# Analysis of User Behavior

of the Faculty of Business & Economics
of Leuphana University of Lüneburg

in fulfillment of the requirements for the degree of
Doctor of Natural Sciences
– Dr. rer. nat. –

approved dissertation by
Ahcène Boubekki

born on May 22, 1987
in Tizi-Ouzou, Algeria

Submitted on: February 14, 2020
Viva-voce defense on: July 8, 2020

First supervisor and reviewer: Prof. Dr. Ulf Brefeld
Second reviewer: Prof. Dr. Robert Jenssen
Thrid reviewer: Prof. Dr. Hendrik Drachsler

The individual contributions to the thesis by publication within the framework of the doctoral procedure are or will be published, as the case may be, including the context chapters as follows:

- A. Boubekki, U. Kröhne, F. Goldhammer, W. Schreiber, and U. Brefeld. Toward Data-Driven Analyses of Electronic Text Books. *Proceedings of the International Conference on Educational Data Mining*, 2015.
- A. Boubekki, U. Kröhne, F. Goldhammer, W. Schreiber, and U. Brefeld. Data-driven analyses of electronic text books. *In Solving Large Scale Learning Tasks. Challenges and Algorithms*, 2016.
- S. Mair, A. Boubekki, and U. Brefeld. Frame-based Data Factorizations. *Proceedings of the International Conference on Machine Learning*, 2017.
- J. Reubold, A. Boubekki, T. Strufe, and U. Brefeld. Infinite Mixtures of Markov Chains. *New Frontiers in Mining Complex Patterns*. 2018.
- A. Boubekki, S. Jain, and U. Brefeld. Mining User Trajectories in Electronic Text Books. *Proceedings of the International Conference on Educational Data Mining*, 2018.
- A. Boubekki and U. Brefeld. Mining Trajectories. Submitted to *Data Mining and Knowledge Discovery*, 2019.
- A. Boubekki, M. Kampffmeyer, R. Jenssen, and U. Brefeld. Theoretically Grounded Centroid-based Deep Clustering. Submitted to the *International Conference on Machine Learning*, 2020.

Year of publication: 2020

Published online on the website of the University Library:
`http://www.leuphana.de/ub`

# Abstract

Online behaviors analysis consists of extracting patterns from server-logs. The works presented here were carried out within the "mBook" project which aimed to develop indicators of the quantity and quality of the learning process of pupils from their usage of an eponymous electronic textbook for History. In this thesis, we investigate several models that adopt different points of view on the data. The studied methods are either well established in the field of pattern mining or transferred from other fields of machine learning and data-mining.

We improve the performance of archetypal analysis in large dimensions and apply it to unveil correlations between visibility time of particular objects in the e-textbook and pupils' motivation. We present next two models based on mixtures of Markov chains. The first extracts users' weekly browsing patterns. The second is designed to process sessions at a fine resolution, which is *sine qua non* to reveal the significance of scrolling behaviors. We also propose a new paradigm for online behaviors analysis that interprets sessions as trajectories within the page-graph. In this respect, we establish a general framework for the study of similarity measures between spatio-temporal trajectories, for which the study of sessions is a particular case. Finally, we construct two centroid-based clustering methods using neural networks and thus lay the foundations for unsupervised behaviors analysis using neural networks.

**Keywords:** online behaviors analysis, educational data mining, Markov models, archetypal analysis, spatio-temporal trajectories, neural network

# Zusammenfassung

Die Online-Verhaltensanalyse beschäftigt sich mit der Extraktion von Mustern aus Server-Logs. Die hier vorgestellten Arbeiten wurden im Kontext des „mBook“-Projekts durchgeführt, das zum Ziel hat, Indikatoren für Qualität und Quantität von Lernprozessen von Schülern zu entwickeln, die auf deren Nutzung eines elektronischen Lehrbuchs für das Fach Geschichte basieren. Wir untersuchen mehrere Modelle, die unterschiedliche Sichtweisen auf die Daten einnehmen. Die verwendeten Methoden sind entweder bereits im Gebiet des pattern mining etabliert oder wurden aus anderen Bereichen des maschinellen Lernens und des Data Mining übertragen.

Wir verbessern die Leistungsfähigkeit der Archetypenanalyse für hochdimensionale Daten decken mit ihrer Hilfe Zusammenhänge zwischen der Sichbarkeitszeit von bestimmten Objekten im elektronischen Lehrbuch und Lernmotivation der Nutzer auf. Wir stellen außerdem zwei Mixturmodelle auf der Basis von Markow-Ketten vor. Das erste dient zur Extraktion von Mustern im wöchentlichen Browsing-Verhalten der Nutzer. Das zweite verarbeitet Sessions auf eine feiner-granulären Ebene, und erlaubt so, bedeutsame Verhaltensweisen im Scrolling aufzuzeigen. Wir stellen des Weiteren ein neues Paradigma der Online-Verhaltensanalyse vor, das Sessions als Trajektorien von Nutzern im Seitengraph interpretiert. In dieser Hinsicht schaffen wir einen Rahmen für die Untersuchung von Maßen für die Ähnlichkeit von räumlich-zeitlichen Trajektorien, in welchem die Analyse von Sessions einen Spezialfall darstellt. Schlussendlich demonstrieren wir zwei Clusteringverfahren mittels zentroidbasierter neuronaler Netze und legen damit die Grundlagen für unüberwachte Mustererkennung unter Verwendung neuronaler Netze.

