Università degli Studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Magistrale in
Scienze Statistiche

Statistical modelling of time-stamped hypergraphs: a model-based clustering approach

Relatrice Prof.ssa Alessandra Rosalba Brazzale
Dipartimento di Scienze Statistiche, Università degli Studi di Padova

Correlatore Dott. Mirko Signorelli
Mathematical Institute, Leiden University

Correlatore Prof. Ernst-Jan Camiel Wit
Institute of Computing, Università della Svizzera Italiana

Laureanda Eugenia Driusso
Matricola 2058300

Anno Accademico 2023/2024

Contents

Chapter 1

Introduction

In the first book of *Politics*, Aristotle famously stated that "Man is by nature a social animal", emphasizing a fundamental aspect of human existence: the need for social connection and interaction, as well as the innate inclination to reflect on the laws governing relationships within society. Indeed, sociality plays a crucial role in the acquisition of knowledge. Through the exchange of opinions, dialogue, and interaction with peers, individuals enrich themselves and advance their development. Therefore, relationships play a vital role in the growth and progress of individuals and societies alike.

In mathematical terms, the connections between individuals assume the form of a network whose structure is capable of representing the intricacies of the mechanisms that underlie numerous real-world phenomena. Rather than focusing on the characteristics of entities, sometimes it's of interest to examine relationships among them, whether they are individuals or abstract objects. In network science, the objects of interest are no longer the entities that populate the network, but the connections that link them.

Network data arise in many fields, starting from studies on social relationships. One of the earliest quantitative research on this area was conducted by the psychiatrist Jacob Levy Moreno (1934), who studied friendship patterns among a group of primary school students and represented this network through what he called the *sociogram*. The fundamental element of the sociogram is that it focuses on the configuration of relationships between the actors, rather than on the distribution of the attributes possessed by the actors themselves.

This framework has then been extended to many applications in the social and behavioral sciences. Epidemiologists, for example, are interested in modeling the spread of infectious diseases, in order to understand and limit the diffusion mechanism. In biology, networks are employed to describe interactions between genes, proteins, or neurons to

unravel biological processes. Through networks, it is also possible to represent interactions among technological infrastructures, such as connections between railway stations or airports, or analyze the flow of information within websites and social networks.

Before delving into the statistical modeling of network data, it's necessary to define mathematical abstractions to represent such data.

1.1 Graphs

Network data can be seen as a set of edges that connect pairs of nodes, where the nodes represent the individuals/objects in the network, and the edges describe the relationship among them.

A graph G is defined as the ordered pair $\mathcal{G} = (V, E)$, where $V = \{v_1, ..., v_N\}$ is the set of N vertices, or nodes, that populate the network, and $E = \{e_1, ..., e_M\}$, $E \subseteq V \times V$, denotes the set of M edges linking these nodes pairwise. Consistent with the graph's definition, an edge can connect two distinct nodes, while there is at most one edge linking any two different nodes.

Alternatively, a graph can be represented through a $N \times N$ square matrix $\mathbf{X}$, called *adjacency matrix*, whose elements are defined as follows:

$$x_{ij} = \begin{cases} 1 & \text{if node } i \text{ relates to node } j \\ 0 & \text{otherwise} \end{cases} . \tag{1.1}$$

An edge that connects a node to itself is called a *self-loop*. In a *simple graph*, where interactions between the same individual are not allowed, all elements on matrix's diagonal are set to zero. In (1.1), the entries are either 1 or 0, depending on whether there exists or not a relationship between two nodes. However, edges can be assigned a numerical value, known as weights, that represents the strength of the relationship. We refer to these types of graphs as *weighted graphs*. Consider a social network where individuals are represented as nodes and friendships between individuals are depicted as edges connecting these nodes. In an unweighted graph of this social network, the edges simply signify the existence of friendships between individuals without specifying their closeness or frequency of interaction; they only indicate that a friendship exists between two individuals. In contrast, a weighted graph assigns numerical values to the edges to convey additional details about the relationships between nodes. For example, in a transportation network, nodes may represent cities, and edges may represent transportation routes between these cities. In a weighted graph of this transportation network, the edges are assigned weights corresponding to factors such as distance, travel

time, or transportation cost between cities. A direct flight between two cities may have a higher weight if it covers a longer distance or if it is more expensive.

According to the nature of the relationships among nodes, graphs can also be classified as directed or undirected. In *undirected graphs*, connections between vertices are bi-directional and symmetric, namely $x_{ij} = x_{ji}$ for all i, j, $i \neq j$. Conversely, *directed graphs*, or *digraphs*, exhibit connections with a specified directionality from one node to another. Let's take a communication network as an example, where each person is a node and the lines between them depict the emails they send to each other. This network is represented by a directed graph since the direction of the edges indicates who the sender and receiver are. Alternatively, the social network where the edges between individuals represent their friendships is an example of an undirected graph because it reflects the mutual nature of friendship between two individuals.

The development of statistical models for network data started in the Eighties, when several probabilistic models were introduced to describe and study social network data, including the p_1 model of Holland & Leinhardt (1981), stochastic blockmodels (Wang & Wong, 1987) and exponential random graph models (Frank & Strauss, 1986). Since then, many different models and approaches have been proposed to simulate and analyse graph data; a review of this topic is beyond the scope of this thesis, and we refer readers interested in the subject to the books of Holland et al. (1983) and Kolaczyk (2009).

1.2 Hypergraphs

Graphs can only represent interactions between pairs of nodes. However, in many real-world scenarios such representation can be restrictive, and it would be more appropriate to consider connections involving more than two individuals. Consider, for example, the relationship among three roommates sharing an apartment, or a team of scientists co-authoring articles together. In these situations, relationships do not occur just between pairs of subjects, but involve groups that can contain any number of people. An extension of the graph called *hypergraph* can be employed to describe such phenomena. In a hypergraph, *hyperedges* can connect any number of nodes, rather than just two.

A hypergraph is defined as $\mathcal{H} = (V, E)$, where $V = \{v_1, ..., v_N\}$ represents the set of N nodes, as in the graph discussed previously, while $E = \{e_1, ..., e_M\}$ denotes the set of M hyperedges. A hyperedge e is a subset of V, and repetitions of the same hyperedge in E are possible.

Hypergraphs can also be represented through an *incidence matrix* $\mathbf{X}$, of dimensions $N \times M$. The elements of this matrix are

$$x_{ij} = \begin{cases} 1 & \text{if node } v_i \text{ appears in hyperedge } e_j \\ 0 & \text{otherwise} \end{cases} .$$

As an example, consider a network where a group of five scientists coauthor four scientific papers. The authors can be seen as nodes v_1, v_2, v_3, v_4, v_5, and papers as the relationships that link them, namely the hyperedges e_1, e_2, e_3, e_4. We know that paper e_1 has been written by authors (v_1, v_4), paper e_2 by (v_2, v_3, v_4), paper e_3 by (v_1, v_2, v_3, v_4), and paper e_4 by (v_2, v_5). This network can be represented by

$$E = \{(v_1, v_4), (v_2, v_3, v_4), (v_1, v_2, v_3, v_4), (v_2, v_5)\}$$

as well as by the incidence matrix

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where the four columns represent the papers and the five rows represent the authors. The entries indicate whether the scientist coauthored the paper or not. Figure 1.1 shows a graphical representation of the cohautorship hypergraph.

Whereas in the probability literature hypergraphs have been studied extensively in the past decades (Karoński & Łuczak, 2002), statistical modeling of hypergraphs is more recent and less developed. Stasi et al. (2014) introduced the hypergraph beta model with three variants, which is a natural extension of the p_1 model for random graphs (Holland and Leinhardt 1981). In their model, the probability of a hyperedge appearing in the hypergraph is parameterized by a vector which represents the *attractiveness* of each vertex, which is a measure of its importance or centrality within the hypergraph. Ng & Murphy (2021) proposed an Extended Latent Class Analysis model for hypergraphs, where hyperedges are partitioned into latent classes, and the probability that a hyperedge contains a vertex depends only on its latent class assignment. Chodrow et al. (2021) and Brusa & Matias (2022) present models that extend the SBM (Stochastic Block Model) within the framework of random hypergraphs. The publication of Crane

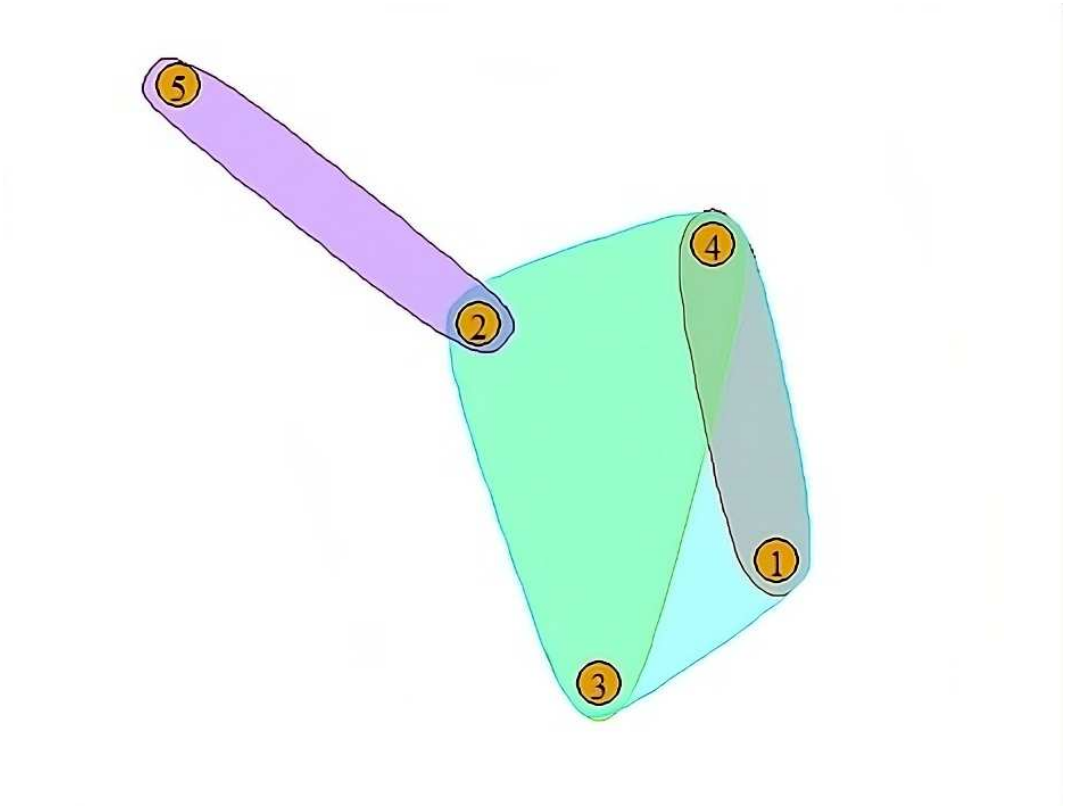

FIGURE 1.1: A hypergraph representation of the coauthorship network. The numbers correspond to the nodes, representing the authors, while the areas illustrate the hyperedges, representing the co-authored papers.

