of best performing measures and EA strategy using Agglomerative clustering on course C1.

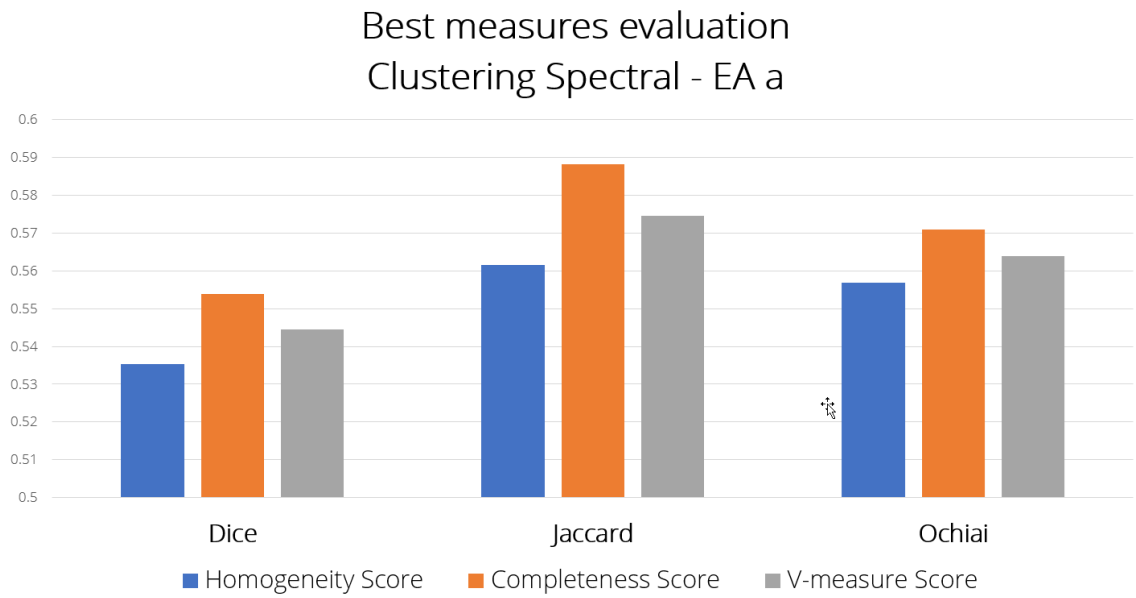

Figure 4.21: Homogeneity, Completeness and V-Measure of best performing measures and EA strategy using Spectral clustering on course C2.

In general each clustering method works best using different similarity measures. All the clustering algorithms have good performances, in terms of V-measure and Homogeneity scores, when used in combination with using Dice, Jaccard, and Ochiai measures. For Spectral clustering, *SS1* measure help the algorithm to obtain good results.

In general, Agglomerative and Spectral clustering algorithms have a better performance in identifying the ground truth students' groupings.

Jaccard and Ochiai similarity measures, in general, are the best performing measures that help the various clustering algorithms to differentiate the students activities and to have more homogeneous and complete cluster assignments. As shown in figures, different clustering algorithms have varying performance when matched with different proximity measures.

A common trait of the best performing measures is that the concurrent absence of activities is not taken in account. In our test setting, considering the granularity parameter γ set at 1 hour time frame, the presence of activities has very sparse values in the students' HSV when compared with the absence of activities. This fact can influence the students' similarity when using measures that take in account the d parameter of the contingency table.

4.3 Session Based Analysis

When a student accesses the course material, his actions are logged by the eLearning system. The basic information that is recorded comprises the *course* accessed, the *timing*, the *action* (e.g. reading announcements, viewing and downloading resources, submitting assignments, answering to forum posts, etc.) and the *specific learning objects* accessed by the student.

The focus of the Session approach is on students' activities in the VLE as it is based on evaluating the presence of activities, their timings and their duration. Using this information, the students' behaviour is modeled by analysing their activities within the VLE and using it as the foundation for eliciting the students' connections, social communities and study behaviour. This approach falls into the category of the network-based community discovery approaches described in Section 2.1.

Basic idea of the Session based analysis is to use the information about co-occurrence of student-system activities within sessions to build a student-student co-occurrence network. The information conveyed by the co-occurrence graph is used to determine students' connections and identify their social communities.

Moreover, this approach has several additional advantages that come from the ability to inspect the network structure. Members of the communities can be differentiated on the basis of their positioning in the network and their neighbourhood connections. For example community *leaders* and *followers* can be identified using the information from centrality measures together with the timing of interactions.

The proposed method, whose workflow is sketched in Figure 4.22, proceeds by first creating the *activities co-occurrence multigraph* (phase b) based on co-occurrence of activities within a specific session threshold between student pairs; it proceeds by merging overlapping activities (phase c) obtaining the *sessions multi-*

graph and finally applying community detection (phase e) in the students' *interaction graph* obtained by applying aggregation and weighting functions (phase d) to the sessions multigraph.

4.3.1 Interaction modelling

As described in Section 4.1.4 the student-system operations recorded in system logs are at *event* granularity level, i.e. they register the type and the beginning time of some event which require the VLE platform service, while it often omit data about detailed local user activities. For example, reading a file containing the course lecture notes will generate a singe event "*read*" to be recorded on the user logs, while all local user operations, such as browsing pages or terminating the application (e.g. closing a PDF reader). The user log information is used to determine the students co-occurrence of activities. This will lead to the identification of sessions that students spent often together when taking actions on the VLE.

This approach is motivated by the assumption that co-occurrence of activities of two students can be the emergent manifestation of some hidden communications that took place between the students to agree on the activities. Hence, in order to hypothesise a communication between students, the activities should happen in a time window, i.e. a *session*, as defined in Section 4.1.3.

We have decided to use graph as the natural representation for this kind of information. Students will be represented as nodes, and labelled links between them will express theco-occurrences of activity, i.e. the existence of hypotetic communication events between them.

The student's communities can then be elicited from this interaction graph weighted with the actual duration of co-occurrence of activities between the students.

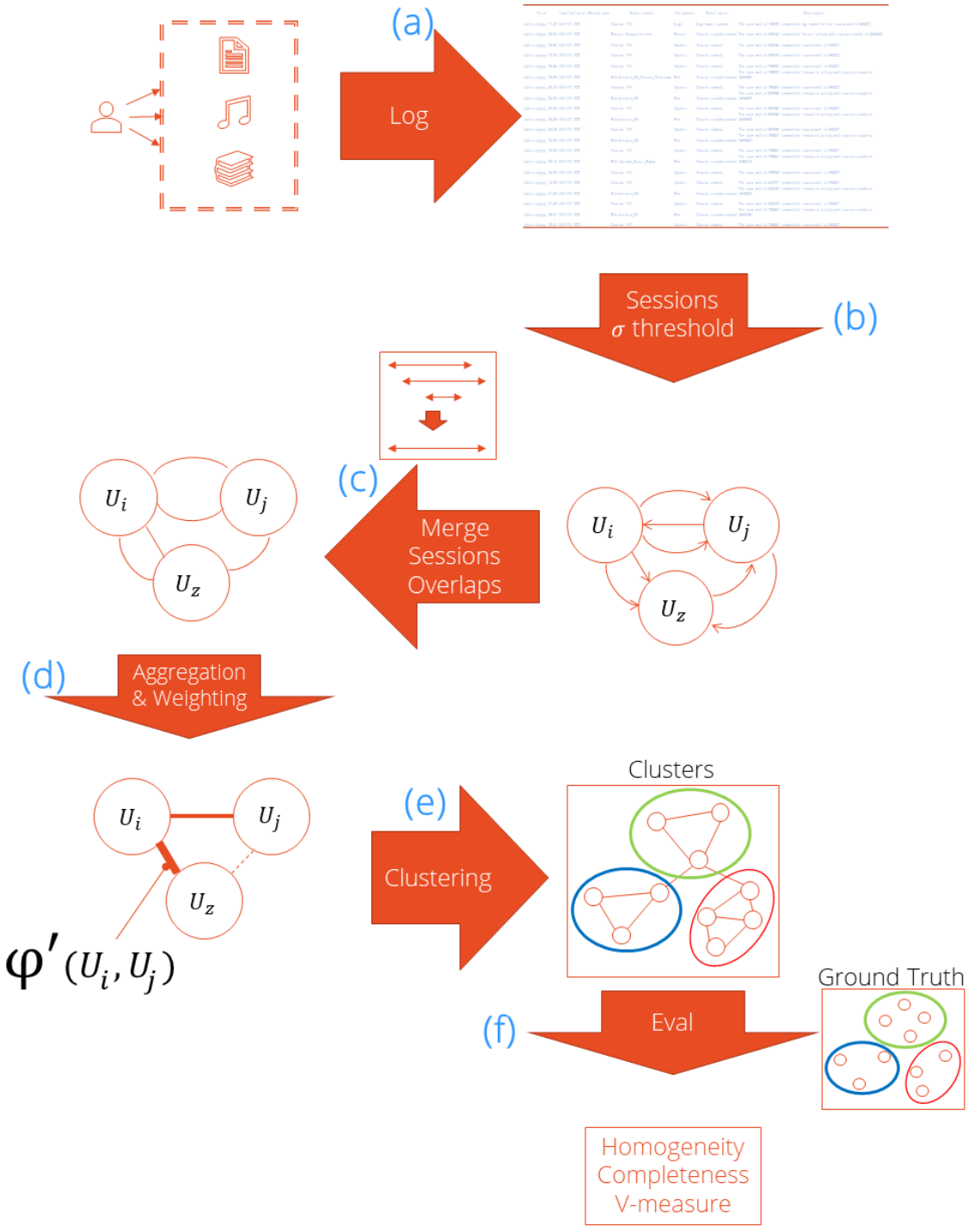

Figure 4.22: Visual scheme of the Session Based Analysis approach.

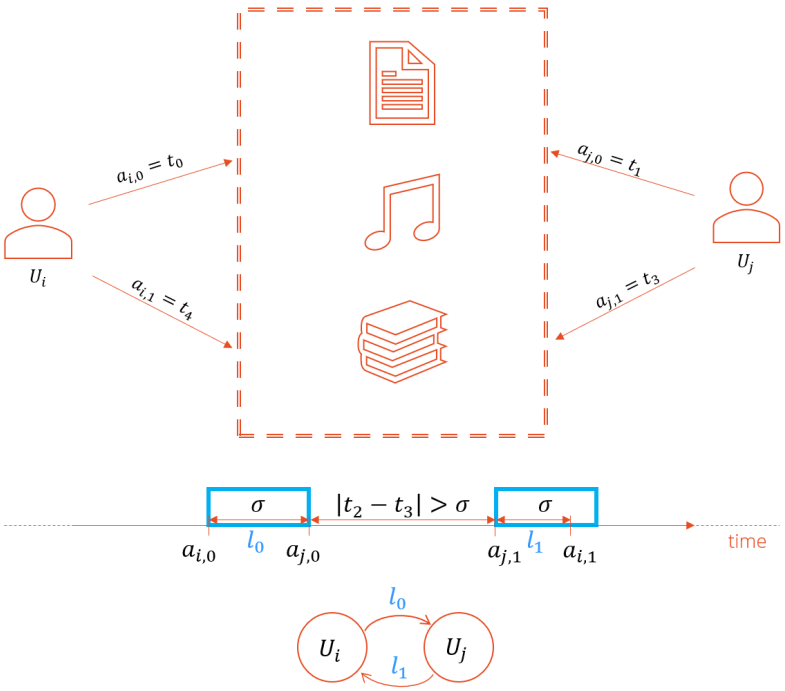

Figure 4.23: Link creation in Activities co-occurrence multigraph.

4.3.2 Activities co-occurrence graph

The information about timing of activities that different students took within the VLE is used to create the *activities co-occurrence multigraph*. Each student is represented by a node in the graph. The links between each students' pair are created by analysing the VLE logs. A link is created if the distance, in time, between each pair of activities of different students is falling inside a session threshold σ. As exemplified in Figure 4.23, a link in the *activities co-occurrence graph* is created when the distance between the timing of activities (e.g. $a_{i,0}$, $a_{i,1}$, $a_{i,0}$, $a_{j,1}$) of students U_i and U_j is less or equal that the σ threshold.

The *session threshold* σ is the basic parameterwhich represents the *session* concept. It identifies and separates the different activities that the students take together on the VLE.

The links created in the multigraph are labelled with starting and ending times, as well as with the effective time difference between the activities. Starting time is the time of the earliest activity in the session (e.g. $a_{i,0}$), ending time is the time of the latest activity in the session logs (e.g. $a_{i,1}$). Links are directional starting from the user whose actions on the system are earlier than the other one.

4.3.3 Sessions graph

Starting from the *activities co-occurrence multigraph*, the overlapping activities are merged to obtain sessions, as depicted in phase c of Figure 4.22.