**Schlüsselwörter:** Online-Verhaltensanalyse, Bildungsdatenanalyse, Markow-Modelle, Archetypenanalyse, räumlich-zeitlich Trajektorien, künstliches neuronales Netz

# Acknowledgment

I would like to acknowledge you reader that I am very thankful to a lot of people that helped me to bring this thesis to an end. Essentially, everyone who was there from the beginning or who joined along the way.

Alles begann mit einer Frage über Markov-Ketten und guten Sound auf der Tanzfläche. Deswegen geht mein erster Danke an Ulf, der an mich geglaubt hat, als ich nur Mathematiker war. Ich antworte hier deine Frage.

And when I say you, I mean all of you: the whole team from KMA to ML3 (just for the rime in french). Especially, Seb to whom I owe more than a crate of beers, Daniel who will one day manage a good coffee, Samuel who is Brazilian but still in the team, Shailee who we can still hear talking, the LaTeX lady, and the always helpful Madlen and Tanja. Of course, I do not forget Maryam and Hamid without whom these years in LG would have been more morose and the introduction never done. I also would like to thank the team in Tromsø for being so patient and great: Robert, Michael, Jonas, Kristoffer, Luigi, and all the others.

For good or bad, research does not stop at the door of the office, not because I was not there that often (I see you Vincent, Christoph and Felix), but because many outside of it influenced me and inspired me. You can count Willy the Swiss, Méri con quien ciertamente no bailo, Laura y Ilias o los 31 saccadas, Laure et le fromage, Conchi und Judith wer mich zum tanzen gebracht haben, und Hamdi, weil das Leben ist zu kurz. I would like to thank in particular Christoph: only with him can Hamburg hold its reputation.

It does not make sense to thank the city of Frankfurt. I need to thank those who make it so beautiful. From Przemo, to Làszló and Theo: thank you all for all these unforgettable experiences, mustaches, and Polonez; Steffi who we don't see; Gitti and Max this summer we'll see; Ewelina and her ewelineries!; Guylaine parce que bien sûr; Avi because you're me; Neha who dompted the klugsch* but great Shiv! Et enfin Michele pour l'exemple. Do polskich naukowców Ewa, Kitek, i Rafał: podobasz mi się!

Sache lecteur, qu'il y a des matheux qui me voient encore comme l'un d'eux: l'Alain au ski, le mathématicien-philosophe Thomas, King Malick, Thibau*d* le Grand, Maÿlis peut être la prochaine fois, et enfin maman Giulia.

Si tu me redemande maintenant quand tout a commencé, je te dirais il y a dix-sept ans. Cette époque avant Palalaguna, Pacha, Julia, les rallys photo à travers Paris ou sur les toits de Rennes, et les quatre saisons du Québec. Une époque sans Pascal mais Wesley, où Luce n'avait pas les cheveux bouclés mais tonton Julien déjà tonton. À l'époque le kamarade Paul tournait déjà autour de la belle Maud et Jérème ne connaissait pas encore Audrey. Ça paraît être hier, et quelle joie!

Bien sûr la vérité est que ça a commencé il y a trente deux ans, avec mes frères Amirouche et Hocine avec qui on s'embrouille toujours pour rien. Et enfin mon père, ma mère: oui, ce n'est pas facile de faire des hommes libres.

Tanemirt.