& Dempsey (2018) provide in-depth insights and further discussion on the subject matter.

1.3 Time-stamped hypergraphs

So far, we have introduced the concept of hypergraph, defined as a set of nodes and a set of hyperedges that can connect multiple nodes. We now turn our focus to *dynamic hypergraphs*, also referred to as *time-stamped hypergraphs*, an extension of hypergraphs where relationships within the network evolve over time.

Consider the example of scientific papers authored by a group of scientists. Initially, each researcher is represented as a vertex in the hypergraph, and each scientific paper they co-author forms a hyperedge connecting the contributing researchers. As time progresses, new papers are published, and researchers continue to collaborate, leading to the creation of new hyperedges. Since publications are events that happen in time, it can be interesting to study the dynamic evolution of the coauthorship hypergraph over time.

In a dynamic hypergraph, each hyperedge is associated with a specific time at which the event occurs. Let N be the number of nodes in the hypergraph, and M be the number of hyperedges observed over time, namely, the number of events that occur within the timeframe in which we observe the hypergraph. The dynamic hypergraph is then defined as $\mathcal{H} = (V, E, T)$, where $V = \{v_1, ..., v_N\}$ represents the set of N nodes, $E = \{e_1, ..., e_M\}$ represents the set of the observed hyperedges, and $T = \{t_1, ..., t_M\}$ represents the times $t_j = (t_{j1}, ..., t_{jK_j})$, $t_{jk} \in \mathbb{R}^+$, $j = 1, ..., M$, $k = 1, .., K_j$ at which each hyperedge occurs. This formulation enables the same hyperedge to be observed multiple times.

Lately, time-stamped hypergraph modeling has emerged as a new area of research. Recent work by Lerner & Lomi (2023) has proposed the relational hyper-event models (RHEM) for the analysis of polyadic interaction networks. RHEM is specifically designed for dynamic hypergraphs, where a sender communicates with multiple receivers simultaneously. The model considers event rates based on hyperedge covariates associated with the sender and all receivers collectively.

1.4 Goals

In this thesis, we will propose a new model-based clustering approach for dynamic hypergraphs, namely a survival generalization of the Latent Class Analysis (LCA) model for time-dependent hypergraphs, where the hazard rate of the hyperedges depends on their latent class membership and on some node-specific parameters.

The thesis is structured as follows.

Chapter 2 will introduce two probabilistic models for hypergraphs that capture the clustering structure of the hyperedges: the Latent Class Analysis model for hypergraphs and the Extended Latent Class Analysis model proposed by Ng & Murphy (2021). Procedures for estimating and selecting models for each approach will be outlined. Additionally, an algorithm for generating the hypergraph matrix in the Extended Latent Class Analysis model will be presented, along with a simulation study to evaluate the performance of ELCA's estimation algorithm.

In Chapter 3, the proposed model for time-stamped hypergraphs, named Temporal LCA (TLCA) model, will be developed. The parameter estimation procedure will be described in detail, focusing particularly on the Expectation-Maximization algorithm employed for the estimation. Furthermore, we will discuss a sequential algorithm developed for simulating the hypergraph data, allowing for hyperedge recurrence within the survival analysis framework.

Chapter 4 will conduct a simulation study to assess the performance of the proposed model, analyzing its strengths and weaknesses.

Finally, in Chapter 5, potential future developments will be discussed.

Chapter 2

Latent class analysis models for hypergraphs

In Chapter 1, we have provided an introduction to the concepts of graphs and hypergraphs, elucidating their fundamental properties and differences. We will now present two models for the analysis of hypergraph data, which inspired the development of the TLCA hypergraph model that we propose in Chapter 3.

2.1 A latent class analysis model for hypergraph data

2.1.1 Model specification

The Latent Class Analysis (LCA) model was introduced by Lazarsfeld & Henry (1968) and then developed by Goodman (1974). It is a mixture model for binary data in which each observation is assigned to one and only one latent class that assumes that the observed variables are mutually independent of each other conditionally on their latent class membership.

Ng & Murphy (2021) proposed to use LCA to model random hypergraphs by categorizing the hyperedges into G latent classes. In the LCA hypergraph model, the probability of a hyperedge e_j containing a node v_i depends only on its latent class assignment.

Suppose that there exist G clusters of hyperedges, and each node v_i has probabilities $p_{i1}, ..., p_{iG}$ of taking part in hyperedges belonging to a cluster g. Denote by z_j the latent group membership of the hyperedge e_j (so that $z_j = g$ if e_j belongs to cluster g),

and by $\pi_g = P(Z_j = g)$ the *a priori* latent class membership probability, which is the probability of a hyperedge e_j being assigned to cluster g. Under these assumptions, the probability of observing a hyperedge $e_j = (x_{1j}, ..., x_{Nj})$ is

$$P(e_j) = \sum_{g=1}^{G} P(e_j|z_j = g)P(z_j = g) = \sum_{g=1}^{G} \pi_g \prod_{i=1}^{N} p_{ig}^{x_{ij}}(1 - p_{ig})^{1-x_{ij}},$$

and the likelihood function of $\mathbf{P} = (p_{ig})$ and $\pi = (\pi_1, ..., \pi_g)$ takes the form of

$$L(\mathbf{P}, \pi|\mathbf{X}) = \prod_{j=1}^{M} \left[\sum_{g=1}^{G} \pi_g \prod_{i=1}^{N} p_{ig}^{x_{ij}}(1 - p_{ig})^{1-x_{ij}} \right]. \tag{2.1}$$

The LCA hypergraph model has $(G - 1) + NG$ parameters, that can be estimated via the Expectation-Maximization (EM) algorithm.

2.1.2 Estimation and model selection

The EM algorithm was formalized by Dempster et al. (1977) to compute maximum likelihood estimates from incomplete data. The EM algorithm can be employed to estimate finite mixture models, as the LCA hypergraph model is. Each iteration of the algorithm consists of an expectation step (*E-step*) followed by a maximization step (*M-step*). Let $l(\theta|x, z) = \log L(X, Z|\theta)$ be the log-likelihood function of the model parameter θ that relies on both the observed data x and unobserved data z. The expectation step entails computing the expected value of the objective function (here the log-likelihood function l) conditionally on the estimated parameters from the last M-step, $\theta^{(t-1)}$:

$$E_Z[l(Z|X, \theta^{(t-1)}].$$

In the LCA model, the likelihood is the one shown in (2.1). In the maximization step, the parameters are updated by maximizing the conditional mean derived in the *E-step*:

$$\theta^{(t)} = \text{argmax}_\theta E_Z[l(Z|X, \theta^{(t-1)}].$$

The EM algorithm estimates the model parameters for a given value of the number of clusters, but it doesn't give any information about the optimal number of latent classes. In order to determine it, we can run the EM algorithm with different number of clusters and then use a criterion to select the best model. For the LCA hypergraph model, Ng & Murphy suggested using the Bayesian Information Criterion (Schwarz,

1978), $BIC = -2\log l + p\log M$, where l is the log-likelihood evaluated at the maximum likelihood estimate (MLE) and p is the number of parameters of the model, to select the optimal number of latent classes. Other selection criteria can also be used, e.g., the Akaike Information Criterion (Akaike, 1974). However, existing literature suggests that the BIC is a reliable indicator of the real number of classes in the context of standard latent class models (Collins et al. 1993, Nylund et al. 2007).

2.2 The extended latent class analysis model

The LCA hypergraph model captures the clustering tendencies and the size fluctuations of the hyperedges, while parameter estimation remains relatively straightforward. However, as the number of nodes in the hypergraph increases, the number of parameters escalates rapidly. For this reason, Ng & Murphy introduced a more parsimonious version of the LCA model, the Extended Latent Class Analysis (ELCA) model, which will be elaborated upon in the forthcoming section.

2.2.1 Model specification

The ELCA model (Ng & Murphy, 2021) reduces the complexity of the LCA hypergraph model by making a proportionality assumption on the latent conditional probabilities $(p_{ig})_{i=1}^{N}$, meaning that some of the conditional probabilities tend to be proportional to each other for different values of g.

Specifically, the ELCA model assumes that the latent conditional probabilities are of the form

$$p_{ig} = a_k \phi_{ig}, \tag{2.2}$$

where the ϕ_{ig} parameters are proportional to the probability of a vertex being included within a hyperedge for $i = 1, ..., N$ and $g = 1, ..., G$, while the vector $a = (a_1, ..., a_K)$ with $0 \leq a_k \leq 1$, $k = 1, ..., K$ captures the fluctuations in the size of the hyperedges, measured by the number of vertices that it links. Therefore, the ELCA model is a latent class model with two clustering structures, a primary clustering structure specified by the parameter ϕ_{ig}, with the corresponding cluster assignment probability π_g, and a secondary clustering structure defined by a_k, with probability τ_k of a hyperedge being assigned to additional cluster k. Note that if $K = 1$, the ELCA model reduces to the LCA hypergraph model.