First the links between each node pair are ordered with respect to their time labels, then the links are merged through an iterative *merging closure* process. All the activities co-occurrence links are initially turned into *session links*, then two links between the same pair of student nodes are merged when their time intervals have a relationship of *inclusion* or they are overlapping. The merging closure is calculated, i.e. the process is iterated in the graph, until no further merge is possible. The duration label of merged links is given by the difference between the earliest *activity start* and the latest *activity end* over all the activities merged. Note that this duration can exceed the σ session threshold parameter value. In Figure 4.24 is shown the process of aggregation of sessions to create links in the *Sessions multigraph*. Starting from the links in the *Activities multigraph*, the process finds the overlapping activities and merges them in sessions and computes the session link weight.

The *sessions graph* is still a multigraph as multiple sessions, i.e. not merging links, can exist between each students' pair at the end of the merging closure process.

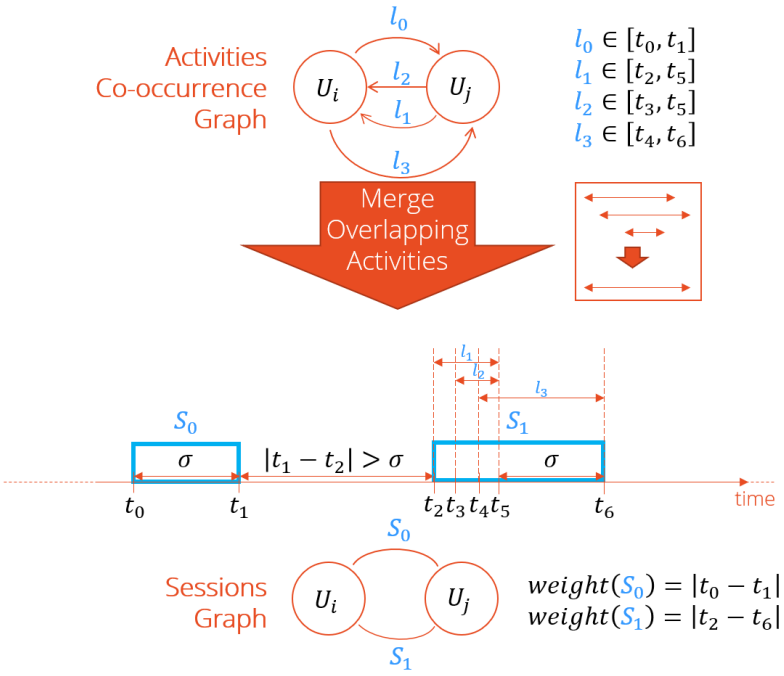

Figure 4.24: Link creation in Sessions multigraph.

4.3.4 Interactions graphs

The *interaction graph* is the actual graph which is used to elicit students' communities using network clustering algorithms. The *interaction graph* is a simple undirected graph and is generated by summarising by aggregation of the session multigraph. Different *interactions graphs* can be obtained using different aggregation functions to the links existing on the *sessions graph*.

The aggregation functions φ' based on the sessions number and time duration between each two students connected by a link. The weight on the edges can be determined using φ' as the *count* of interactions, other reasonable possibile φ' are *sum*, *max*, *min*, and *mean* aggregation functions. The aggregation process is shown in Figure 4.25.

Each *interactions graph* obtained with one of the different aggregation functions

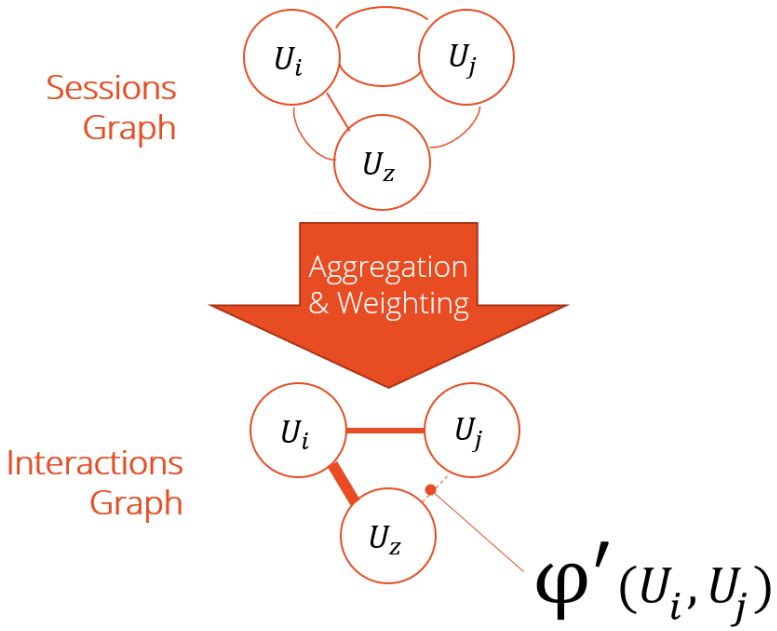

Figure 4.25: Link creation in Interactions graph.

is a simple graph. The computation of aggregation functions reduces to a single link the multiple links found between each student pair on the *sessions multigraph*.

4.3.5 Community Detection by Graph Analysis

Given a *students interaction* graph, the goal is to detect all the groups of students in it. Our approach is general as it can use various techniques to elicit the students' communities.

The community detection problem has been solved using various algorithms as introduced in Section 2.1.3. In this work we are using the *Modularity Maximization* approach 2.1.3, specifically with the *Louvain Method*. This method makes use of the weight on the edges to elicit the social communities. Using different weighted graphs, different modularity class assignments are detected.

4.3.6 Experiments and Evaluation Criteria

Experiments have been held using the dataset described in Section 4.1.1, extracting from it information about students' activity time and anonymised student IDs. The approach has been tested at course abstraction level. Students' co-occurrence of activities is computed by setting the session parameter σ at 3600 seconds (1 hour). The resulting graphs are then analysed to elicit the communities of students. The resulting social communities are compared with the ground truth groupings of students.

The modularity class label assignments have been computed for each course using the different aggregation functions *count*, *sum*, *max*, *min*, and *mean* to create the interactions graphs.

The evaluation criteria for cluster labellings has been introduced in Section 4.1.6. Homogeneity, Completeness and V-measure scores are computed to evaluate the quality of the retrieved social communities when compared to the ground truth.

4.3.7 Session Based Analysis Discussion

Starting from the interactions graph, the modularity class assignments have been computed. Figure 4.26 shows the interactions graph resulting from the analysis of course C1 using the , the *sum* of session timings as aggregation function. The graph is almost complete as it has 3866 links between the 89 students, the complete graph should have 3916 links hence only 50 links are missing. In this figure, the node colour represents the modularity class assignment and the node size represents its betweenness centrality measure. Basing on the latter, bigger nodes can be identified as leaders in the groups while smaller ones are followers. The visual representation is made by aggregating the followers of each community close to their leaders.

Figure 4.27 shows the interactions graph of course C2, as it results from the

Figure 4.26: Interactions graph of course C1 based on φ'_{sum} aggregation function.

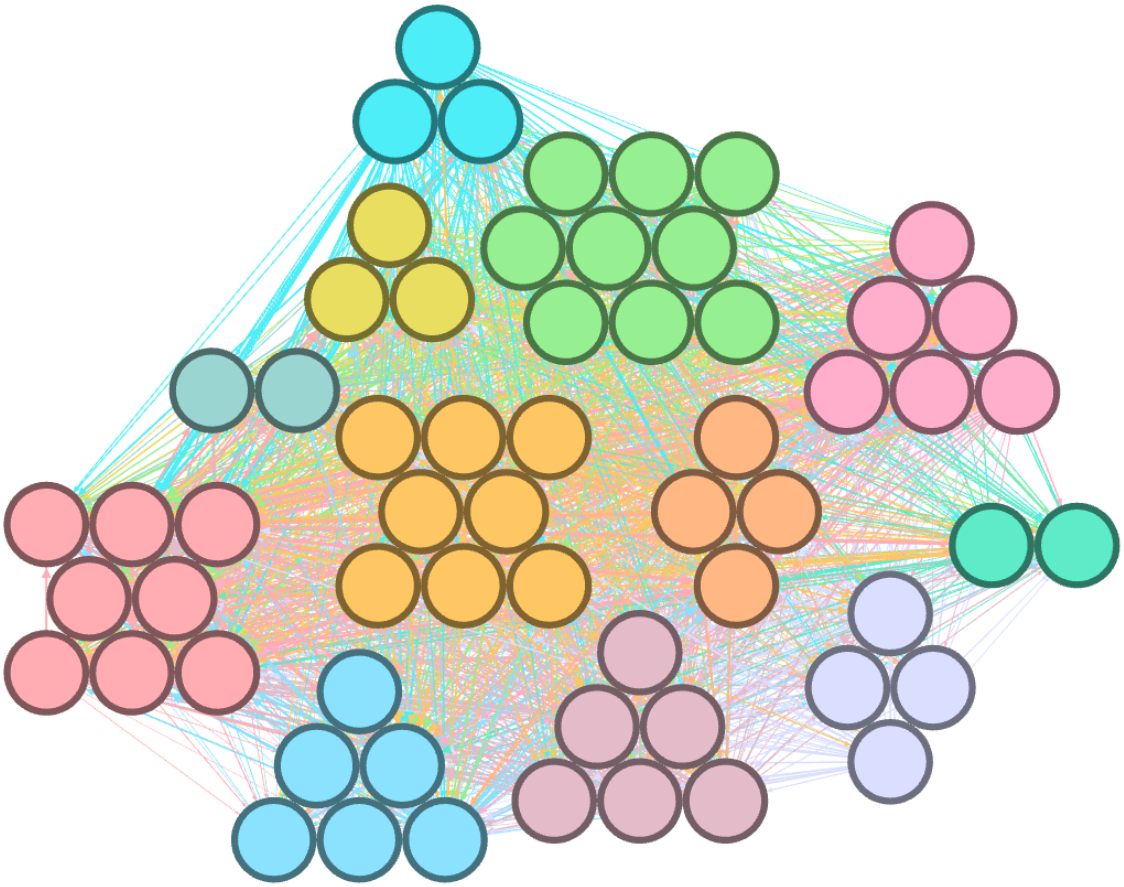

Figure 4.27: Interactions graph of course C2 based on φ'_{sum} aggregation function.

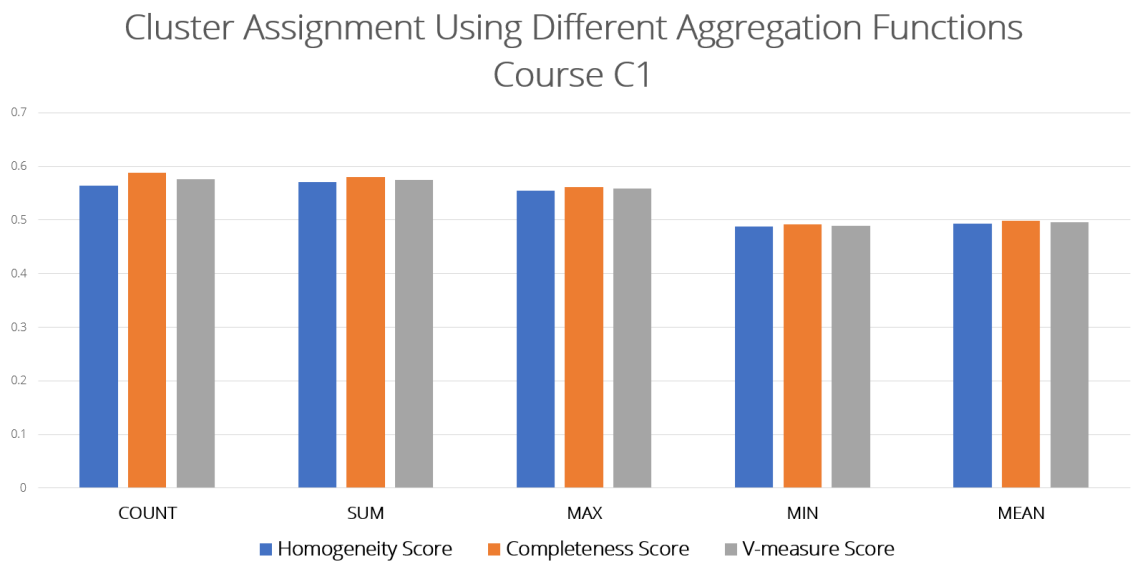

Figure 4.28: Homogeneity, Completeness and V-Measure of modularity class assignments on course C1.

aggregation of links from sessions graph using φ'_{sum}. In this case the users cannot be differentiated between leaders and followers using the betweenness centrality measure, as the graph is complete. In fact, for complete graphs the betweenness centrality measure is null for all the nodes. The node colours represent the modularity class assignment computed using our approach.