Let z_{1j} and z_{2j} be respectively the primary and secondary latent class membership of the hyperedge e_j. The ELCA model assumes that the two clustering structures

are *a priori* independent, therefore the probability that a hyperedge is assigned to primary cluster g and secondary cluster k is $P(e_j|z_{1j} = g, z_{2j} = k) = \pi_g\tau_k$. Under these assumptions, the probability of observing a hyperedge $e_j = (x_{1j}, ..., x_{Nj})$ becomes

$$\begin{aligned} P(e_j) &= \sum_{g=1}^{G} P(e_j|z_{1j} = g, z_{2j} = k)P(z_{1j} = g)P(z_{2j} = k) \\ &= \sum_{g=1}^{G}\sum_{k=1}^{K} \pi_g\tau_k \prod_{i=1}^{N} (a_k\phi_{ig})^{x_{ij}}(1 - a_k\phi_{ig})^{1-x_{ij}}. \end{aligned}$$

The likelihood function of the ELCA model can be expressed as

$$L(\pi, \tau, \boldsymbol{\phi}, \alpha|\mathbf{X};) = \prod_{j=1}^{M} \left[\sum_{g=1}^{G}\sum_{k=1}^{K} \pi_g\tau_k \prod_{i=1}^{N} (a_k\phi_{ig})^{x_{ij}}(1 - a_k\phi_{ig})^{1-x_{ij}} \right]. \tag{2.3}$$

To ensure the identifiability of the model, some constraints have to be imposed on the parameters $\theta = (\pi, \tau, \boldsymbol{\phi}, a)$, in particular:

- $\pi = (\pi_1, ..., \pi_G) \in [0, 1]^G$, $\sum_{g=1}^{G} \pi_g = 1$;
- $\tau = (\tau_1, ..., \tau_K) \in [0, 1]^K$, $\sum_{k=1}^{K} \tau_k = 1$;
- $\boldsymbol{\phi} = (\phi_{ig})$, $\phi_{ig} \in [0, 1]$. This follows from $p_{ig} = a_k\phi_{ig} \in [0, 1]$, because if $K = 1$ we have $\phi_{ig} = p_{ig} \in [0, 1]$;
- $a = (a_1, ..., a_K)$, ranked by increasing order $0 < a_1 < a_2 < ... < a_K = 1$.

The number of parameters in the ELCA model is $GN+2(K-1)+(G-1)$. Note that the ELCA model with G primary clusters and K secondary clusters is a restriction of to the standard LCA hypergraph model with KG clusters. The number of parameters in the correspondent LCA model is then $GKN + (GK - 1)$. Therefore, the ELCA model achieves a substantial reduction in the number of parameters.

Tables 2.1 and 2.2 provide numerical insights into the extent of complexity reduction achieved by the ELCA model in comparison to the LCA model when $n = 10$ and $n = 50$. These tables include the number of primary and secondary clusters in the ELCA model, the corresponding number of clusters in the LCA model, and the number of parameters to be estimated in both models.

We observe that the ELCA model substantially reduces the number of parameters to be estimated compared to the LCA model, even with a small number of primary and secondary clusters: a reduction of 42% is observed with $G = 2$ and $K = 2$ with $n = 10$,

and of 48% with $n = 50$. The gain in model parsimony increases with both the number of primary and secondary clusters, and with the number of nodes.

N	G_{ELCA}	K_{ELCA}	G_{LCA}	$\lvert\theta_{LCA}\rvert$	$\lvert\theta_{ELCA}\rvert$	difference	reduction
10	2	2	4	43	25	18	42%
10	3	2	6	65	37	28	43%
10	4	2	8	87	49	38	44%
10	2	3	6	65	29	36	55%
10	3	3	9	98	42	56	57%
10	4	3	12	131	55	76	58%

TABLE 2.1: Model complexity for the ELCA model versus the LCA model with 10 nodes. Here, G_{ELCA} and K_{ELCA} represents the number of primary and secondary clusters in the ELCA model, G_{LCA} represents the number of clusters in the LCA model, $|\theta_{LCA}|$ and $|\theta_{ELCA}|$ represent the total number of parameters in the LCA and TLCA model, respectively.

N	G_{ELCA}	K_{ELCA}	G_{LCA}	$\lvert\theta_{LCA}\rvert$	$\lvert\theta_{ELCA}\rvert$	difference	reduction
50	2	2	4	203	105	98	48%
50	3	2	6	305	157	148	49%
50	4	2	8	407	209	198	49%
50	2	3	6	305	109	196	64%
50	3	3	9	458	162	296	65%
50	4	3	12	611	215	396	65%

TABLE 2.2: Model complexity for the ELCA model versus the LCA model with 50 nodes. Here, G_{ELCA} and K_{ELCA} represents the number of primary and secondary clusters in the ELCA model, G_{LCA} represents the number of clusters in the LCA model, $|\theta_{LCA}|$ and $|\theta_{ELCA}|$ represent the total number of parameters in the LCA and TLCA model, respectively.

2.2.2 Estimation and model selection

Although the parameters $\theta = (\pi, \tau, \boldsymbol{\phi}, a)$ of the ELCA model can be estimated via the EM algorithm, the M-step is quite complex to compute. Thus, Ng & Murphy (2021) replaced the Expectation-Conditional Maximization (ECM) algorithm (Meng & Rubin, 1993), which involves a series of conditional maximizations with respect to the model parameters. Since the maximizations concerning $\boldsymbol{\phi}$ and a do not have a closed-form solution, Ng & Murphy (2021) proposed to use the Minorization Maximization (MM) algorithm (Hunter & Lange, 2004), which consists of lower bounding the objective function to obtain a minorizing function that is subsequently maximized to estimate the model parameters.

Similarly to the LCA hypergraph model, also for the ELCA model the number of primary and secondary clusters can be chosen minimizing the Bayesian Information Criterion. The BIC tends to be more accurate when the number of hyperedges is much larger than the number of nodes in the hypergraph.

2.2.3 Simulating hypergraph data

In order to gain a deeper understanding of the data-generating mechanism, which is crucial for extending these models within the context of dynamic hypergraphs, we describe the procedure that we developed to simulate the hypergraph data matrix from the ELCA model. It is worth noting that in the case of a single secondary cluster, so for $K = 1$, the ELCA model is equivalent to the LCA model. This data-generating mechanism will be employed in the simulation studies discussed in Section 2.2.4.

We propose the following procedure to generate the incidence matrix $\mathbf{X} = (x_{ij})$, of dimensions $N \times M$:

1. Set N, M, G, K;
2. Given N, M, G, K, fix the model parameters $\pi, \tau, \boldsymbol{\phi}, a$;
3. For $j = 1, ..., M$:
   (a) use a pseudo-random number generator from the multinomial distribution to simulate the primary latent class membership,
   $$z_{1j} \sim Multinom_G(\pi);$$
   (b) use a pseudo-random number generator from the multinomial distribution to simulate the secondary latent class membership,
   $$z_{2j} \sim Multinom_K(\tau).$$
   Note that the two clustering levels are simulated independently because ELCA assumes independence between the primary and secondary cluster.
4. Based on z_{1j}, z_{2j}, $a_{iz_{2j}}$ and $\phi_{iz_{1j}}$, apply the Acceptance-Rejection sampling method (Rizzo, 2019) to determine $\mathbf{X}$:
   (a) Draw NM pseudo-random numbers $u_1, ..., u_{NM}$ from a uniform distribution;

(b) For $j = 1, .., M$ and $i = 1, ..., N$: set

$$x_{ij} = \begin{cases} 1 & \text{if } u_{ij} < \phi_{iz_{1j}} a_{iz_{2j}} \\ 0 & \text{otherwise} \end{cases} .$$

In fact, according to the ELCA proportionality assumption (2.2), the latent conditional probabilities are of the form

$$P(x_{ij} = 1 | z_{1j}, z_{2j}, \phi_{iz_{1j}}, a_{iz_{2j}}) = a_{iz_{2j}} \phi_{iz_{1j}}.$$

2.2.4 Simulation study

In this section, we show the results of two Monte Carlo simulation studies based on 1000 replicates that we designed to assess the performance of the ELCA model. In particular, we want to assess the convergence behavior of the EM algorithm with various values of the model parameters $\theta = (\pi, \tau, a, \boldsymbol{\phi})$ and of N, G and M.

In Table 2.3, the hyperedges are simulated from the ELCA model with $G = 2$ primary clusters and $K = 2$ secondary clusters, in Table 2.4, hyperedges are simulated from ELCA model with $G = 3$ and $K = 2$.

The specific model parameters for Table 2.3 are:

$$\begin{aligned} \pi &= (\frac{1}{2}, \frac{1}{2}) \\ \tau &= (\frac{1}{2}, \frac{1}{2}) \\ a &= (\frac{1}{2}, 1) \\ \boldsymbol{\phi} &= \begin{pmatrix} 0.8 & ... & 0.8 & 0.1 & ... & 0.1 \\ 0.4 & ... & 0.4 & 0.4 & ... & 0.4 \end{pmatrix}. \end{aligned}$$

The model parameters for Table 2.4 are:

$$\begin{aligned} \pi &= (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}) \\ \tau &= (\frac{1}{2}, \frac{1}{2}) \\ a &= (\frac{1}{2}, 1) \\ \boldsymbol{\phi} &= \begin{pmatrix} 0.8 & ... & 0.8 & 0.1 & ... & 0.1 \\ 0.1 & ... & 0.1 & 0.8 & ... & 0.8 \\ 0.4 & ... & 0.4 & 0.4 & ... & 0.4 \end{pmatrix}. \end{aligned}$$

The l_1 distances between the true parameters and the estimated ones are presented in Table 2.3 and Table 2.4. They are calculated as the sum of the absolute differences between each element in the parameter divided by its cardinality. The *misclassification rates*, namely the number of incorrect predictions divided by the total number of predictions, are also presented for both the primary (mis_1) and the secondary (mis_2) clusters.

We observe that, for a fixed value of the number of nodes N, the estimated parameters converge to the true values as the number of hyperedges M increases. Notably, this convergence tends to occur more rapidly in scenarios characterized by two primary clusters, compared to those with three primary clusters. Moreover, we observe that the misclassification rates for both primary and secondary clusters are more significantly influenced by the number of nodes N rather than the number of hyperedges M in the hypergraph. Specifically, as the number of nodes grows, we observe a corresponding increase in the proportion of accurate predictions. The performance of the ELCA model across varying values of M and N in this simulation study is consistent with the findings outlined by Ng & Murphy (2021).

N	M	$l_1(\pi)$	$l_1(\tau)$	$l_1(a)$	$l_1(\boldsymbol{\phi})$	mis_1	mis_2
10	100	0.115	0.158	0.086	0.099	0.232	0.252
	500	0.046	0.071	0.029	0.039	0.186	0.224
	1000	0.031	0.050	0.020	0.027	0.183	0.218
20	100	0.067	0.088	0.047	0.075	0.128	0.154
	500	0.026	0.033	0.015	0.030	0.106	0.131
	1000	0.017	0.023	0.010	0.021	0.103	0.129
40	100	0.054	0.066	0.043	0.075	0.062	0.081
	500	0.022	0.027	0.014	0.034	0.046	0.061
	1000	0.014	0.018	0.010	0.025	0.044	0.060

TABLE 2.3: Convergence analysis of the EM algorithm for the ELCA model with 2 primary clusters and 2 additional clusters. $l_1(\pi)$, $l_1(\tau)$, $l_1(a)$ and $l_1(\boldsymbol{\phi})$ refers to the l_1 distance between the true parameters and the estimated ones. mis_1 and mis_2 represent the misclassification rates for the primary and the secondary clusters.