The performance in detection of student social communities, using the modularity class assignments, has been evaluated using the criteria and the measures described in Section 4.1.6. The Homogeneity, Completeness and V-measure scores for each of the aggregation functions computed on course C1 are represented in Figure 4.28 and Table 4.3. As shown, the best performance in terms of V-measure is obtained by counting the number of sessions shared by each students pair. Using this aggregation function the completeness score achieves the best results. The sum of all the sessions graph's links duration brings to achieve the best homogeneity score, thus creating communities that contain only students from the same ground

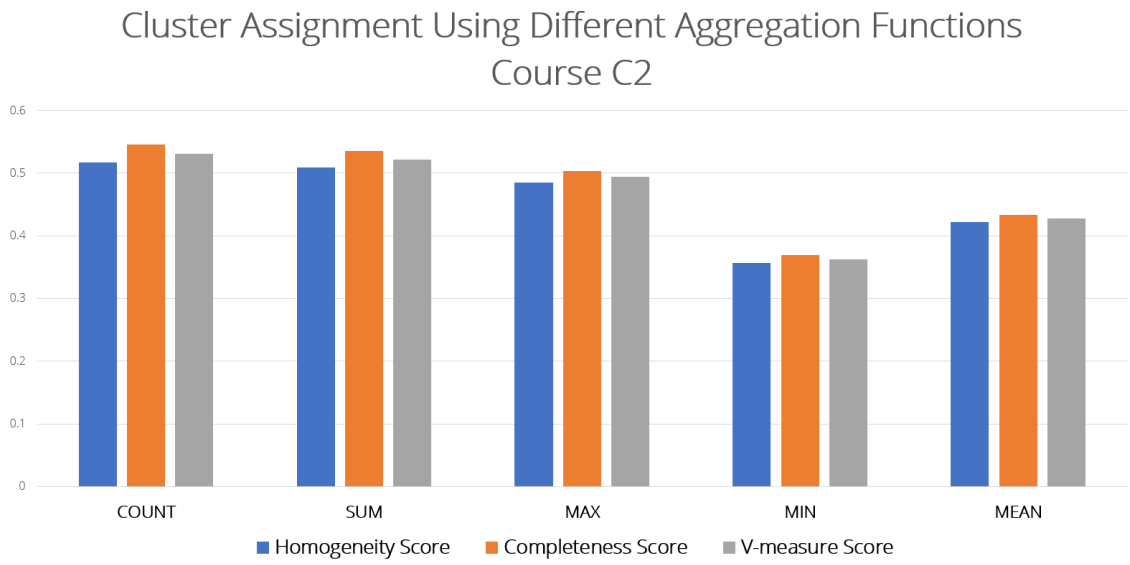

Figure 4.29: Homogeneity, Completeness and V-Measure of modularity class assignments on course C2.

truth group. In general the scores of *count*, *sum* and *max* aggregation functions are very close to each other.

The scores for course C2 community detection are represented in Figure 4.29 and Table 4.4. The results confirm the patterns already found in course C1 and the same considerations apply. *Count*, *sum* and *max* aggregation functions bring very similar results with slightly better ones for the counting of students sessions.

Table 4.3: Experiment's results using different aggregation functions for Session Based Analysis on C1 course.

Aggregation Function	Homogeneity Score	Completeness Score	V-measure Score
COUNT	0.563946972	**0.588506856**	**0.575965217**
SUM	**0.570671662**	0.579791394	0.575195382
MAX	0.555074718	0.561344604	0.558192055
MIN	0.487618224	0.49199713	0.48979789
MEAN	0.493767447	0.498787509	0.496264783

Table 4.4: Experiment's results using different aggregation functions for Session Based Analysis on C2 course.

Aggregation Function	Homogeneity Score	Completeness Score	V-measure Score
COUNT	**0.517577634**	**0.545905643**	**0.531364352**
SUM	0.509247829	0.535132634	0.521869456
MAX	0.485035688	0.503677624	0.494180912
MIN	0.35675456	0.369206826	0.362873897
MEAN	0.421874103	0.433565858	0.427640082

Chapter 5

Experimental Results Evaluation

Experimental results obtained using the different approaches introduced in this thesis have been shown in previous Section 4.2.7 and Section 4.3.7. The *History Based Correlation Analysis* and *Session Based Analysis* have both excellent performance in eliciting the social communities starting from the activities of students.

The results of the two different approaches differ slightly when we compare the social communities elicitation performances.

As shown in Figure 5.1 for course C1 the Homogeneity, Completeness, and V-measure score differences between the History based and Session based approaches are within 5%. Similar results are shown in Figure 5.2 for course C2.

The social communities elicited using our approaches are composed of evenly distributed number of components. As shown in Table 5.1, the standard deviation between the communities is low meaning that the number of students is similar in each of the groups. The variance is high, especially for course C1 using the Session Based approach. Going deeper in the analysis we found that only one of the groups is composed of a bigger number of components compared to the others.

The low variance and limited number of big communities, in combination with the evaluation scores, are evidence of the quality of the approaches introduced in

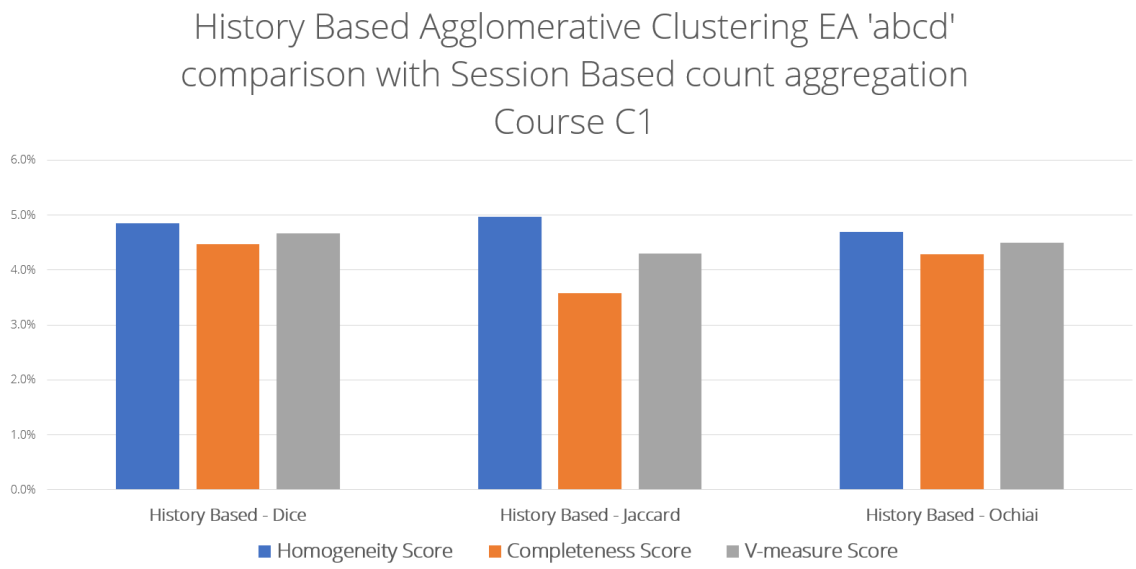

Figure 5.1: Homogeneity, Completeness and V-score differences in percentage between History Based approach best measures and Session Based approach count aggregation on course C1.

this thesis.

Hystory Based Correlation Analysis approach is computationally faster but the trade-off is that some details of the activities and interactions are lost in features extraction, i.e. binary summarisation which reduces co-occurrence information to uniform time slots. On the other hand, the Session Based Analysis approach can maintain a deeper level of detail about the actions and interactions, but it is computationally slower for the extremely high number of links created in the Activities co-occurrence graph.

History Based approach has, in general, better evaluation in Homogeneity, Completeness and V-measure scores as well as slightly better quality social communities when compared with the Session Based approach. In fact, Homogeneity of the found solutions is increasing more than the Completeness.

Analysing in detail the results we have found that the better Homogeneity score is

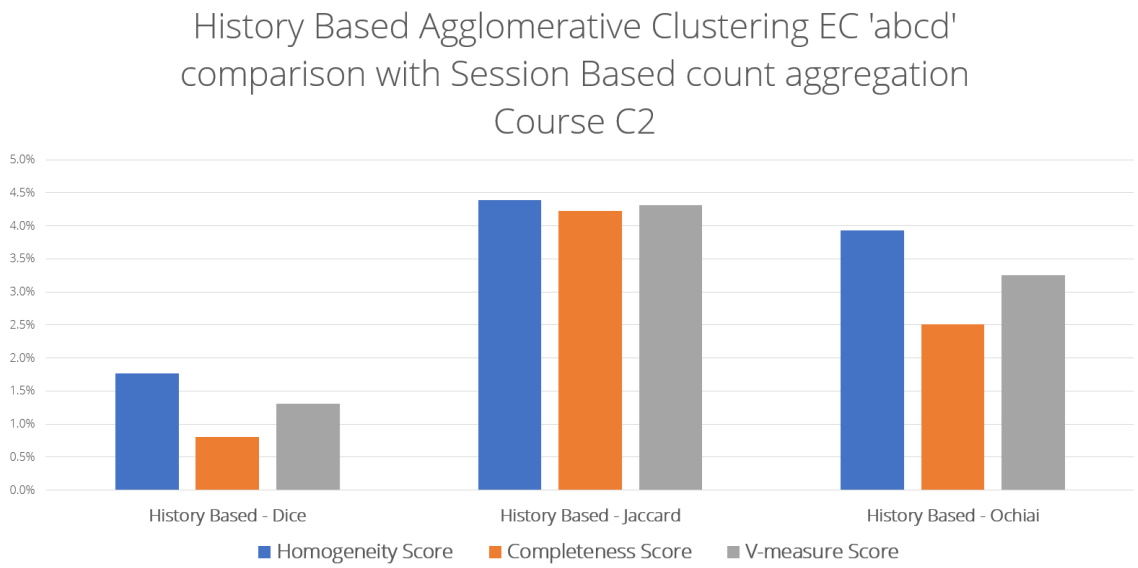

Figure 5.2: Homogeneity, Completeness and V-score differences in percentage between History Based approach best measures and Session Based approach count aggregation on course C2.

Table 5.1: Community size analysis on course C1 and C2 using the best strategies for each approach.

Course	Approach	Average Size	Variance	Standard Deviation
C1	History Based	4.68	5.78	2.40
C1	Session Based	5.08	11.10	3.33
C2	History Based	4.52	6.81	2.61
C2	Session Based	5.08	7.90	2.81

caused by the different assignments of students to groups. More groups found using the History Based approach contain only students coming from the same groups in the ground truth, as shown in the sample clustering assignment in Table 5.2.

The lower increase in Completeness score, which is also connected to the variance and standard deviation of the community sizes, is influenced by the fact that a ground truth group is split among different assignments. As shown in Table 5.3, the groups in the Ground Truth can be split in different clusters.

The Completeness score favours bigger groups, where all the group members of the ground truth communities are assigned to the same group. On the opposite the Homogeneity score favours smaller groups, even when the ground truth group is split among different groups. Which feature is preferable is dependent on the problem. In general the increase of both features is important but it is a difficult task. The V-measure score reflects a good trade-off between the two values to evaluate the quality of an assignment. In our results we can see that the increase in Homogeneity leads to a similar increase for V-measure score thus improving the general quality of the solution.

Using the Session Based approach we can elicit social communities that have comparable qualitative performance with the History Based approach, while retaining the additional information which can be further analysed from the graph representation in order to elicit social roles of nodes (e.g. leaders and followers).

Table 5.2: Sample group assignment where a cluster is formed by elements from two different Ground Truth groups.

UserID	Cluster	GroundTruth
id82102	11	5
id75417	11	13
id82847	11	13
id82525	11	13

Table 5.3: Sample group assignment where a Ground Truth group is split between two different clusters.

UserID	Cluster	GroundTruth
id75417	11	13
id82847	11	13
id82525	11	13
id62649	15	13

Conclusions

In this thesis we introduced innovative approaches to elicit hidden communities by observing user activities. Starting from the analysis of the co-occurrence of activities and actions taken by individuals, we can elicit their underlying social relationships. The social communities discovery approaches introduced in this thesis use indirect information about the social communication channels and links between people.

The first approach we introduced exploits the activity histories similarity to elicit the hidden social communities. The similarity measures performances have been augmented by the introduction of novel entropy amplification factors. This helps the similarity measurement to take in account additional information to contextualise the activities.

We further investigated the problem by using the session co-occurrence information and graph based representation. In this approach the graph features are used to discover the social communities and retrieve additional information based on network based measures.

Experimental results on real world dataset demonstrate that it is possible to discover social communities basing only on the observation of individuals' behaviour. An external observer, not involved in the communications and looking only at the explicit and observable activities of the individuals, can discover the underlying social communities. From experimental evidence, the social communities elicited using our approaches are an emerging phenomenon of activities co-occurrence and

side channel communications. The results indicate that our approaches could not only retrieve social communities from real world activities information, but also attain excellent precision in terms of Homogeneity, Completeness and V-measure scores when compared with the communities in the ground truth.

Community detection from implicit interactions is a challenging and open ended research topic. The need of community detection techniques from various sectors of the society requires continuous improvements.

New approaches are in development taking in account the timings of users' activities to elicit the underlying social communities. This new works will exploit information diffusion based approaches and novel techniques to characterise the user behaviour embedding. Evolution and dynamics of communities over time are considered to be explored in future works. This will bring new and deeper understandings on the interactions between the individuals.

Bibliography

[1] L. A. Adamic and E. Adar. Friends and neighbors on the Web. *Social Networks*, 2003.

[2] E. Adar, L. Zhang, L. Adamic, and R. Lukose. Implicit structure and the dynamics of blogspace. *HP Information Dynamics Lab*, 2004.

[3] R. M. Anderson and R. M. May. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1992.

[4] V. Arnaboldi, M. Conti, M. La Gala, A. Passarella, and F. Pezzoni. Ego network structure in online social networks and its impact on information diffusion. *Computer Communications*, 2016.

[5] K. E. Arnold, M. D. Pistilli, and K. E. Arnold. Course Signals at Purdue: Using Learning Analytics to Increase Student Success. *2nd International Conference on Learning Analytics and Knowledge*, (May):2–5, 2012.

[6] A. D. Athanassopoulos. Customer satisfaction cues to support market segmentation and explain switching behavior. *Journal of Business Research*, 2000.

[7] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[8] M. Bailly-Bechet, C. Borgs, A. Braunstein, J. Chayes, A. Dagkessamanskaia, J.-M. Francois, and R. Zecchina. Finding undetected protein associations in cell signaling by belief propagation. *Proceedings of the National Academy of Sciences*, 108(2):882–887, 2011.

[9] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic. The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*, pages 519–528. ACM, 2012.

[10] Q. Bao, W. K. Cheung, and J. Liu. Inferring motif-based diffusion models for social networks. In *IJCAI*, pages 3677–3683, 2016.

[11] Q. Bao, W. K. Cheung, Y. Zhang, and J. Liu. A component-based diffusion model with structural diversity for social networks. *IEEE Trans. Cybernetics*, 47(4):1078–1089, 2017.

[12] O. Barkan and N. Koenigstein. ITEM2VEC: Neural item embedding for collaborative filtering. In *IEEE International Workshop on Machine Learning for Signal Processing, MLSP*, 2016.

[13] A. Bavelas. Communication Patterns in Task-Oriented Groups. *The Journal of the Acoustical Society of America*, 1950.

[14] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A Neural Probabilistic Language Model. In *Journal of Machine Learning Research*, 2003.

[15] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), 2008.

[16] B. Bollobás. *Modern graph theory*, volume 184. Springer Science & Business Media, 2013.

[17] K. Bontcheva and D. Rout. Making sense of social media streams through semantics: a survey. *Semantic Web*, 5(5):373–403, 2014.

[18] U. Brandes, D. Delling, M. Gaertler, R. Goerke, M. Hoefer, Z. Nikoloski, and D. Wagner. Maximizing Modularity is hard. (001907), 2006.

[19] C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

[20] A. Budanitsky and G. Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and other lexical resources*, volume 2, pages 2–2, 2001.

[21] C. Castelfranchi. Modeling social action for AI agents. In *IJCAI International Joint Conference on Artificial Intelligence*, 1997.

[22] C. Castellano, F. Cecconi, V. Loreto, D. Parisi, and F. Radicchi. Self-contained algorithms to detect communities in networks. In *European Physical Journal B*, 2004.

[23] J. Castellini, V. Poggioni, and G. Sorbi. Fake Twitter followers detection by denoising autoencoder. In *Proceedings of the International Conference on Web Intelligence - WI '17*, 2017.

[24] L. D. Catledge and J. E. Pitkow. Characterizing browsing strategies in the World-Wide web. *Computer Networks and ISDN Systems*, 1995.

[25] M. A. Chatti, A. L. Dyckhoff, U. Schroeder, and H. Thüs. A reference model for learning analytics. *International Journal of Technology Enhanced Learning*, 4(5/6):318, 2012.

[26] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, J. Sun, and S. Health. Gram : Graph-based attention model for healtcare representation learning. *ICLR*, 2017.

[27] D. Clark. Moodle Activity Viewer (MAV), 2013.

[28] M. de Raadt. Block Heatmap, 2016.

[29] A. Dempster, N. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B Methodological*, 1977.

[30] L. R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, 1945.

[31] F. K. Diekert. The tragedy of the commons from a game-theoretic perspective. *Sustainability*, 2012.

[32] M. Doebeli and C. Hauert. Models of cooperation based on the Prisoner's Dilemma and the Snowdrift game, 2005.

[33] W. E. Donath and A. J. Hoffman. Lower Bounds for the Partitioning of Graphs. *IBM Journal of Research and Development*, 1973.

[34] M. Fei and D.-Y. Yeung. Temporal Models for Predicting Student Dropout in Massive Open Online Courses. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 256–263. IEEE, nov 2015.

[35] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[36] L. R. J. Ford and D. R. Fulkerson. Maximial flow through a newtork. *Can J. Math*, 1956.

[37] S. Fortunato. Community detection in graphs, 2010.

[38] E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

[39] V. Franzoni, Y. Li, and P. Mengoni. A path-based model for emotion abstraction on facebook using sentiment analysis and taxonomy knowledge. *Proceedings of the International Conference on Web Intelligence - WI '17*, pages 947–952, 2017.

[40] V. Franzoni, Y. Li, P. Mengoni, and A. Milani. Clustering facebook for biased context extraction. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10404, pages 717–729. 2017.

[41] V. Franzoni, M. Mencacci, P. Mengoni, and A. Milani. Semantic Heuristic Search in Collaborative Networks: Measures and Contexts. In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1, pages 141–148. IEEE, aug 2014.

[42] L. C. Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 1977.

[43] M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences). *Computers and Intractability*, 1979.

[44] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[45] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring Networks of Diffusion and Influence. *ACM Transactions on Knowledge Discovery from Data*, 5(4):1–37, feb 2012.

[46] R. Guimerà, M. Sales-Pardo, and L. A. Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 2004.

[47] M. S. Handcock, A. E. Raftery, and J. M. Tantrum. Model-Based Clustering for Social Networks. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 2007.

[48] G. Hardin. The Tragedy of the Unmanaged Commons. In *Evolutionary Perspectives on Environmental Problems*. 2009.

[49] L. M. Hogg and N. R. Jennings. Socially intelligent reasoning for autonomous agents. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, 2001.

[50] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[51] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. ImprovingWord Representations via Global Context and MultipleWord Prototypes. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 2012.

[52] P. Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37(JANUARY 1901):547–579, 1901.

[53] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[54] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, 2003.

[55] B. W. Kernighan and S. Lin. An Efficient Heuristic Procedure for Partitioning Graphs. *Bell System Technical Journal*, 1970.

[56] T. Killingback, M. Doebeli, and C. Hauert. Diversity of cooperation in the tragedy of the commons. *Biological Theory*, 5(1):3–6, 2010.

[57] R. A. Kolvoord. Visual Insights: A Practical Guide to Making Sense of Data. *Online Information Review*, 38(7):994–995, nov 2014.

[58] J. A. Konstan and J. Riedl. Recommender systems: From algorithms to user experience, 2012.

[59] A. D. Kramer. The spread of emotion via facebook. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 767–770. ACM, 2012.

[60] B. Krishnamurthy and J. Wang. On network-aware clustering of Web clients. *ACM SIGCOMM Computer Communication Review*, 2000.

[61] J. M. Kumpula, J. Saramäki, K. Kaski, and J. Kertész. Limited resolution in complex network community detection with Potts model approach. *The European Physical Journal B*, 2007.

[62] S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent Convolutional Neural Networks for Text Classification. *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[63] C. Leacock and M. Chodorow. Combining Local Context and WordNet Similarity for Word Sense Identification. *In: WordNet: An electronic lexical database.*, 1998.

[64] J. Leskovec and J. J. Mcauley. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, pages 539–547, 2012.

[65] S. P. Lloyd. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 1982.

[66] L. López, G. Del Rey Almansa, S. Paquelet, and A. Fernández. A mathematical model for the TCP Tragedy of the Commons. *Theoretical Computer Science*, 2005.

[67] D. J. C. Mackay. Information Theory , Inference , and Learning Algorithms. *Learning*, 2003.

[68] J. Macqueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[69] D. Mehrzadi and D. G. Feitelson. On extracting session data from activity logs. In *Proceedings of the 5th Annual International Systems and Storage Conference*, SYSTOR '12, pages 3:1–3:7, New York, NY, USA, 2012. ACM.

[70] T. Mikolov, G. Corrado, K. Chen, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, 2013.

[71] T. Mikolov, J. Kopecky, L. Burget, O. Glembek, and J. Cernocky. Neural network based language models for highly inflective languages. *Icassp-2009*, 2009.

[72] G. A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 1995.

[73] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 2004.

[74] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 2006.

[75] A. Y. Ng, M. I. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and Algorithm. *Adv. Neural Inf. Process. Syst.*, 2001.

[76] A. OCHIAI. Zoogeographical Studies on the Soleoid Fishes Found in Japan and its Neighhouring Regions-II. *NIPPON SUISAN GAKKAISHI*, 22(9):526–530, 1957.

[77] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. *World Wide Web Internet And Web Information Systems*, 1998.

[78] K. Pearson. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London (1854-1905)*, 1895.

[79] T. P. Peixoto. Efficient Monte Carlo and greedy heuristic for the inference of stochastic block models. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 89(1), 2014.

[80] S. Poria, E. Cambria, and A. Gelbukh. Aspect Extraction for Opinion Miningwith a Deep Convolutional Neural Network. *Knowledge-Based Systems*, 2016.

[81] M. Ramezani, A. Khodadadi, and H. R. Rabiee. Community detection using diffusion information. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(2):20, 2018.

[82] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[83] C. J. Rhodes and E. M. J. Keefe. Social network topology: a Bayesian approach. *The Journal of the Operational Research Society*, 2007.

[84] C. V. Rijsbergen. Information Retrieval. *Information Retrieval*, page 208, 1979.

[85] S. G. Roberts, R. I. Dunbar, T. V. Pollet, and T. Kuppens. Exploring variation in active network size: Constraints and ego characteristics. *Social Networks*, 2009.

[86] M. G. Rodriguez, J. Leskovec, B. Schoelkopf, M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. Modeling information propagation with survival theory. In *International Conference on Machine Learning*, pages 666–674, 2013.

[87] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language (EMNLP-CoNLL'07)*, 1(June):410–420, 2007.

[88] P. F. Rusell and T. R. Rao. On Habitat and Association of Species of Anopheline Larvae in South-Eastern Madras. *Journal of the Malaria Institute of India*, 3(1):153–178, 1940.

[89] S. Saha and S. Sen. Local decision procedures for avoiding the tragedy of commons. *Distributed Computing: Iwdc 2003*, 2003.

[90] B. A. Schwendimann, M. J. Rodriguez-Triana, A. Vozniuk, L. P. Prieto, M. S. Boroujeni, A. Holzer, D. Gillet, and P. Dillenbourg. Perceiving Learning at a Glance: A Systematic Literature Review of Learning Dashboard Research. *IEEE Transactions on Learning Technologies*, 10(1):30–41, jan 2017.

[91] O. Sen and S. Sen. Averting the tragedy of the commons by adapting aspiration levels. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012.

[92] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[93] P. H. A. Sneath and R. R. Sokal. The Principles and Practice of Numerical Taxonomy. *Taxon*, 12(5):190, 1963.

[94] C. Strapparava and R. Mihalcea. SemEval-2007 Task 14: Affective Text. In *Proceedings of the International Workshop on Semantic Evaluations (SemEval-2007)*, 2007.

[95] P. R. Suaris and G. Kedem. An Algorithm for Quadrisection and Its Application to Standard Cell Placement. *IEEE Transactions on Circuits and Systems*, 1988.

[96] M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment in twitter events. *Journal of the American Society for Information Science and Technology*, 62(2):406–418, 2011.

[97] J. Turian, L. Ratinov, Y. Bengio, and J. Turian. Word Representations: A Simple and General Method for Semi-supervised Learning. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010.

[98] R. M. Turner. *The tragedy of the commons and distributed AI systems*. Department of Computer Science, University of New Hampshire, 1993.

[99] K. Verbert, S. Govaerts, E. Duval, J. L. Santos, F. Van Assche, G. Parra, and J. Klerkx. Learning dashboards: an overview and future research opportunities. *Personal and Ubiquitous Computing*, 18(6):1499–1514, nov 2013.

[100] Y. Wang, H. Shen, S. Liu, J. Gao, and X. Cheng. Cascade dynamics modeling with attention-based recurrent neural network. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2985–2991. AAAI Press, 2017.

[101] G. Weikum. Foundations of statistical natural language processing. *ACM SIGMOD Record*, 2002.

[102] I. H. Witten and D. N. Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. 2008.

[103] R. S. Wu and P. H. Chou. Customer segmentation of multiple category data in e-commerce using a soft-clustering approach. *Electronic Commerce Research and Applications*, 2011.

[104] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

[105] L. Yu, P. Cui, C. Song, T. Zhang, and S. Yang. A Temporally Heterogeneous Survival Framework with Application to Social Behavior Dynamics. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17*, 2017.

[106] G. U. Yule. On the Methods of Measuring Association Between Two Attributes. *Journal of the Royal Statistical Society*, 1912.