N	M	$l_1(\pi)$	$l_1(\tau)$	$l_1(a)$	$l_1(\boldsymbol{\phi})$	mis_1	mis_2
10	100	0.102	0.176	0.108	0.139	0.334	0.248
	500	0.057	0.071	0.032	0.063	0.275	0.217
	1000	0.034	0.048	0.020	0.039	0.253	0.208
20	100	0.086	0.114	0.090	0.110	0.234	0.171
	500	0.024	0.031	0.015	0.037	0.146	0.121
	1000	0.017	0.022	0.010	0.026	0.141	0.120
40	100	0.066	0.094	0.103	0.096	0.132	0.104
	500	0.019	0.023	0.012	0.035	0.060	0.051
	1000	0.014	0.016	0.009	0.026	0.059	0.050

TABLE 2.4: Convergence analysis of the EM algorithm for the ELCA model with 3 primary clusters and 2 additional clusters. $l_1(\pi)$, $l_1(\tau)$, $l_1(a)$ and $l_1(\boldsymbol{\phi})$ refers to the l_1 distance between the true parameters and the estimated ones. mis_1 and mis_2 represent the misclassification rates for the primary and the secondary clusters.

Chapter 3

Model-based clustering for time-stamped hypergraphs

In this chapter, we propose a model for the analysis of dynamic hypergraphs that extends the LCA model by including the temporal component. We call this new model we propose Temporal LCA (TLCA) model.

First and foremost, it's essential to underline that the LCA hypergraph model and TLCA model have significantly different goals. While the LCA model takes the set of M observed hyperedges as given, and models solely the group membership of such hyperedges, our model considers every potential hyperedge at risk, and studies which factors affect the occurrence and timing of hyperedges.

3.1 Model specification

The concept of dynamic hypergraphs involves the association of each hyperedge with a specific time, indicating the occurrence of an event. We recall that dynamic hypergraphs can be represented as $\mathcal{H} = (V, E, T)$, where $V = \{v_1, ..., v_N\}$ represents the set of N nodes, $E = \{e_1, ..., e_M\}$ represents the set of the M observed hyperedges, and $T = \{t_1, ..., t_M\}$ represents the times $t_j = (t_{j1}, ..., t_{jK_j})$, $t_{kj} \in \mathbb{R}^+$, $j = 1, ..., M$, $k = 1, .., K_j$ at which each hyperedge occurs.

In a dynamic hypergraph, hyperedges are therefore considered as events that happen sequentially in time. To model the time process that leads to the formation of hyperedges, we approach dynamic hypergraph modeling from the perspective of survival analysis, and extend the LCA hypergraph model of Ng & Murphy (2021) to a survival setting.

Survival data are generally described and modeled by means of two related functions, the *survival function* and the *hazard function*. Let $T \in [0, \infty)$ be a continuous random variable representing the time until an event occurs. Its probability density function is $f(t)$, and the cumulative distribution function is given by $F(t) = P(T \leq t) = \int_{-\infty}^{t} f(u)du$. In the survival analysis framework, the survival function $S(t)$ represents the probability that the event of interest has not occurred between the time of origin and time t, and can be expressed as

$$S(t) = P(T > t) = \int_{t}^{\infty} f(u)du = 1 - F(t).$$

The hazard function $\lambda(t)$ expresses the conditional probability that the event will occur within $[t, t + \delta t)$, given that it has not occurred before. Its expression is given by

$$\lambda(t) = \lim_{\delta t \to 0} \frac{P(t \leq T < t + \delta t \,|\, T \geq t)}{\delta t} = \lim_{\delta t \to 0} \frac{P(t \leq T < t + \delta t)}{\delta t \cdot S(t)} = \frac{f(t)}{S(t)}.$$

Finally, the *cumulative hazard function*

$$\Lambda(t) = \int_{-\infty}^{t} \lambda(u)du = -\log(S(t))$$

measures the "total amount of risk" that has been accumulated up to time t.

As custumary in survival analysis, in the TLCA model it's necessary to consider not only the events that occur over the study period (the M observed hyperedges), but also all the ones that could have potentially happened and are therefore considered *at risk* to occur. For this reason, in our model we consider the set $H = \{h_1, ..., h_{2^N}\}$ of all the possible hyperedges that can be obtained as combinations of the N binary entries $x_{ij} \in \{0, 1\}$ of the hyperedge vector.

In the TLCA model, the same hyperedge can occur multiple times. If a hyperedge doesn't occur within the end of the study, it will be classified as *right-censored*. We indicate right censoring through the binary variable δ. If $\delta_j = 1$, the hyperedge h_j is observed, otherwise, if $\delta_j = 0$ the hyperedge h_j does not occur before the end of the study period.

Since our model is a survival extension of the LCA hypergraph model for time-stamped hypergraphs, we assume that the 2^N possible hyperedges are distributed across G latent classes. Each hyperedge $(h_j)_{j=1}^{2^N}$ is characterized by a timestamp vector $t_j = (t_{j1}, ..., t_{jK_j})$ and an event indicator vector $\delta_j = (\delta_{j1}, ..., \delta_{jK_j})$. It is worth noting that if $k < K_j$, then $\delta_{jk} = 1$, while $\delta_{jk} = 0$ for $k = K_j$. This means that the hyperedge h_j is

observed $K_j - 1$ times, and the last time t_{Kj} corresponds to the end of the study, when the hyperedge is censored.

In order to model the hazard function, we turn to the Cox proportional hazards model (Cox, 1972). This model enables us to examine how the covariates influence the hazard rate over time. Specifically, for a hyperedge h_j belonging to latent class g, the hazard function is expressed as:

$$\lambda_{jg}(t) = \lambda_0(t)e^{\eta_{jg}} = \lambda_0(t)e^{\gamma_g + \sum_{i=1}^{N} \theta_{ig} x_{ij}},$$

where η_{jg} is the linear predictor, $\lambda_0(t)$ represents the baseline hazard, namely the hazard calculated in the reference group (where all covariates are equal to zero), and $\gamma = (\gamma_1, \ldots, \gamma_G)$ and $\boldsymbol{\theta} = (\theta_{ig})$ denote the model parameters.

In particular, γ_g allows for a distinct baseline hazard for each latent group or cluster g, represented as $\lambda_g(t) = \lambda_0(t)e^{\gamma_g}$. From an interpretive perspective, $\gamma_g > 0$ suggests that the g-th cluster exhibits a baseline hazard higher than that of the reference group. In practical terms, this means that within the hypergraph, there is a greater likelihood of observing a hyperedge associated with group g over time. Consequently, the survival, defined as the probability of the event not occurring, diminishes within group g, reflecting a higher risk of event occurrence. Conversely, $\gamma_g < 0$ suggests a lower baseline hazard, signifying a reduced inherent risk within that specific group. Similarly, the effects of θ_{ig} also play a role in shaping the hazard function. These effects determine the likelihood of a hyperedge occurring, based on the participation of specific nodes within the hyperedge. In essence, $\boldsymbol{\theta}$ influences whether the event of interest is more or less likely to happen, depending on the composition of nodes within the hyperedge.

Let z_j denote the latent group membership of hyperedge h_j (so that $z_j = g$ if h_j belongs to cluster g), and by $\pi_g = P(Z_j = g)$ the *a priori* latent class membership probability of a hyperedge h_j being assigned to cluster g. According to Andersen & Gill (1982)'s Cox regression model for recurrent events, the contribution to the full likelihood of hyperedge h_j is

$$P(h_j | z_j = g) = \prod_{k=1}^{K_j} \left[\lambda_0(t_{jk})\, e^{\eta_{jg}}\right]^{\delta_{jk}} \exp\left\{-\Lambda_0(t_{jk})\, e^{\eta_{jg}}\right\}, \tag{3.1}$$

where η_{jg} is the linear predictor, $\lambda_0(t)$ is the baseline hazard function and $\Lambda_0(t)$ is the cumulative baseline hazard function. Note that, since $\Lambda_0(t) = -\log(S_0(t))$, we can rewrite (3.1) in terms of the survival function for the reference category (where all the covariates are equal to zero) as

$$
\begin{aligned}
P(h_j|z_j = g) &= \prod_{k=1}^{K_j} [\lambda_0(t_{jk})\, e^{\eta_{jg}}\,]^{\delta_{jk}} \exp\{-(-\log(S_0(t_{jk})))e^{\eta_{jg}}\} \\
&= \prod_{k=1}^{K_j} [\lambda_0(t_{jk})\, e^{\eta_{jg}}\,]^{\delta_{jk}} \exp\{\log(S_0(t_{jk}))e^{\eta_{jg}}\} \\
&= \prod_{k=1}^{K_j} [\lambda_0(t_{jk})\, e^{\eta_{jg}}\,]^{\delta_{jk}}\, S_0(t_{jk})^{e^{\eta_{jg}}}.
\end{aligned} \tag{3.2}
$$

Therefore, the probability of observing a hyperedge $h_j = (x_{1j}, ..., x_{Nj})$ becomes

$$
\begin{aligned}
P(h_j) &= \sum_{g=1}^{G} P(h_j|z_j = g)P(z_j = g) \\
&= \sum_{g=1}^{G} \pi_g \prod_{k=1}^{K_j} [\lambda_0(t_{jk})\, e^{\eta_{jg}}\,]^{\delta_{jk}} \exp\{-\Lambda_0(t_{jk})\, e^{\eta_{jg}}\}\,,
\end{aligned}
$$

and the likelihood function of π, γ and $\boldsymbol{\theta}$ takes the form of

$$
L(\pi, \gamma, \boldsymbol{\theta}; \mathbf{X}) = \prod_{j=1}^{M} \left[\sum_{g=1}^{G} \pi_g \prod_{k=1}^{K_j} [\lambda_0(t_{jk})\, e^{\eta_{jg}}\,]^{\delta_{jk}} \exp\{-\Lambda_0(t_{jk})\, e^{\eta_{jg}}\} \right]. \tag{3.3}
$$

To ensure the model's identifiability, some constraints have to be imposed on the parameters $(\pi, \gamma, \boldsymbol{\theta})$, in particular:

- $\pi = (\pi_1, ..., \pi_G) \in [0, 1]^G$, $\sum_{g=1}^{G} \pi_g = 1$;
- $\gamma = (\gamma_1, ..., \gamma_G)$, $\gamma_g \in \mathbb{R}$ for $g > 1$, $\gamma_1 = 0$;
- $\boldsymbol{\theta} = (\theta_{ig})$, $\theta_{ig} \in \mathbb{R}$.