Appendix A

Histories Binary Correlation Results

Table A.1: Complete Experiments Results for History Binary Analysis on C1 course.

Experiment	Homogeneity Score	Completeness Score	V-measure Score
cooccurrence AC none C Agglomerative	0.3793	0.4978	0.4306
cooccurrence AC none C GaussianMixture	0.4345	0.5595	0.4892
cooccurrence AC none C KMeans	0.4143	0.5216	0.4618
cooccurrence AC none C Spectral	0.3422	0.4526	0.3897
cooccurrence AC ad C Agglomerative	0.2694	0.5872	0.3693
cooccurrence AC ad C GaussianMixture	0.2329	0.4736	0.3122
cooccurrence AC ad C KMeans	0.3052	0.5167	0.3838
cooccurrence AC ad C Spectral	0.2032	0.5479	0.2965
cooccurrence AC abcd C Agglomerative	0.2694	0.5871	0.3693
cooccurrence AC abcd C GaussianMixture	0.2329	0.4736	0.3122
cooccurrence AC abcd C KMeans	0.3052	0.5167	0.3838

Experiment	Homogeneity Score	Completeness Score	V-measure Score
cooccurrence AC abcd C Spectral	0.2188	0.5464	0.3125
cooccurrence AC a C Agglomerative	0.2694	0.5872	0.3693
cooccurrence AC a C GaussianMixture	0.2329	0.4736	0.3122
cooccurrence AC a C KMeans	0.3052	0.5167	0.3838
cooccurrence AC a C Spectral	0.2188	0.5464	0.3125
dice AC none C Agglomerative	0.5843	0.6026	0.5933
dice AC none C GaussianMixture	0.5886	0.6052	0.5967
dice AC none C KMeans	0.5442	0.5649	0.5543
dice AC none C Spectral	0.5624	0.5800	0.5711
dice AC ad C Agglomerative	0.6124	0.6332	0.6226
dice AC ad C GaussianMixture	0.5769	0.5842	0.5805
dice AC ad C KMeans	0.5643	0.5765	0.5703
dice AC ad C Spectral	0.6105	0.6209	0.6157
dice AC abcd C Agglomerative	0.6124	0.6332	0.6226
dice AC abcd C GaussianMixture	0.5769	0.5842	0.5805
dice AC abcd C KMeans	0.5643	0.5765	0.5703
dice AC abcd C Spectral	0.6105	0.6209	0.6157
dice AC a C Agglomerative	0.5923	0.6058	0.5990
dice AC a C GaussianMixture	0.5830	0.5868	0.5849
dice AC a C KMeans	0.5605	0.5772	0.5687
dice AC a C Spectral	0.5970	0.6146	0.6057
jaccard AC none C Agglomerative	0.6092	0.6259	0.6175
jaccard AC none C GaussianMixture	0.5777	0.5941	0.5858
jaccard AC none C KMeans	0.5358	0.5568	0.5461
jaccard AC none C Spectral	0.5679	0.5868	0.5772

Experiment	Homogeneity Score	Completeness Score	V-measure Score
jaccard AC ad C Agglomerative	0.6136	0.6242	0.6189
jaccard AC ad C GaussianMixture	0.5827	0.5918	0.5872
jaccard AC ad C KMeans	0.5458	0.5579	0.5518
jaccard AC ad C Spectral	0.5499	0.5913	0.5699
jaccard AC abcd C Agglomerative	0.6136	0.6242	0.6189
jaccard AC abcd C GaussianMixture	0.5827	0.5918	0.5872
jaccard AC abcd C KMeans	0.5458	0.5579	0.5518
jaccard AC abcd C Spectral	0.5499	0.5913	0.5699
jaccard AC a C Agglomerative	0.5978	0.6134	0.6055
jaccard AC a C GaussianMixture	0.5908	0.5939	0.5923
jaccard AC a C KMeans	0.5657	0.5826	0.5741
jaccard AC a C Spectral	0.6064	0.6276	0.6168
ochiai AC none C Agglomerative	0.5805	0.6006	0.5904
ochiai AC none C GaussianMixture	0.5888	0.5947	0.5917
ochiai AC none C KMeans	0.5446	0.5639	0.5540
ochiai AC none C Spectral	0.5333	0.5473	0.5402
ochiai AC ad C Agglomerative	0.6108	0.6313	0.6209
ochiai AC ad C GaussianMixture	0.5788	0.5838	0.5813
ochiai AC ad C KMeans	0.5571	0.5702	0.5636
ochiai AC ad C Spectral	0.5592	0.5735	0.5663
ochiai AC abcd C Agglomerative	0.6108	0.6313	0.6209
ochiai AC abcd C GaussianMixture	0.5788	0.5838	0.5813
ochiai AC abcd C KMeans	0.5571	0.5702	0.5636
ochiai AC abcd C Spectral	0.5592	0.5735	0.5663
ochiai AC a C Agglomerative	0.6376	0.6530	0.6452

Experiment	Homogeneity Score	Completeness Score	V-measure Score
ochiai AC a C GaussianMixture	0.5349	0.5421	0.5385
ochiai AC a C KMeans	0.5599	0.5813	0.5704
ochiai AC a C Spectral	0.5708	0.5860	0.5783
rr AC none C Agglomerative	0.3800	0.4982	0.4311
rr AC none C GaussianMixture	0.4319	0.5491	0.4835
rr AC none C KMeans	0.4143	0.5216	0.4618
rr AC none C Spectral	0.4684	0.5890	0.5218
rr AC ad C Agglomerative	0.2694	0.5872	0.3693
rr AC ad C GaussianMixture	0.2594	0.4802	0.3368
rr AC ad C KMeans	0.3003	0.4983	0.3747
rr AC ad C Spectral	0.3036	0.5063	0.3795
rr AC abcd C Agglomerative	0.2694	0.5872	0.3693
rr AC abcd C GaussianMixture	0.3101	0.4573	0.3696
rr AC abcd C KMeans	0.3052	0.5167	0.3838
rr AC abcd C Spectral	0.2859	0.5709	0.3810
rr AC a C Agglomerative	0.2694	0.5872	0.3693
rr AC a C GaussianMixture	0.2983	0.4687	0.3646
rr AC a C KMeans	0.3052	0.5167	0.3838
rr AC a C Spectral	0.2945	0.5955	0.3941
sm AC none C Agglomerative	0.3904	0.4798	0.4305
sm AC none C GaussianMixture	0.4803	0.5164	0.4977
sm AC none C KMeans	0.4817	0.5046	0.4928
sm AC none C Spectral	0.3896	0.5120	0.4425
sm AC ad C Agglomerative	0.3961	0.4847	0.4359
sm AC ad C GaussianMixture	0.4234	0.4886	0.4537

Experiment	Homogeneity Score	Completeness Score	V-measure Score
sm AC ad C KMeans	0.4702	0.5042	0.4866
sm AC ad C Spectral	0.3739	0.4912	0.4246
sm AC abcd C Agglomerative	0.3990	0.4877	0.4389
sm AC abcd C GaussianMixture	0.4305	0.5119	0.4677
sm AC abcd C KMeans	0.4701	0.5031	0.4860
sm AC abcd C Spectral	0.2510	0.4771	0.3290
sm AC a C Agglomerative	0.3904	0.4798	0.4305
sm AC a C GaussianMixture	0.4803	0.5164	0.4977
sm AC a C KMeans	0.4824	0.5050	0.4934
sm AC a C Spectral	0.3226	0.5071	0.3943
ss1 AC none C Agglomerative	0.3615	0.4671	0.4076
ss1 AC none C GaussianMixture	0.4045	0.4802	0.4391
ss1 AC none C KMeans	0.4852	0.5161	0.5002
ss1 AC none C Spectral	0.4799	0.5909	0.5297
ss1 AC ad C Agglomerative	0.3333	0.4579	0.3858
ss1 AC ad C GaussianMixture	0.3714	0.4674	0.4139
ss1 AC ad C KMeans	0.4577	0.5044	0.4799
ss1 AC ad C Spectral	0.4105	0.5582	0.4731
ss1 AC abcd C Agglomerative	0.3461	0.4810	0.4025
ss1 AC abcd C GaussianMixture	0.3433	0.5050	0.4088
ss1 AC abcd C KMeans	0.4281	0.4741	0.4499
ss1 AC abcd C Spectral	0.3400	0.5412	0.4176
ss1 AC a C Agglomerative	0.3976	0.5053	0.4451
ss1 AC a C GaussianMixture	0.4052	0.4728	0.4364
ss1 AC a C KMeans	0.4558	0.4925	0.4735

Experiment	Homogeneity Score	Completeness Score	V-measure Score
ss1 AC a C Spectral	0.4575	0.5648	0.5055

Table A.2: Complete Experiments Results for History Binary Analysis on C2 course.

Experiment	Homogeneity Score	Completeness Score	V-measure Score
cooccurrence AC none C Agglomerative	0.3748	0.4728	0.4181
cooccurrence AC none C GaussianMixture	0.3431	0.4662	0.3953
cooccurrence AC none C KMeans	0.3843	0.4780	0.4260
cooccurrence AC none C Spectral	0.3154	0.4389	0.3670
cooccurrence AC ad C Agglomerative	0.2192	0.4400	0.2926
cooccurrence AC ad C GaussianMixture	0.2095	0.4532	0.2865
cooccurrence AC ad C KMeans	0.2024	0.5226	0.2918
cooccurrence AC ad C Spectral	0.1874	0.5148	0.2748
cooccurrence AC abcd C Agglomerative	0.2192	0.4400	0.2926
cooccurrence AC abcd C GaussianMixture	0.2095	0.4532	0.2865
cooccurrence AC abcd C KMeans	0.2024	0.5226	0.2918
cooccurrence AC abcd C Spectral	0.2266	0.4340	0.2977
cooccurrence AC a C Agglomerative	0.2192	0.4400	0.2926
cooccurrence AC a C GaussianMixture	0.2095	0.4532	0.2865
cooccurrence AC a C KMeans	0.2024	0.5226	0.2918
cooccurrence AC a C Spectral	0.2266	0.4340	0.2977
dice AC none C Agglomerative	0.5150	0.5424	0.5283
dice AC none C GaussianMixture	0.5162	0.5229	0.5195
dice AC none C KMeans	0.5121	0.5365	0.5240
dice AC none C Spectral	0.5571	0.5841	0.5703
dice AC ad C Agglomerative	0.5411	0.5514	0.5462
dice AC ad C GaussianMixture	0.5276	0.5350	0.5313
dice AC ad C KMeans	0.5249	0.5369	0.5308

Experiment	Homogeneity Score	Completeness Score	V-measure Score
dice AC ad C Spectral	0.5412	0.5483	0.5447
dice AC abcd C Agglomerative	0.5410	0.5514	0.5462
dice AC abcd C GaussianMixture	0.5276	0.5350	0.5313
dice AC abcd C KMeans	0.5249	0.5369	0.5308
dice AC abcd C Spectral	0.5412	0.5483	0.5447
dice AC a C Agglomerative	0.5606	0.5856	0.5729
dice AC a C GaussianMixture	0.5218	0.5274	0.5246
dice AC a C KMeans	0.4894	0.5226	0.5054
dice AC a C Spectral	0.5352	0.5539	0.5444
jaccard AC none C Agglomerative	0.5033	0.5407	0.5213
jaccard AC none C GaussianMixture	0.5021	0.5147	0.5083
jaccard AC none C KMeans	0.5150	0.5551	0.5343
jaccard AC none C Spectral	0.5687	0.5741	0.5714
jaccard AC ad C Agglomerative	0.5456	0.5680	0.5566
jaccard AC ad C GaussianMixture	0.5321	0.5422	0.5371
jaccard AC ad C KMeans	0.5004	0.5103	0.5053
jaccard AC ad C Spectral	0.5704	0.5863	0.5782
jaccard AC abcd C Agglomerative	0.5456	0.5680	0.5566
jaccard AC abcd C GaussianMixture	0.5321	0.5422	0.5371
jaccard AC abcd C KMeans	0.5004	0.5103	0.5053
jaccard AC abcd C Spectral	0.5704	0.5863	0.5782
jaccard AC a C Agglomerative	0.5606	0.5856	0.5729
jaccard AC a C GaussianMixture	0.4725	0.4804	0.4764
jaccard AC a C KMeans	0.4948	0.5253	0.5096