The TLCA model thus has $2(G - 1) + NG$ parameters, which can be estimated using the EM algorithm.

3.2 Estimation

Since the TLCA model is an extension of the LCA model for hypergraphs, which takes the form of a finite mixture model, the Expectation Maximization algorithm can be employed to estimate the model parameters $(\pi, \gamma, \boldsymbol{\theta})$.

The E-step of the EM algorithm involves computing the expected value of the complete data log-likelihood given in Equation (3.3). The M-step involves maximizing the expected log-likelihood from the E-step.

3.2.1 Algorithm initialization

Let G be the number of latent classes, which is also the number of components in our mixture model, Z_j be the latent membership random variable, $\hat{\mathbf{P}}^{(t)}$ be a matrix whose elements $\hat{p}_{jg}^{(t)}$ contain the estimated probability memberships for each hyperedge at iteration t, so that $\hat{p}_{jg}^{(t)} = P(Z_j = g \text{ at iteration } t)$. Notice that $\sum_{g=1}^{G} \hat{p}_{jg}^{(t)} = 1$ $\forall j \in \left\{1, ..., 2^N\right\}$. Finally, let $\hat{\pi}^{(t)} = (\hat{\pi}_1^{(t)}, ..., \hat{\pi}_G^{(t)})$ be the estimate of $\pi = (\pi_1, ..., \pi_G)$ at iteration t, and $\hat{\gamma}^{(t)} = (\hat{\gamma}_1^{(t)}, ..., \hat{\gamma}_G^{(t)})$ and $\hat{\boldsymbol{\theta}}^{(t)} = (\hat{\theta}_{ig}^{(t)})$ be the estimate of the parameters $\gamma = (\gamma_1, ..., \gamma_G)$ and $\boldsymbol{\theta} = (\theta_{ig})$ at iteration t. We also denote by $\hat{S}^{(t)}$ the estimate of the survival function $S(t)$.

The EM algorithm is initialized by picking a random starting value for $\pi^{(1)}$ and then drawing random initial probability memberships $\hat{\mathbf{P}}^{(1)}$ based on $\pi^{(1)}$ such that $E(p_{jg}^{(1)}) = \pi_g^{(1)}$ $\forall g \in \{1, ..., G\}$. The first M-step consists in finding $\gamma^{(1)}$, $\boldsymbol{\theta}^{(1)}$ and $\hat{S}^{(1)}$ that maximizes the conditional log-likelihood that uses $\hat{\mathbf{P}}^{(1)}$ and $\pi^{(1)}$.

3.2.2 Expectation step

The expectation step entails computing the expected value of the objective function conditionally on the estimated parameters from the last M-step. Practically, in the E-step we update estimates $\hat{p}_{jg}^{(t)}$ based on the latest parameter estimates $\hat{\gamma}^{(t-1)}$, $\hat{\boldsymbol{\theta}}^{(t-1)}$, $\hat{\pi}^{(t-1)}$ and $\hat{S}^{(t-1)}$:

$$
\begin{aligned}
\hat{p}_{jg}^{(t)} &= \frac{\hat{\pi}_g^{(t-1)} \hat{f}_{jg}(t_j)}{\sum_{g=1}^{G} \hat{\pi}_g^{(t-1)} \hat{f}_{jg}(t_j)} \\
&= \frac{\hat{\pi}_g^{(t-1)} \hat{\lambda}_{jg}(t_j) \hat{S}_{jg}^{(t-1)}(t_j)}{\sum_{g=1}^{G} \hat{\pi}_g^{(t-1)} \hat{\lambda}_{jg}(t_j) \hat{S}_{jg}^{(t-1)}(t_j)} \\
&= \frac{\hat{\pi}_g^{(t-1)} \prod_{k=1}^{K_j} \left[\hat{\lambda}_{jg}(t_{jk})\right]^{\delta_{jk}} \hat{S}_{jg}^{(t-1)}(t_{jk})}{\sum_{g=1}^{G} \hat{\pi}_g^{(t-1)} \prod_{k=1}^{K_j} \left[\hat{\lambda}_{jg}(t_{jk})\right]^{\delta_{jk}} \hat{S}_{jg}^{(t-1)}(t_{jk})} \\
&= \frac{\hat{\pi}_g^{(t-1)} \prod_{k=1}^{K_j} \left[\hat{\lambda}_0(t_{jk})\, e^{\hat{\gamma}_g^{(t-1)} + \sum_{1=1}^{N} \hat{\theta}_{ig}^{(t-1)} x_{ij}}\right]^{\delta_{jk}} \hat{S}_{jg}^{(t-1)}(t_{jk})}{\sum_{g=1}^{G} \hat{\pi}_g^{(t-1)} \prod_{k=1}^{K_j} \left[\hat{\lambda}_0(t_{jk})\, e^{\hat{\gamma}_g^{(t-1)} + \sum_{1=1}^{N} \hat{\theta}_{ig}^{(t-1)} x_{ij}}\right]^{\delta_{jk}} \hat{S}_{jg}^{(t-1)}(t_{jk})}.
\end{aligned}
$$

Note that since $\hat{\lambda}_0(t_{jk})$ does not depend on g, it simplifies. Therefore, we don't need to estimate it, and the equation for the E-step becomes:

$$\hat{p}_{jg}^{(t)} = \frac{\hat{\pi}_g^{(t-1)} \prod_{k=1}^{K_j} \left[e^{\hat{\gamma}_g^{(t-1)} + \sum_{1=1}^{N} \hat{\theta}_{ig}^{(t-1)} x_{ij}} \right]^{\delta_{jk}} \hat{S}_{jg}^{(t-1)}(t_{jk})}{\sum_{g=1}^{G} \hat{\pi}_g^{(t-1)} \prod_{k=1}^{K_j} \left[e^{\hat{\gamma}_g^{(t-1)} + \sum_{1=1}^{N} \hat{\theta}_{ig}^{(t-1)} x_{ij}} \right]^{\delta_{jk}} \hat{S}_{jg}^{(t-1)}(t_{jk})}. \tag{3.4}$$

3.2.3 Maximization step

The maximization step consists of maximizing over the model parameters the expectation computed in the E-step:

$$(\hat{\gamma}^{(t)}, \hat{\boldsymbol{\theta}}^{(t)}, \hat{\pi}^{(t)}, \hat{S}^{(t)}) = \arg\max_{\gamma, \boldsymbol{\theta}, \pi, S} \frac{\hat{\pi}_g^{(t-1)} \prod_{k=1}^{K_j} \left[e^{\hat{\gamma}_g^{(t-1)} + \sum_{1=1}^{N} \hat{\theta}_{ig}^{(t-1)} x_{ij}} \right]^{\delta_{jk}} \hat{S}_{jg}^{(t-1)}(t_{jk})}{\sum_{g=1}^{G} \hat{\pi}_g^{(t-1)} \prod_{k=1}^{K_j} \left[e^{\hat{\gamma}_g^{(t-1)} + \sum_{1=1}^{N} \hat{\theta}_{ig}^{(t-1)} x_{ij}} \right]^{\delta_{jk}} \hat{S}_{jg}^{(t-1)}(t_{jk})}. \tag{3.5}$$

For simplicity, we estimate $\hat{\pi}^{(t)}$ using the closed-form expression below, instead of obtaining it in the numeric optimization in Equation (3.5).

$$\hat{\pi}_g^{(t)} \simeq \frac{1}{2^N} \sum_{j=1}^{2^N} \hat{p}_{jg}^{(t)} \quad \forall g = 1, ..., G.$$

$\hat{\gamma}^{(t)}$, $\hat{\boldsymbol{\theta}}^{(t)}$ and $\hat{S}^{(t)}$ are estimated numerically using a weighted Cox model for recurrent events, where the weights are given by $\hat{p}_{jg}^{(t)}$.

3.3 Simulating time-stamped hypergraph data

In this section, we outline the procedure developed to generate the hypergraph data matrix based on the TLCA model that will be employed in the simulation study discussed in Chapter 4.

It consist of the following steps:

1. Set N, M, G;
2. Given N, M, G, fix the model parameters $\pi, \gamma, \boldsymbol{\theta}$.
3. Construct the incidence matrix $\mathbf{X} = (x_{ij})$, $i = 1, ..., N$, $j = 1, .., 2^N$, containing all the possible hyperedges h_j, namely the 2^N permutations of the N binary entries $x_{ij} \in \{0, 1\}$ of the hyperedge vector;

4. For $j = 1, ..., 2^N$, use a pseudo-random number generator from the multinomial distribution to simulate the latent class membership,

$$z_j \sim Multinom_G(\pi);$$

5. Given $\mathbf{X}$ and z, apply the Inverse Transform sampling Method (ITM, Bender et al., 2005) to generate the survival times $t = (t_1, ..., t_{2^N})$ from a Cox model.

   (a) Draw 2^N pseudo-random numbers $u_1, ..., u_{2^N}$ from a uniform distribution;

   (b) For $j = 1, .., 2^N$, set

$$t_j = \Lambda_0^{-1}\left[-\log(u_j)\exp(\gamma_{z_j} + \sum_{i=1}^{N} x_{ij}\theta_{iz_j})\right].$$

   Note that, in our implementation, times are generated from an exponential distribution with shape parameter λ:

$$t_j = -\frac{\log(u_j)}{\lambda\exp(\gamma_{z_j} + \sum_{i=1}^{N} x_{ij}\theta_{iz_j})}.$$

   However, other probability distributions can be used to generate the times-to-event, e.g., the Weibull distribution, the gamma distribution, etc;

6. Apply the following sequential algorithm to identify the M observed events e_m and their respective occurrence times $t^{(m)}$, where $m = 1, ..., M$. Here's how the algorithm operates:
   For $m = 1, ..., M$:

   (a) Determine the index

$$k = \arg\min(t)$$

   corresponding to the minimum value in the vector of simulated times $t = (t_1, ..., t_{2^N})$;

   (b) Set the observed event time $t^{(m)} = t_k$, the corresponding hyperedge $e_m = h_k$, and $\delta_m = 1$;

   (c) Draw a new time

$$t_k^{'} = t_k + t_{\text{new}},$$

where t_{new} is sampled similarly to Step 5 and originates from the same distribution as t_k, ensuring it's associated with the same hyperedge h_j and thus allowing for hyperedge recurrence.