Experiment	Homogeneity Score	Completeness Score	V-measure Score
jaccard AC a C Spectral	0.5614	0.5881	0.5744
ochiai AC none C Agglomerative	0.5171	0.5528	0.5344
ochiai AC none C GaussianMixture	0.5582	0.5703	0.56425
ochiai AC none C KMeans	0.5474	0.5756	0.5612
ochiai AC none C Spectral	0.5790	0.5942	0.5865
ochiai AC ad C Agglomerative	0.5669	0.5905	0.5785
ochiai AC ad C GaussianMixture	0.5109	0.5166	0.5138
ochiai AC ad C KMeans	0.5360	0.5471	0.5415
ochiai AC ad C Spectral	0.5693	0.5893	0.5791
ochiai AC abcd C Agglomerative	0.5669	0.5905	0.5785
ochiai AC abcd C GaussianMixture	0.5109	0.5166	0.5138
ochiai AC abcd C KMeans	0.5360	0.5471	0.5415
ochiai AC abcd C Spectral	0.5693	0.5893	0.5791
ochiai AC a C Agglomerative	0.5723	0.5948	0.5833
ochiai AC a C GaussianMixture	0.5791	0.5846	0.5818
ochiai AC a C KMeans	0.4882	0.5138	0.5007
ochiai AC a C Spectral	0.5568	0.5710	0.5638
rr AC none C Agglomerative	0.3748	0.4728	0.4181
rr AC none C GaussianMixture	0.3479	0.4816	0.4040
rr AC none C KMeans	0.3843	0.4780	0.4260
rr AC none C Spectral	0.4042	0.4982	0.4463
rr AC ad C Agglomerative	0.2321	0.4432	0.3046
rr AC ad C GaussianMixture	0.2078	0.4254	0.2792
rr AC ad C KMeans	0.1900	0.5219	0.2786

Experiment	Homogeneity Score	Completeness Score	V-measure Score
rr AC ad C Spectral	0.3288	0.4958	0.3954
rr AC abcd C Agglomerative	0.2192	0.4400	0.2926
rr AC abcd C GaussianMixture	0.1900	0.3364	0.2428
rr AC abcd C KMeans	0.2036	0.5257	0.2935
rr AC abcd C Spectral	0.2557	0.4793	0.3335
rr AC a C Agglomerative	0.2192	0.4400	0.2926
rr AC a C GaussianMixture	0.2031	0.3629	0.2604
rr AC a C KMeans	0.2024	0.5226	0.2918
rr AC a C Spectral	0.2557	0.4793	0.3335
sm AC none C Agglomerative	0.3095	0.3730	0.3383
sm AC none C GaussianMixture	0.3736	0.4153	0.3934
sm AC none C KMeans	0.3524	0.3980	0.3738
sm AC none C Spectral	0.3697	0.4934	0.4227
sm AC ad C Agglomerative	0.2779	0.3596	0.3135
sm AC ad C GaussianMixture	0.3410	0.3917	0.3646
sm AC ad C KMeans	0.3522	0.3860	0.3683
sm AC ad C Spectral	0.2798	0.3956	0.3278
sm AC abcd C Agglomerative	0.2814	0.3667	0.3184
sm AC abcd C GaussianMixture	0.3596	0.4087	0.3826
sm AC abcd C KMeans	0.3830	0.4182	0.3998
sm AC abcd C Spectral	0.2284	0.4583	0.3049
sm AC a C Agglomerative	0.3095	0.3730	0.3383
sm AC a C GaussianMixture	0.3736	0.4153	0.3934
sm AC a C KMeans	0.3524	0.3980	0.3738

Experiment	Homogeneity Score	Completeness Score	V-measure Score
sm AC a C Spectral	0.3697	0.4934	0.4227
ss1 AC none C Agglomerative	0.2638	0.3486	0.3003
ss1 AC none C GaussianMixture	0.3492	0.4119	0.3780
ss1 AC none C KMeans	0.4130	0.4520	0.4316
ss1 AC none C Spectral	0.5159	0.5656	0.5396
ss1 AC ad C Agglomerative	0.2644	0.3481	0.3005
ss1 AC ad C GaussianMixture	0.2823	0.3574	0.3154
ss1 AC ad C KMeans	0.3706	0.4076	0.3882
ss1 AC ad C Spectral	0.3650	0.5222	0.4297
ss1 AC abcd C Agglomerative	0.2258	0.3300	0.2682
ss1 AC abcd C GaussianMixture	0.2696	0.3640	0.3098
ss1 AC abcd C KMeans	0.3764	0.4155	0.3950
ss1 AC abcd C Spectral	0.3102	0.4941	0.3812
ss1 AC a C Agglomerative	0.2959	0.3725	0.3298
ss1 AC a C GaussianMixture	0.3882	0.4518	0.4176
ss1 AC a C KMeans	0.4270	0.4597	0.4428
ss1 AC a C Spectral	0.4159	0.4998	0.4540

Appendix B

Session Based Analysis Graph Representations

In this appendix the interaction graphs resulting from the computation of the different aggregation functions are reported.

Figure B.1: Interactions graph of course C1 based on *sum* aggregation function.

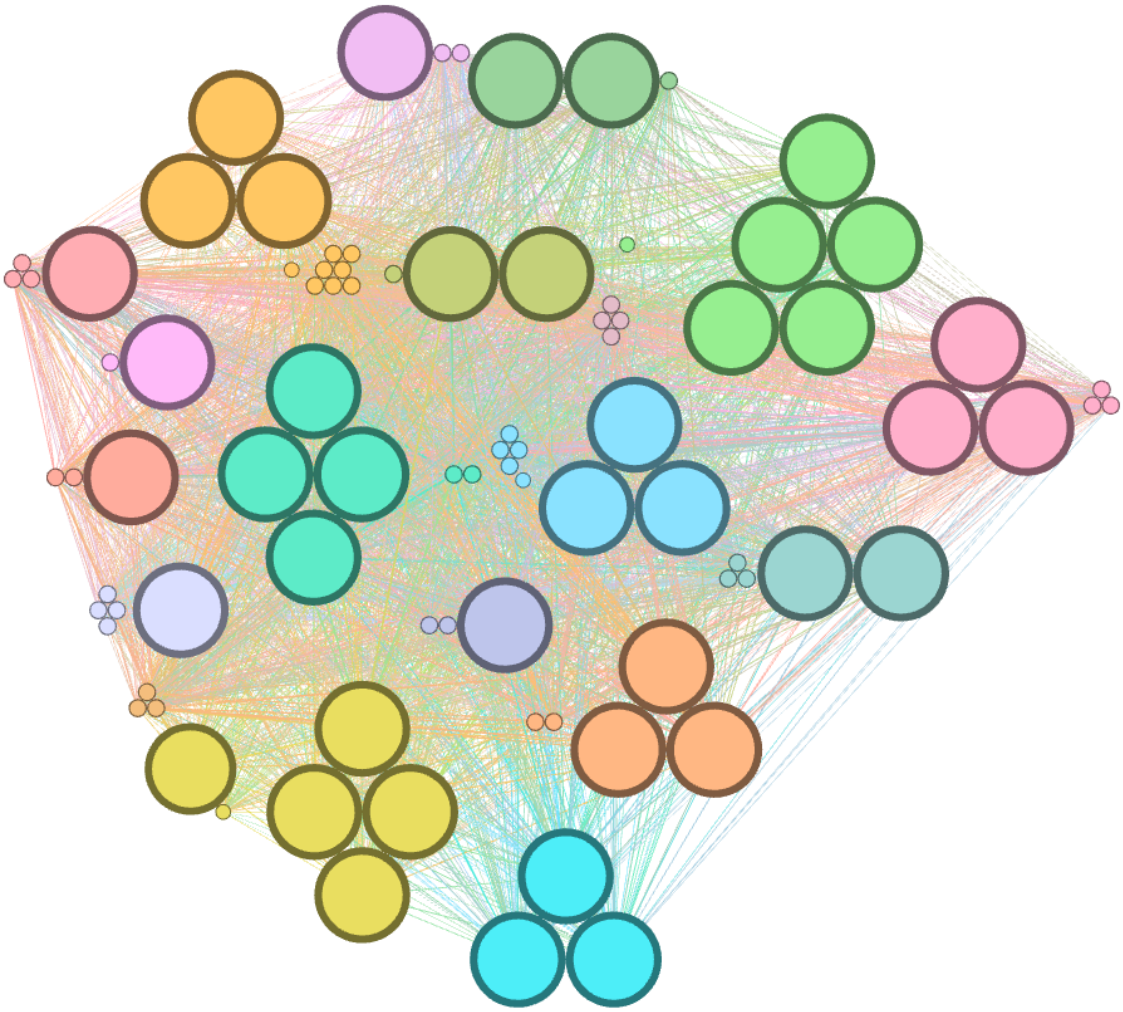

Figure B.2: Interactions graph of course C1 based on *count* aggregation function.

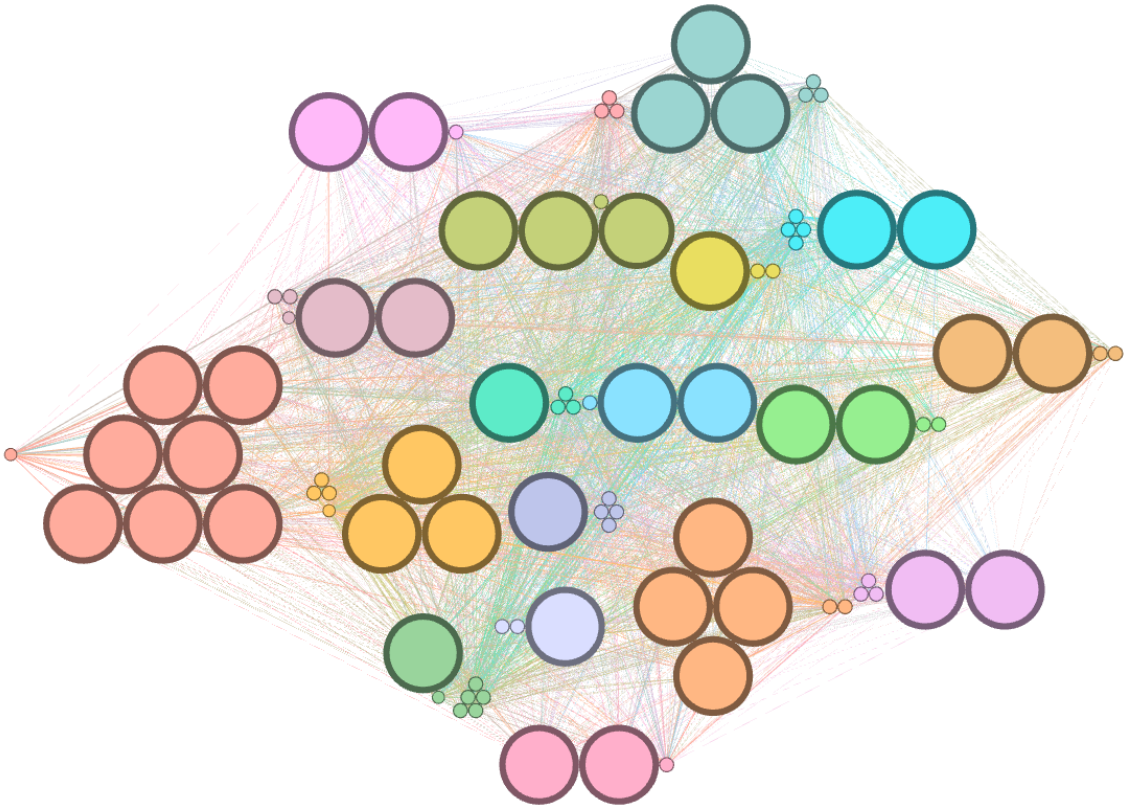

Figure B.3: Interactions graph of course C1 based on *min* aggregation function.