(d) Replace the existing time t_k with $t_k^{'}$ within the t vector.

In the dataset that we obtain from this procedure, each possible hyperedge $(h_j)_{j=1}^{2^N}$ is associated with the pair (t_j, δ_j), where $t_j = (t_{j1}, ..., t_{jK_j})$ and $\delta_j = (\delta_{j1}, ..., \delta_{jK_j})$. It is worth noting that if $k < K_j$, then $\delta_{jk} = 1$, while $\delta_{jk} = 0$ for $k = K_j$. This means that the hyperedge h_j is observed $K_j - 1$ times, and the last time t_{Kj} corresponds to the end of the study, when the hyperedge is right-censored. After obtaining the dataset in wide format, it can be converted into long format, where each observation corresponds to a specific time point within a hyperedge's duration. Specifically, each hyperedge $(h_j)_{j=1}^{2^N}$ will be associated with pairs of start and stop times $(t_{\text{start}}, t_{\text{stop}})$, where t_{start} represents the beginning of the time interval and t_{stop} marks the end of it. This dataset can now be analyzed through a Cox model for recurrent events. The long format dataset comprises $2^N + M - 1$ observations.

Table 3.1 provides an illustration of the long format for a dynamic hypergraph. This hypergraph comprises three nodes, eight possible hyperedges, and twelve observed hyperedges. The first event in this dynamic hypergraph occurs at time $t^{(1)} = 0.1$, and this leads to the observation of the first hyperedge, $e_1 = h_2$. At time $t^{(2)} = 0.8$, the second hyperedge $e_2 = h_7$ occurs, followed by the third event at time $t^{(3)} = 0.9$. This event is also associated with the hyperedge h_7, indicating that we are observing this hyperedge for the second time, therefore $e_3 = h_7$. The same pattern is observed for the hyperedges $e_4, e_5, .., e_{11}$. The final event at time $t^{(12)} = 10$ corresponds to the end of the study, and the last hyperedge $e_{12} = h_5$ is observed. Notably, all hyperedges, even those that have not occurred yet (namely hyperedges h_1, h_3, and h_4) are censored at $t = 10$.

		v_1	v_2	v_3	z	t_{start}	t_{stop}	δ
h_1		0	0	0	1	0	10.0	0
h_2	e_1	1	0	0	2	0	0.6	1
h_2	e_5	1	0	0	2	0.6	3.2	1
h_2	e_7	1	0	0	2	3.2	6.6	1
h_2		1	0	0	2	6.6	10.0	0
h_3		0	1	0	1	0	10.0	0
h_4		1	1	0	1	0	10.0	0
h_5	e_{12}	0	0	1	2	0	10.0	1
h_6	e_4	1	0	1	1	0	1.0	1
h_6	e_{10}	1	0	1	1	1.0	8.6	1
h_6		1	0	1	1	8.6	10.0	0
h_7	e_2	0	1	1	2	0	0.8	1
h_7	e_3	0	1	1	2	0.8	0.9	1
h_7	e_6	0	1	1	2	0.9	3.9	1
h_7	e_8	0	1	1	2	3.9	7.6	1
h_7		0	1	1	2	7.6	10.0	0
h_8	e_9	1	1	1	1	0	8.5	1
h_8	e_{11}	1	1	1	1	8.5	9.3	1
h_8		1	1	1	1	9.3	10.0	0

TABLE 3.1: Time-stamped hypergraph data matrix in long format with $N = 3$ nodes (v), 2^N possible hyperedges (h), $M = 12$ observed hyperedges (e) and $G = 2$ latent classes (z). t_{start} and t_{stop} mark the beginning and the end of the time intervals for each observation, and δ represent the event indicator.

Chapter 4

Simulation study

In this Chapter, we present a Monte Carlo simulation study designed to evaluate the performance of the proposed TLCA model.

Our primary focus is to evaluate the classification performance of the TLCA model, that is, its ability to correctly classify hyperedges into the G latent groups in different scenarios. Thus, the values of two metrics are provided: the *accuracy* and the *Area Under the ROC Curve* (AUC). Accuracy quantifies the percentage of hyperedges correctly classified, calculated as the ratio of the number of correct predictions to the total number of predictions, with a fixed classification threshold set at 0.5. On the other hand, the AUC evaluates the model's classification performance across the entire range of possible threshold values.

Additionally, we want to examine the convergence of the proposed method regarding the latent class membership probability parameter π, the group parameter to model the baseline hazard γ, and the node-specific parameter $\boldsymbol{\theta}$. We aim to assess whether the estimated parameters converge to the true values as the number of observed hyperedges M in the model increases. To do so, we calculate the l_1 distance between the true parameter and the estimated ones, which is the sum of the absolute differences between each element in the parameter divided by its cardinality. It's important to note that in the TLCA model, the number of observations is $2^N + M - 1$. Therefore, it relies on both the number of nodes and the number of observed hyperedges in the hypergraph.

We present the results of a Monte Carlo simulation study consisting of 250 repetitions. For each Monte Carlo repetition, four starting values for the parameter $\pi^{(1)}$ of the EM algorithm are considered, and the one that maximizes the log-likelihood function is chosen. Different simulation scenarios are shown, characterized by different number of nodes in the hypergraph N, of observed hyperedges M, of latent classes G, and model parameters $(\pi, \gamma, \boldsymbol{\theta})$. To perform this analysis, data are generated using the

data-generating procedure described in Section 3.3. In every scenario, the times-to-event are drawn from an exponential distribution with shape parameter $\lambda = 1$. Below are the numerical values assigned to the model parameters.

We consider $G = 2$ latent classes, a number of nodes N equal to $8, 9$ and 10, resulting in $2^8 = 256$, $2^9 = 512$, and $2^{10} = 1024$ hyperedges at risk respectively, and $M = 250, 500, 1000$ and 2000 observed hyperedges. The latent class membership probability parameter π is set to $(\frac{2}{3}, \frac{1}{3})$ in every scenario. We show the results for two different values of γ, $\gamma = (0, 1.3)$ and $\gamma = (0, 0.4)$. The $\boldsymbol{\theta} = (\theta_{ig})$ matrices used for the three values of N are:

$$\boldsymbol{\theta}_{8\times2} = \begin{bmatrix} -0.127 & 0.173 \\ -0.055 & 0.230 \\ 0.264 & -0.273 \\ 0.017 & 0.235 \\ 0.031 & -0.026 \\ 0.274 & -0.028 \\ 0.107 & 0.044 \\ -0.238 & 0.240 \end{bmatrix}; \boldsymbol{\theta}_{9\times2} = \begin{bmatrix} -0.127 & 0.173 \\ -0.055 & 0.230 \\ 0.264 & -0.273 \\ 0.017 & 0.235 \\ 0.031 & -0.026 \\ 0.274 & -0.028 \\ 0.107 & 0.044 \\ -0.238 & 0.240 \\ -0.152 & -0.275 \end{bmatrix}; \boldsymbol{\theta}_{10\times2} = \begin{bmatrix} -0.127 & 0.173 \\ -0.055 & 0.230 \\ 0.264 & -0.273 \\ 0.017 & 0.235 \\ 0.031 & -0.026 \\ 0.274 & -0.028 \\ 0.107 & 0.044 \\ -0.238 & 0.240 \\ -0.152 & -0.275 \\ -0.103 & 0.273 \end{bmatrix}.$$

4.1 EM algorithm behaviour

Before evaluating the performance of the TLCA model, we need to check the reliability of the EM algorithm. The E-step of the algorithm entails computing the expected value of the log-likelihood function conditionally on the estimated parameters from the last M-step. In the maximization step, the parameters are updated by maximizing the conditional mean derived in the E-step. This algorithm proceeds iteratively until convergence, thus maximizing the log-likelihood function.

In Figure 4.1 are some graphs confirming the monotonically increasing trend of the log-likelihood at each step of the EM algorithm. We observe that the log-likelihood always stabilizes towards the last iterations, and the convergence of the algorithm is faster when the number of observed hyperedges M is higher.

Similarly to the likelihood, the parameter estimates also tend to stabilize before the EM algorithm converges. Figure 4.2 illustrates the trend in estimating the parameter π_1. Four initial values (0.2, 0.4, 0.6, and 0.8) for the parameter estimator $\pi^{(1)}$ of the EM algorithm are considered. For each combination of (N, M), the estimators tend to

converge towards the true value of $\pi_1 = \frac{2}{3}$ or $\pi_2 = \frac{1}{3}$. This occurs because the EM algorithm estimates the parameter $\pi = (\pi_1, \pi_2)$, but label switching can occur. Label switching refers to the phenomenon where the assignments of components in a mixture model can change during iterations, leading to the exchangeability of component labels. It is evident that as the number of hyperedges M in the hypergraph increases, the convergence of the parameter becomes faster and more accurate.

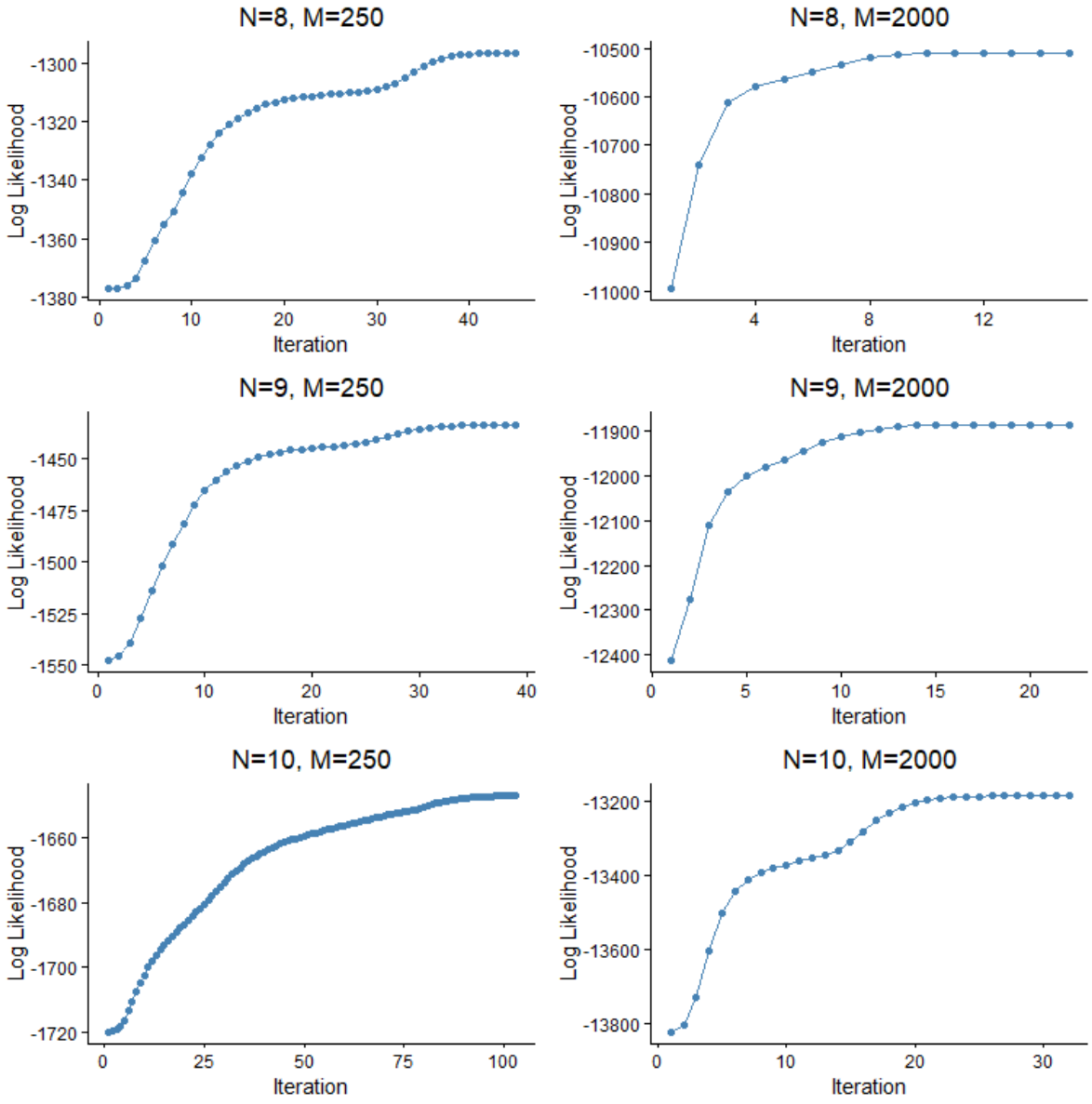

FIGURE 4.1: Log-likelihood behaviour of the EM algorithm in simulations with $\gamma = (0, 1.3)$ and different values of N and M. Each of the four lines represents a distinct starting point for the parameter estimator $\pi^{(1)}$.

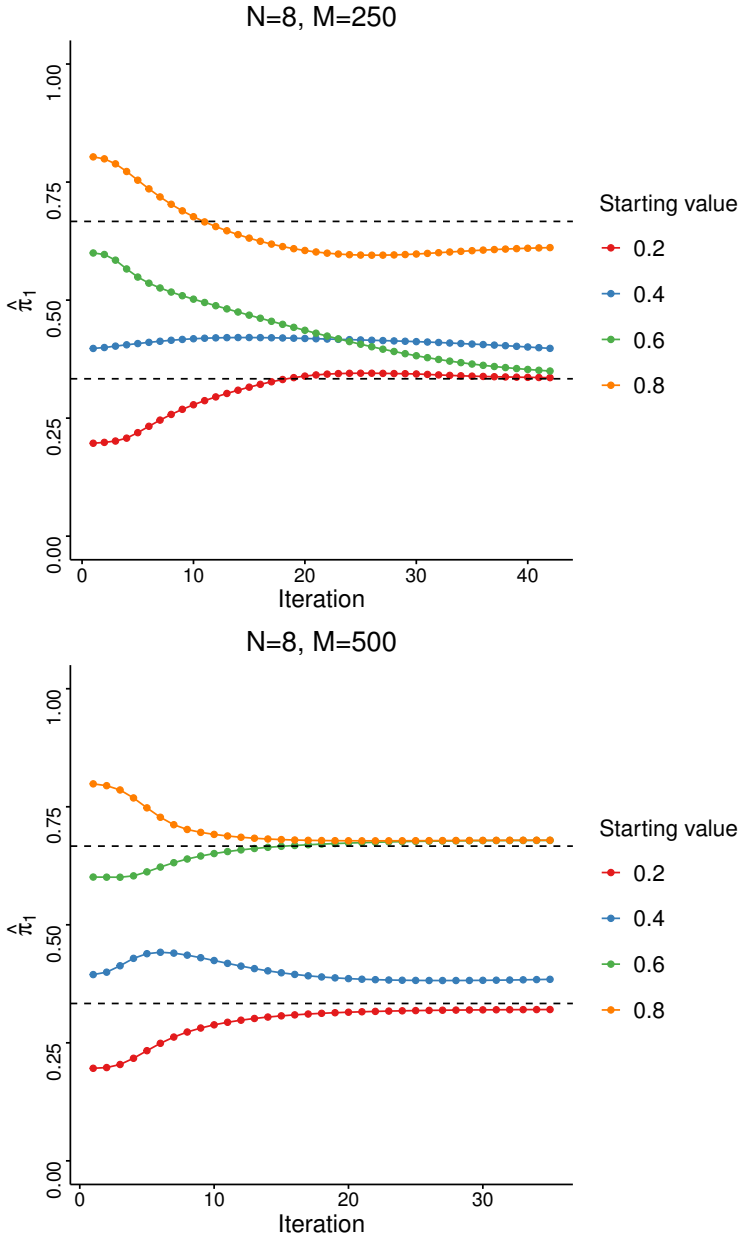

FIGURE 4.2: $\hat{\pi}_1$ behaviour of the EM algorithm in simulations with $\gamma = (0, 1.3)$ and different values of N and M. The horizontal lines represent the values of the true parameter $\pi_1 = \frac{2}{3}$ and $\pi_2 = \frac{1}{3}$.

4.2 Clustering accuracy

Our primary interest lies in the classification property of the TLCA model. In Figure 4.3, as the number of observed hyperedges M grows, we observe a corresponding increase in the proportion of accurate predictions when $\gamma_2 = 1.3$. As we expect, this trend is less pronounced if $\gamma_2 = 0.4$. This difference arises from the effect of γ_g, $g = 1, ..., G$, on the baseline hazard of the hyperedges belonging to latent class g, which is multiplied by a factor of $\exp(\gamma_g)$. Here, the likelihood of observing a hyperedge in group 2 is higher if

$\gamma_2 = 1.3$ compared to $\gamma_2 = 0.4$. Therefore, a higher value of γ_g helps to more clearly distinguish between the different groups.

In our analysis, we also examine the AUC of the TLCA model, as depicted in Figure 4.4. Here, we observe patterns similar to those seen in the accuracy analysis. Specifically, with an increase in the number of observed hyperedges, there is a corresponding rise in the AUC value when $\gamma_2 = 1.3$. Conversely, this trend is less prominent when $\gamma_2 = 0.4$.

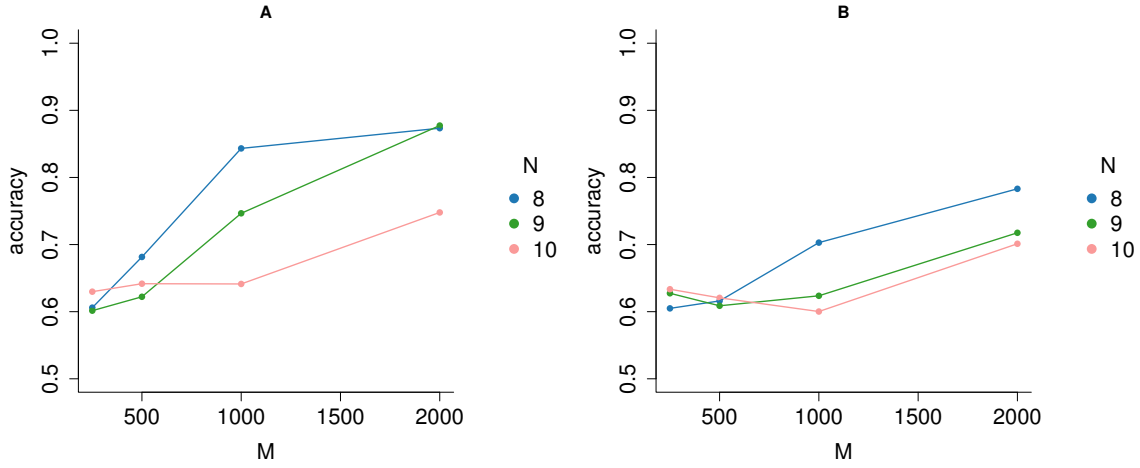

FIGURE 4.3: TLCA model accuracy with different values of N and M. In the left Figure, γ_2 is set to 1.3, while in the right one $\gamma_2 = 0.4$.

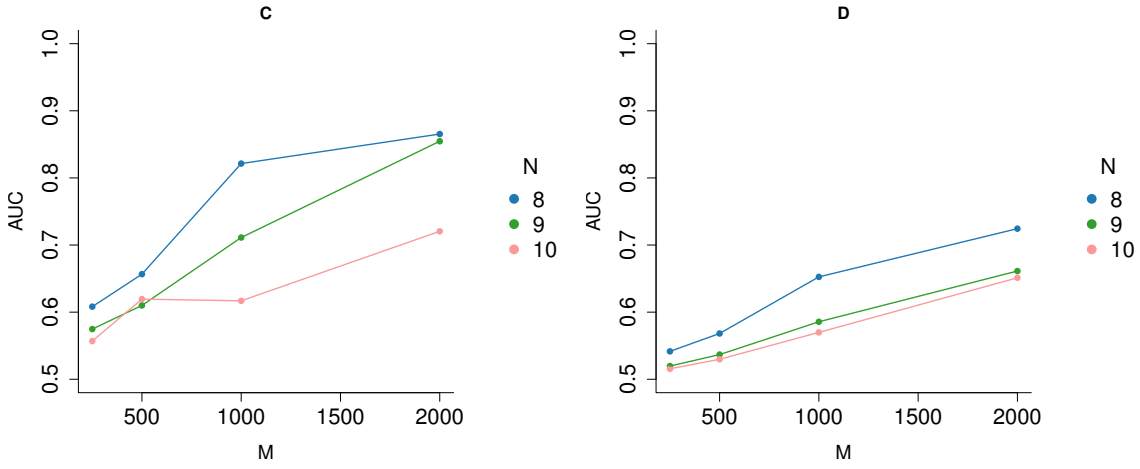

FIGURE 4.4: TLCA model AUC with different values of N and M. In the left Figure, γ_2 is set to 1.3, while in the right one $\gamma_2 = 0.4$.