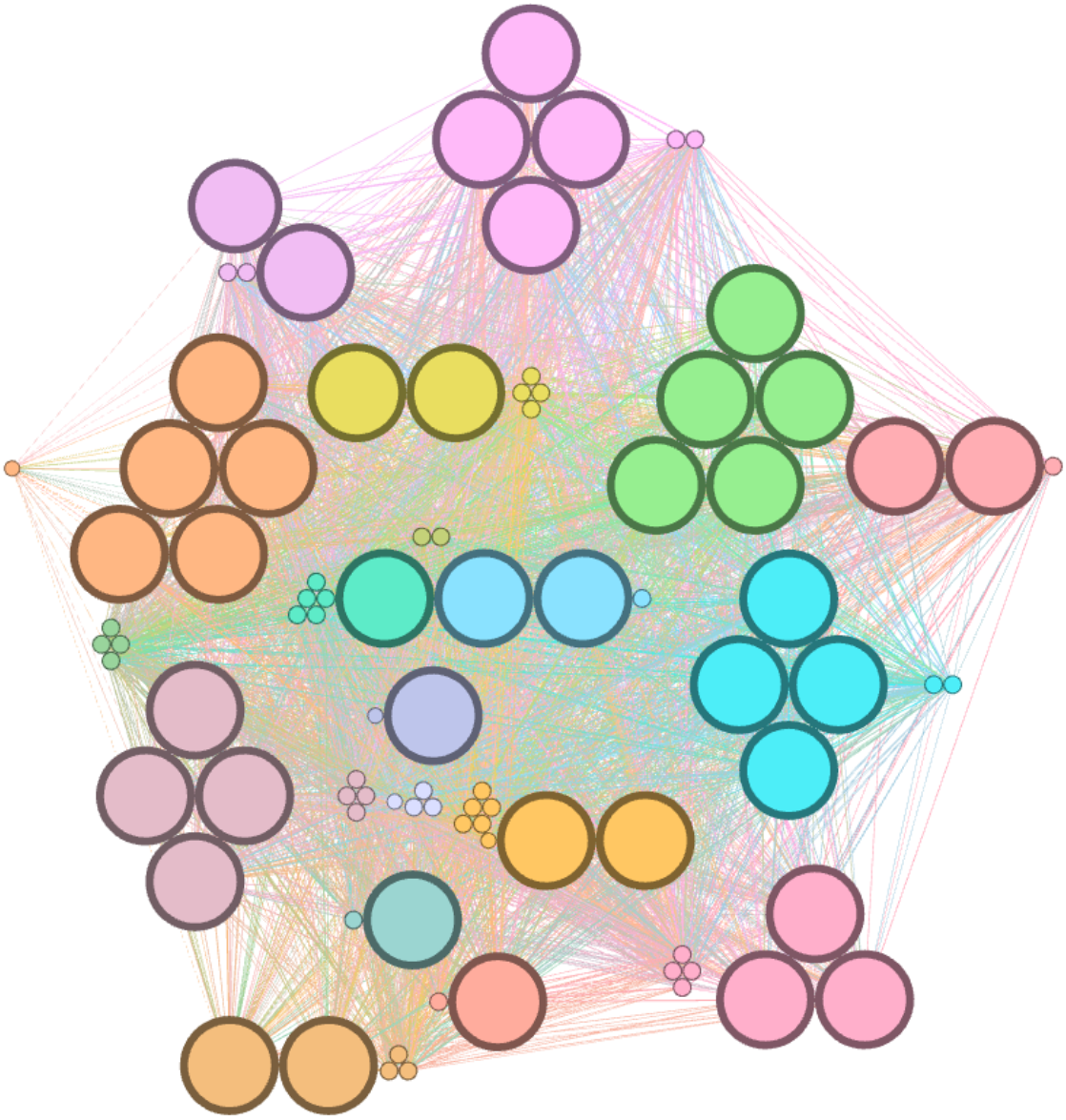

Figure B.4: Interactions graph of course C1 based on *max* aggregation function.

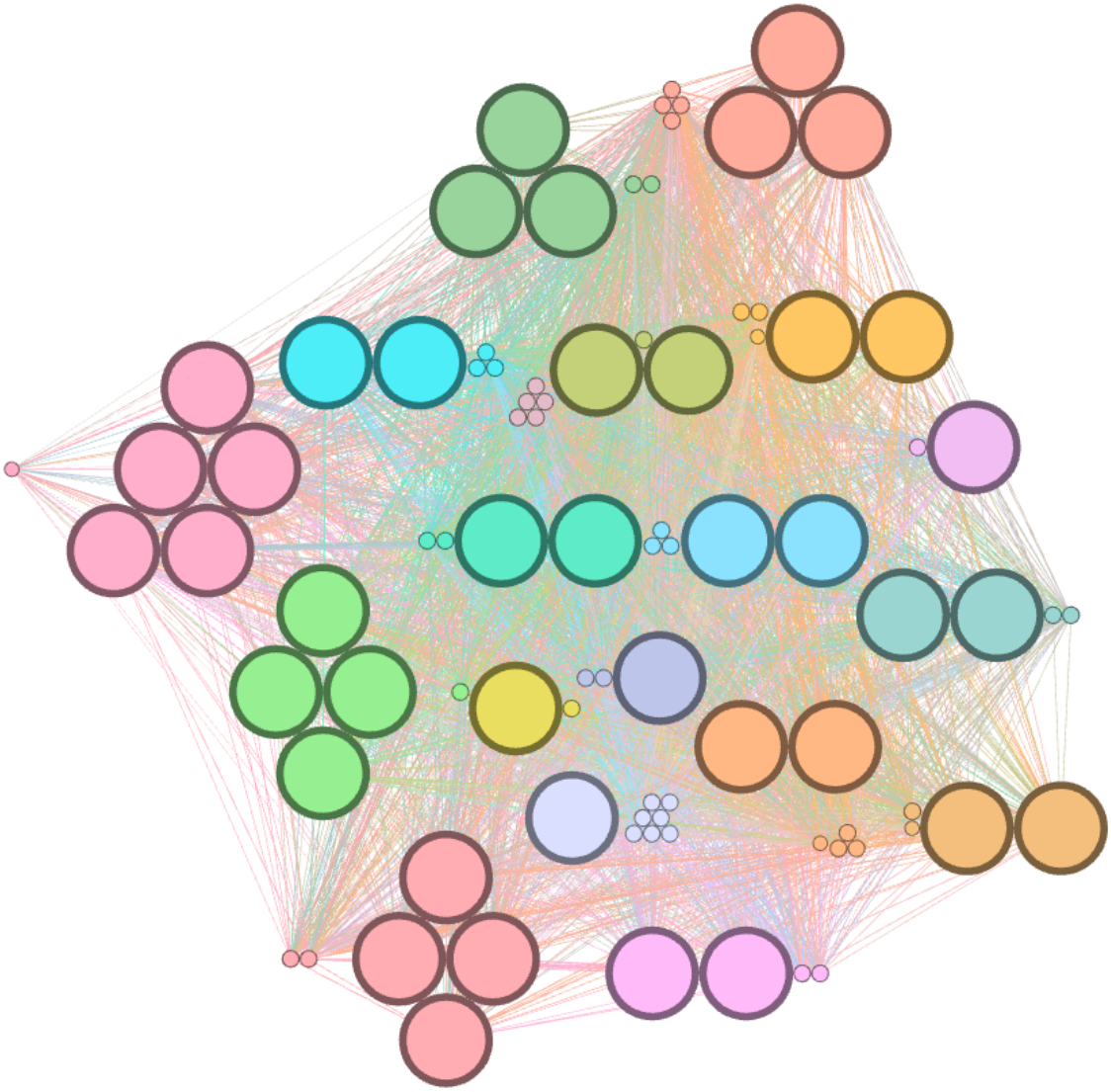

Figure B.5: Interactions graph of course C1 based on *mean* aggregation function.

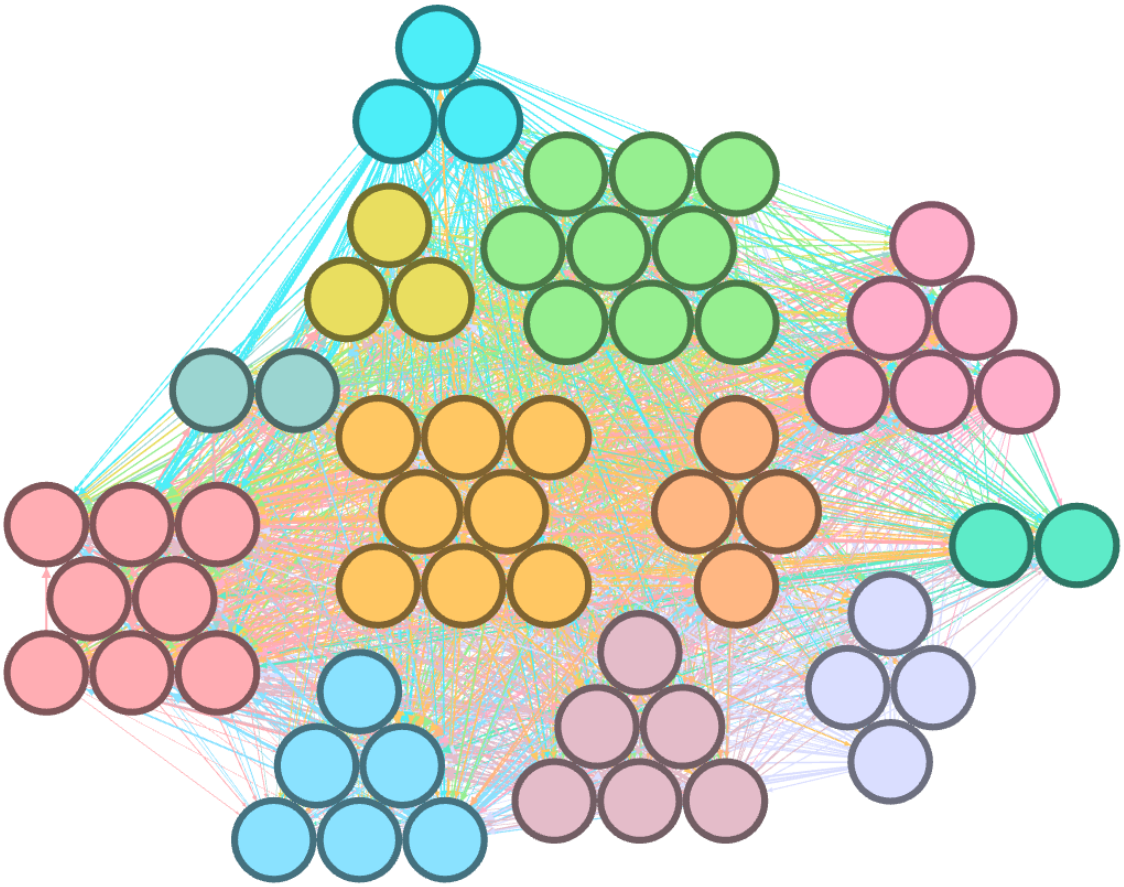

Figure B.6: Interactions graph of course C2 based on *sum* aggregation function.

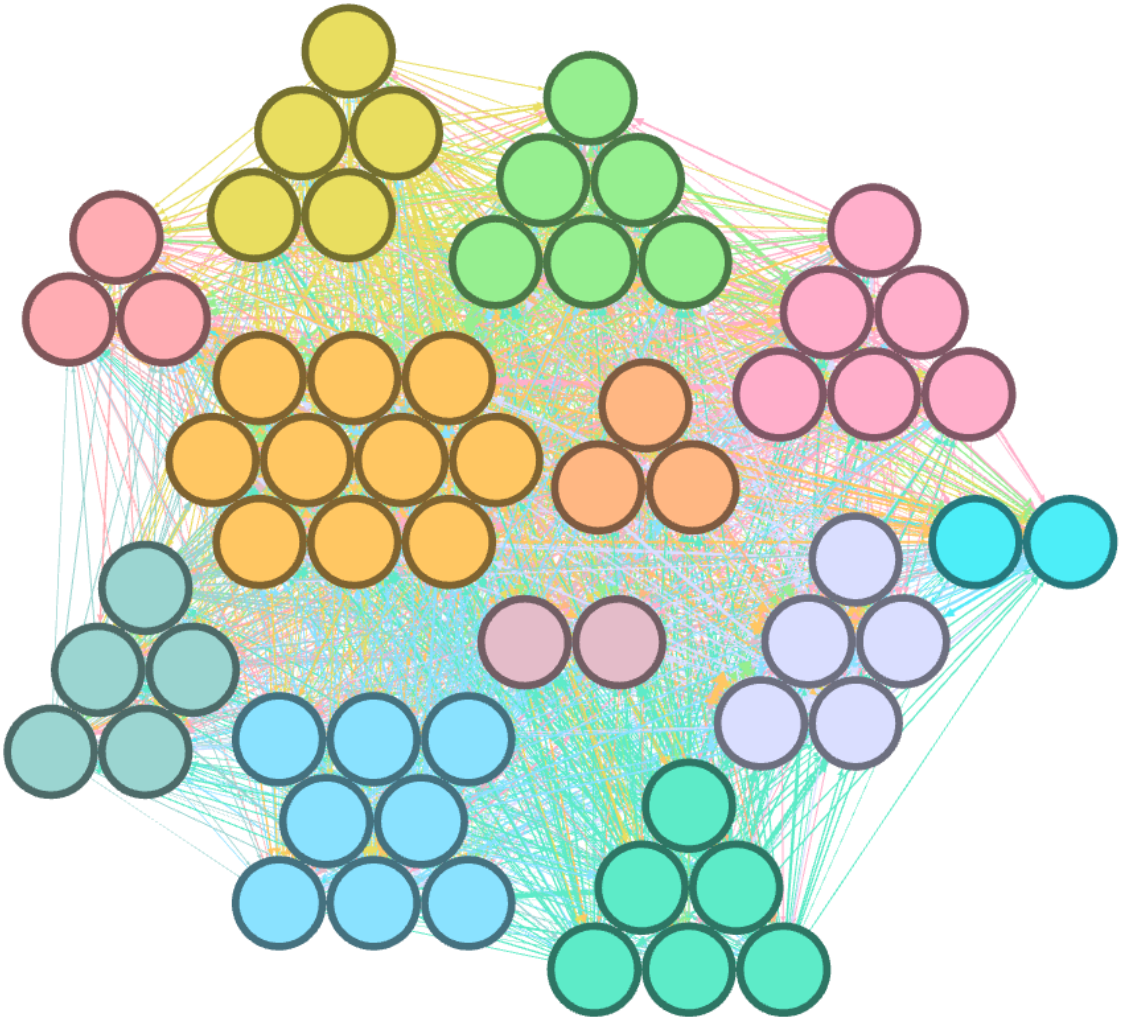

Figure B.7: Interactions graph of course C2 based on *count* aggregation function.