4.3 Convergence analysis of the model parameters

In this section, we want to assess the convergence of the model parameters. As depicted in Table 4.1 and Table 4.2, we observe that, for fixed values of the number of nodes

N, as the number of observed hyperedges M, and consequently the overall number of observations, increases, the estimated parameters $\hat{\pi}$, $\hat{\gamma}$, and $\hat{\boldsymbol{\theta}}$ tend to converge toward their true values.

On the other hand, we also note that as the number of nodes increases, the discrepancy between the estimated parameters and their true values tends to widen. This observation necessitates careful consideration when comparing scenarios with different values of N, given that our model encompasses all hyperedges at risk, leading to a rapid escalation in the number of observations with increasing N.

Finally, it is evident that the estimation of parameter $\pi = (\pi_1, \pi_2)$ can be significantly improved by increasing the value of γ_2 from 0.4 to 1.3. This behavior can be explained by the fact that a higher value of γ_2 makes the hyperedges belong to two latent classes that are more distinguishable, as mentioned earlier.

N	M	$l_1(\pi)$	$l_1(\gamma)$	$l_1(\boldsymbol{\theta})$	accuracy	AUC
8	250	0.067	4.161	1.207	0.606	0.608
	500	0.076	1.574	0.485	0.682	0.657
	1000	0.046	0.501	0.223	0.843	0.821
	2000	0.046	0.440	0.187	0.873	0.865
9	250	0.072	5.267	1.167	0.602	0.575
	500	0.064	2.767	0.738	0.622	0.610
	1000	0.058	0.727	0.271	0.747	0.711
	2000	0.032	0.254	0.153	0.877	0.855
10	250	0.096	5.454	1.190	0.630	0.557
	500	0.081	5.337	1.090	0.642	0.619
	1000	0.060	2.160	0.512	0.641	0.617
	2000	0.057	0.868	0.281	0.748	0.720

TABLE 4.1: Convergence analysis of the EM algorithm for the TLCA model with $G = 2$ latent classes and $\gamma = (0, 1.3)$. $l_1(\pi)$, $l_1(\gamma)$ and $l_1(\boldsymbol{\theta})$ refers to the l_1 distance between the true parameters and the estimated ones. The model accuracy and AUC are also shown.

N	M	$l_1(\pi)$	$l_1(\gamma)$	$l_1(\boldsymbol{\theta})$	accuracy	AUC
8	250	0.081	1.724	0.640	0.605	0.542
	500	0.080	0.943	0.428	0.616	0.568
	1000	0.080	0.405	0.233	0.703	0.653
	2000	0.050	0.205	0.142	0.783	0.724
9	250	0.081	2.815	0.814	0.627	0.520
	500	0.088	1.539	0.532	0.609	0.537
	1000	0.091	0.716	0.320	0.624	0.586
	2000	0.069	0.285	0.182	0.718	0.661
10	250	0.077	4.295	1.011	0.633	0.516
	500	0.083	2.387	0.716	0.621	0.530
	1000	0.082	1.214	0.437	0.600	0.570
	2000	0.078	0.376	0.232	0.701	0.651

TABLE 4.2: Convergence analysis of the EM algorithm for the TLCA model with $G = 2$ latent classes and $\gamma = (0, 0.4)$. $l_1(\pi)$, $l_1(\gamma)$ and $l_1(\boldsymbol{\theta})$ refers to the l_1 distance between the true parameters and the estimated ones. The model accuracy and AUC are also shown.

Chapter 5

Discussion

In this work, we developed a model-based clustering approach for the analysis of dynamic hypergraphs, named Temporal LCA (TLCA). The proposed methodology is a generalization to time-dependent hypergraphs of the Latent Class Analysis model for static hypergraphs proposed by Ng & Murphy (2021). In the TLCA model, the hazard rate of the hyperedges is dependent on the group parameters $\gamma_g = (\gamma_1, ..., \gamma_G)$, the node-specific parameters $\boldsymbol{\theta} = (\theta_{ig})$ and the baseline hazard $\lambda_0(t)$. The TLCA model follows the customary approach of survival analysis, where not only the hyperedges occurring over the study period are considered, but also the ones that could have potentially happened, and are therefore considered at risk to occur. The model aims to investigate which factors affect the occurrence and timing of hyperedges in dynamic hypergraphs. To achieve this, we adapted a mixture of Cox regression models for recurrent events to the dynamic hypergraph framework.

As the TLCA model takes the form of a finite mixture model, we implemented the Expectation-Maximization algorithm to estimate the model parameters. The E-step involves computing the expected value of the complete data log-likelihood to derive the estimated probability memberships for each hyperedge conditionally on the parameters from the last M-step. In the M-step, the parameters are updated by maximizing the conditional expectation derived in the E-step.

We proposed a procedure to generate the hypergraph data matrix, where we implemented a sequential algorithm to identify the observed hyperedges and their respective occurrence times over all the possible hyperedges at risk, allowing for hyperedge recurrence.

We conducted a simulation study to evaluate the performance of the TLCA model. The results revealed that as the number of observed hyperedges in the hypergraph increases, there is a corresponding increase in the proportion of accurate predictions.

This trend is more prominent when the group parameter of the Cox model is higher. The model parameters tend to converge to their true value as the number of observed hyperedges increases. It is worth noting that our simulation study was limited to $G = 2$ latent groups due to time constraints, but the model is extendable to $G > 2$ groups.

Our TLCA model represents a first attempt to model dynamic hypergraphs using a model-based clustering approach. This model can be the basis for future extensions and alternative modelling approaches.

One of the challenges associated with the TLCA model is its computational cost, which increases exponentially as the number of nodes in the hypergraph grows. Specifically, the sample size increases by a factor of 2^N as the number of nodes N increases, since all possible hyperedges are considered at risk. Indeed, the data matrix in Table 3.1 becomes huge as N increases. To address this issue, Chapter 4 simulations were conducted with a relatively small number of nodes and with a limited number of Monte Carlo repetitions. While this limitation makes it difficult to analyze hypergraphs of larger sizes, one potential solution is to employ nested case-control sampling (CCS, Vu et al., 2015) to select the at-risk hyperedges. This technique involves randomly sampling an unobserved hyperedge for each observed hyperedge, reducing the number of hyperedges that become part of the model from 2^N to $2M$. As a result, the estimation procedure becomes much more efficient.

One other extension could involve the development of a more parsimonious modeling approach, similar to the ELCA model proposed by Ng & Murphy (2021). This could be achieved through the introduction of a double clustering level, where the two independent clustering structures would be employed to capture the fluctuations in the size of the hyperedges.

Furthermore, the TLCA model could be reformulated to semparate the timing of hyperedges and the decision of nodes to join an event/hyperedge. The Cox model could capture the event times that arise from G different densities based on their latent class assignment, while the Bernoulli distribution could model the likelihood of nodes joining a hyperedge. This alternative model specification relaxes the proportionality assumption of the Cox model, allowing for greater flexibility in modeling event times that are not necessarily proportional to each other. Moreover, parametric distributions such as the exponential or the Weibull distribution can also be employed to model the event times as an alternative to the Cox model.

Finally, a possible extension to the TLCA model could involve integrating two independent clustering structures for the hyperedges and the nodes. This approach could provide a more comprehensive and accurate representation of the underlying patterns

of many real-world phenomena.

Bibliography

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control* **19**, 716–723.

Andersen, P. K. & Gill, R. D. (1982). Cox's regression model for counting processes: a large sample study. *The Annals of Statistics* , 1100–1120.

Bender, R., Augustin, T. & Blettner, M. (2005). Generating survival times to simulate Cox proportional hazards models. *Statistics in Medicine* **24**, 1713–1723.

Brusa, L. & Matias, C. (2022). Model-based clustering in simple hypergraphs through a stochastic blockmodel. *arXiv preprint arXiv:2210.05983* .

Chodrow, P. S., Veldt, N. & Benson, A. R. (2021). Generative hypergraph clustering: From blockmodels to modularity. *Science Advances* **7**, eabh1303.

Collins, L. M., Fidler, P. L., Wugalter, S. E. & Long, J. D. (1993). Goodness-of-fit testing for latent class models. *Multivariate Behavioral Research* **28**, 375–389.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society, Series B: Methodological* **34**, 187–202.

Crane, H. & Dempsey, W. (2018). Edge exchangeable models for interaction networks. *Journal of the American Statistical Association* **113**, 1311–1326.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B: Methodological* **39**, 1–22.

Frank, O. & Strauss, D. (1986). Markov graphs. *Journal of the American Statistical association* **81**, 832–842.

Goodman, L. A. (1974). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika* **61**, 215–231.

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

Holland, P. W. & Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association* **76**, 33–50.

Hunter, D. R. & Lange, K. (2004). A tutorial on MM algorithms. *The American Statistician* **58**, 30–37.

Karoński, M. & Łuczak, T. (2002). The phase transition in a random hypergraph. *Journal of Computational and Applied Mathematics* **142**, 125–135.

Kolaczyk, E. D. (2009). *Statistical Analysis of Network Data*. Springer.

Lazarsfeld, P. & Henry, N. (1968). *Latent Structure Analysis*. Houghton Mifflin.

Lerner, J. & Lomi, A. (2023). Relational hyperevent models for polyadic interaction networks. *Journal of the Royal Statistical Society, Series A: Statistics in Society* **186**, 577–600.

Meng, X.-L. & Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika* **80**, 267–278.

Moreno, J. L. (1934). *Who shall survive? A new approach to the problem of human interrelations*. Nervous and Mental Disease Publishing.

Ng, T. L. J. & Murphy, T. B. (2021). Model-based clustering for random hypergraphs. *Advances in Data Analysis and Classification* , 1–33.

Nylund, K. L., Asparouhov, T. & Muthén, B. O. (2007). Deciding on the number of classes in latent class analysis and growth mixture modeling: A Monte Carlo simulation study. *Structural Equation Modeling: A Multidisciplinary Journal* **14**, 535–569.

Rizzo, M. L. (2019). *Statistical Computing with R*. CRC Press.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics* , 461–464.

Stasi, D., Sadeghi, K., Rinaldo, A., Petrović, S. & Fienberg, S. E. (2014). β models for random hypergraphs with a given degree sequence. *arXiv preprint arXiv:1407.1004* .

Vu, D., Pattison, P. & Robins, G. (2015). Relational event models for social learning in MOOCs. *Social Networks* **43**, 121–135.

Wang, Y. J. & Wong, G. Y. (1987). Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association* **82**, 8–19.