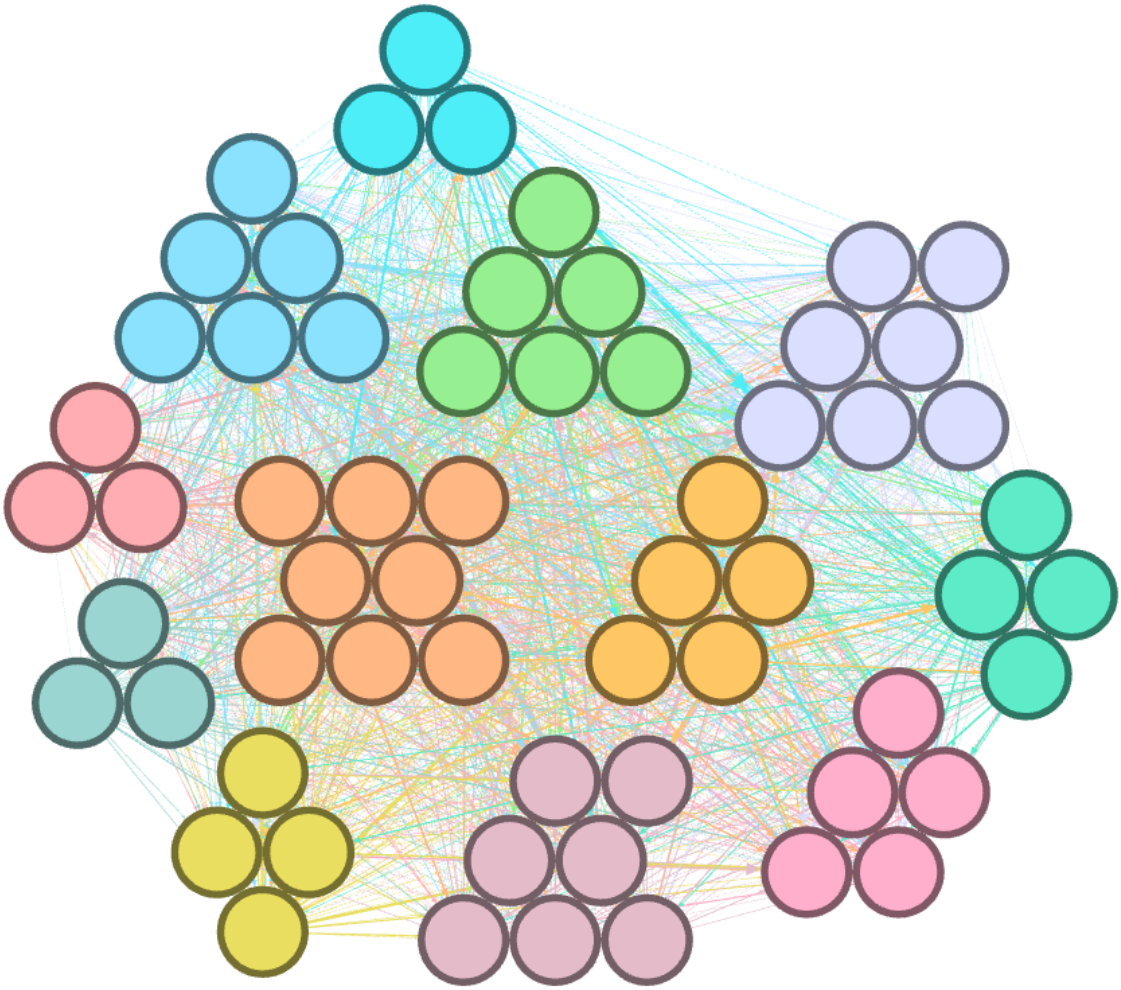

Figure B.8: Interactions graph of course C2 based on *min* aggregation function.

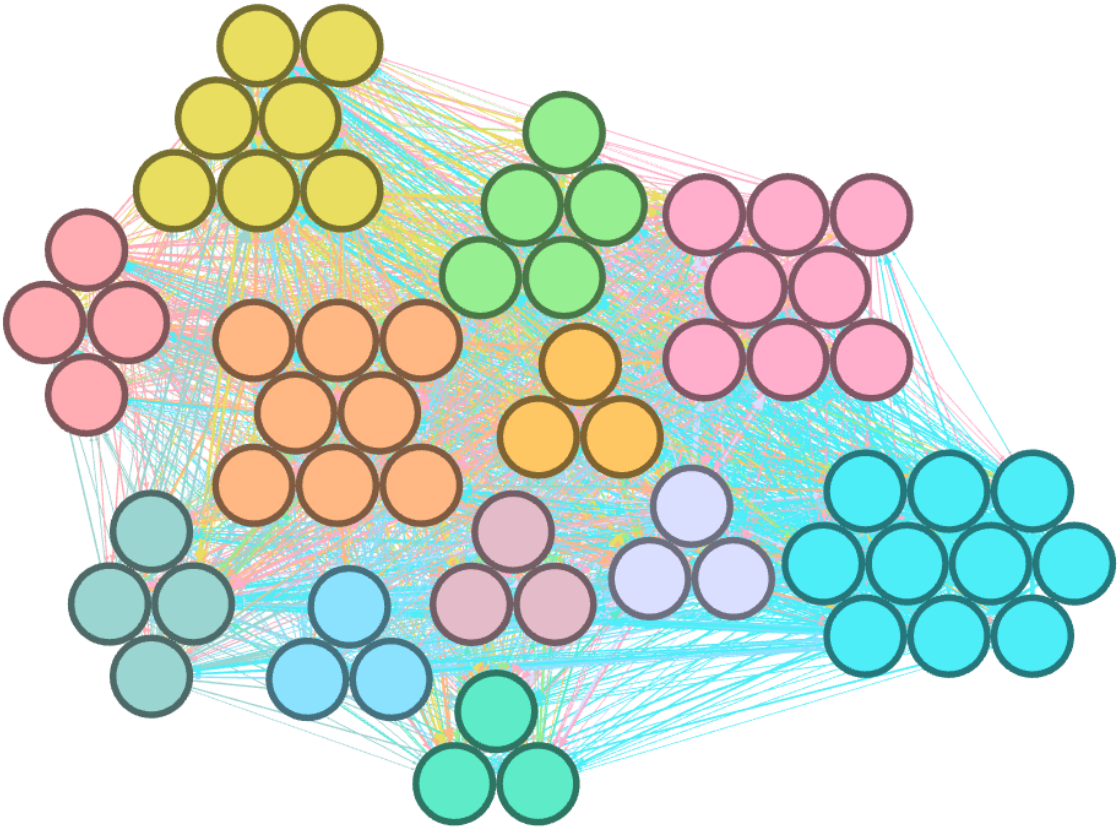

Figure B.9: Interactions graph of course C2 based on *max* aggregation function.

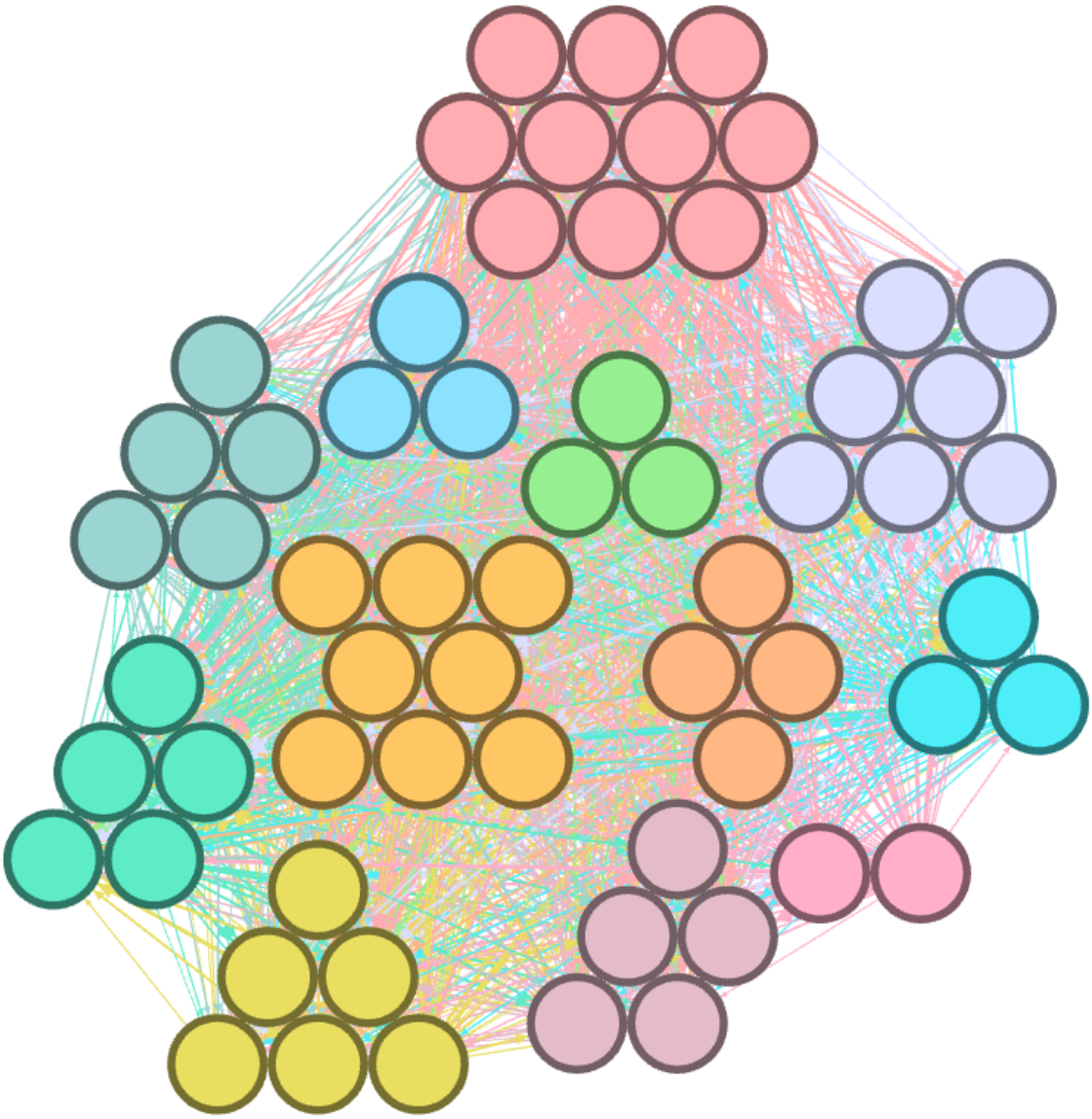

Figure B.10: Interactions graph of course C2 based on *mean* aggregation function.

Publications

1. S. Akarsh, A. Kishor, R. Niyogi, A. Milani, and Paolo Mengoni "Social cooperation in Autonomous Agents to Avoid the Tragedy of the Commons." *International Journal of Agricultural and Environmental Information Systems (IJAEIS)*, Vol. 8, no. 2, pp. 1-19, 2017

2. V. Franzoni, Y. Li, P. Mengoni, A. Milani "Clustering Facebook for Biased Context Extraction", *Computational Science and Its Applications - ICCSA 2017, Lecture Notes in Computer Science Volume 10404*, 2017, pp 717–729, Springer, Jul 2017

3. V. Franzoni, Y. Li, and P. Mengoni, "A path-based model for emotion abstraction on facebook using sentiment analysis and taxonomy knowledge", in *Proceedings of the International Conference on Web Intelligence*, WI '17, (New York, NY, USA), pp. 947—952, ACM, 2017

4. P. Mengoni, A. Milani, and Y. Li, "Community Graph Elicitation from Students' Interactions in Virtual Learning Environments", in *Computational Science and Its Applications - ICCSA 2018*, pp. 414-425, Springer, Cham, 2018.

5. P. Mengoni, A. Milani, and Y. Li, "Clustering Students Interactions in eLearning Systems for Group Elicitation", in *Computational Science and Its Applications - ICCSA 2018*, pp. 398-413, Springer, Cham, 2018.

6. V. Franzoni, P. Mengoni and A. Milani "Dimensional Morphing Interface for Dynamic Learning Evaluation", in *2018 22nd International Conference Information Visualisation (IV)*, pp. 332-337, IEEE, Fisciano, Italy, 2018.

7. P. Mengoni, A. Milani and Y. Li "Multi-term, Semantic Context Elicitation from Collaborative Networks", in *2018 IEEE First International Conference*

on Artificial Intelligence and Knowledge Engineering (AIKE), pp. 234-238, IEEE, Laguna Hills, CA, 2018.

8. J. J. Deng, C. H. C. Leung, Y. Li and P. Mengoni "Emotion Recognition from Human Behaviors using Attention Model", in *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pp. 249-253, IEEE, Laguna Hills, CA, 2018.

9. S. W. Chan, V. Franzoni, P. Mengoni and A. Milani "Context-based Image Semantic Similarity for Prosthetic Knowledge", in *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pp. 254-258, IEEE, Laguna Hills, CA, 2018.