# Contents

# Notation

### Mathematical Notation

| | |
|---|---|
| $\mathbb{N}$ | set of non negative integers |
| $[a \,..\, b]$ | set of integers between the integers $a$ and $b$, with $a \leq b$ |
| $\mathbb{R}$ | set of real numbers |
| $\mathbb{R}_{\geq 0}$ | set of non negative real numbers |
| $\mathbb{R}_{+}$ | set of positive real numbers |
| $\mathbb{R}_{\geq 0}^{+\infty}$ | set of the postivie real number extended with the infitiny |
| $\mathbb{R}^p$ | real vector-space of dimension $p$ or set of real seqeunces of length $p$ |
| $\mathbb{R}^\infty$ | set of infinite real sequences |
| $\mathbb{R}^{p\times q}$ | set of real matrices of dimension $(p, q)$ |
| $x \in \mathbb{R}$ | a real number |
| $\boldsymbol{x}, \boldsymbol{y}, \ldots$ | vectors or sequences |
| $\langle x_1, \ldots, x_p \rangle$ | vector of dimension $p$ or a sequence of length $p$ |
| $\langle x_i \rangle_{1\leq i\leq p}$ | idem |
| $\langle x_i \rangle_p$ | idem |
| $\langle x_i \rangle_{\mathbb{N}}$ | infinite sequence |
| $\mathbf{0}_p,\ \mathbf{1}_p,\ \mathbf{2}_p, \ldots$ | $= \langle 0 \rangle_p,\ \langle 1 \rangle_p,\ \langle 5 \rangle_p, \ldots$ |
| $\boldsymbol{A}, \boldsymbol{B}, \ldots$ | matrices |
| $\mathbf{I}_p$ | identity matrix of $\mathbb{R}^{p\times p}$ |
| $\mathbb{S}^p$ | $= \{\boldsymbol{x} \in \mathbb{R}_{\geq 0}^p : \sum_{i=1}^p x_i = 1\}$, the stochastic vectors of $\mathbb{R}^p$ |
| $\mathbb{S}^\infty$ | infinite stochastic sequences |
| $\mathbb{1}_S, \mathbb{1}_a$ | indicator functions of a set $S$ and of the interval $[0, a]$ |
| $\delta_{uv}$ | Kronecker delta, $\delta_{uv} = 1$ if $u = v$ and 0 otherwise |

### Abbreviation

| | |
|---|---|
| iid | independent and identically distributed |
| pdf | probability density function |
| iif | if, and only if |
| wrt | with respect to |

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Since humans are rational animals [9], the study of their behavior, be it online, may uncover personality traits or intentions. Online Behavior Analysis (OBA) is part of the broader field of pattern mining. It aims to extract information relevant to a specific application from the online traces left by users. It has proven successful in several domains, such as online advertisement [64], e-commerce [45, 158], or streaming services [227]. It has also been used in network security, where the access to critical resources can be granted or not depending on the users activity [153, 70]. In our case, we focus on applications to educational science.

The emerging field of educational data mining (EDM) [16, 15] offers multiple use-cases for OBA. The one that interests us here is the study of pupils' usage of a History electronic textbook, called the *mBook*. Interestingly enough, little is known about the impact of electronic aids on learning. As we shall see, the analysis of pupils' behaviors can reveal patterns that correlate with different levels of competency or motivation. In a long run, OBA for EDM could also be used as a tool for teachers to better grasp the dynamic within a class group, and for educational specialists to evaluate pedagogical approaches.

The research presented here contributes to OBA in general and with a focus on educational data mining. We propose several approaches with different perspectives on the data collected from the *mBook*, which yields equally diverse insights into the pupils' behaviors. Without going into too much technicalities, we present in the following an introduction to online behavior analysis. We then give some key information about the "mBook" project. After a survey of the research in the fields of OBA and EDM, we outline our contributions.

## 1.1 Online Behavior Analysis

The term "behavior", i.e., the way a person acts, is an intuitive and subjective concept that is sadly not prone to a formal definition. Yet, "the set of actions a person undertakes" may serve as an approximation, albeit with shortcomings. First, in order to be computationally processed, these actions must be encoded in a digital format, resulting necessarily in a loss of information. Furthermore, they can only be interpreted within a context. Consider a user clicking repetitively and frantically on a "refresh" button to refresh an online feed (e.g., news or social network) . Depending on the context, this sequence of clicks very close in time could be interpreted as excitement or frustration (e.g., due to a small bandwidth).

Online Behavior Analysis splits into three steps: representation, modeling and clustering. One could also add a step for interpretation and exploitation of the results, but it is usually carried out in a second time. Nevertheless, by whom and how the outcomes are used affects several upstream choices. Prior to reviewing each step of OBA, we discuss issues related to the collection and pre-processing of the data.

### Data and Pre-Processing

The base material of OBA are the traces of the interactions between the clients and a web-service. The data is stored on the server-side in the so-called *log-files*. The granularity of the tracking and description of events may vary with the technology used. However, the data must contain at least keys to identify the action, the user, the session, and the time the event occurred. Time is a transversal dimension in the data. It appears as connection times, which can be used to distinguish behaviors in class or at home. As we shall see, speed of scrolls, which is computed from the events' timestamps, characterizes different types of pupils .

Events can only be fully interpreted within a context. A challenge is, therefore, to capture as much of it as possible in the logs in a format that can be later transformed into features by a pre-processing of the data. For example, not all pages in the mBook have the same learning potential. It is, therefore, interesting to include the list of objects visible by the users. In practice, the granularity of the data collection comes as a trade-off. If on one hand, a large population and a precise context are required for finer analyses and to give statistical guarantees. On the other hand, this means larger log-files, which lead to storage and computational scalability issues.Privacy issues are also a concern [201], particularly since the introduction of the European General Data Protection Regulation, or GDPR [216]. It is thus necessary to decide

beforehand on a compromise between available resources, privacy protection, and the granularity of the tracking. Especially, since these choices affect the implementation of the web-service, the pre-processing of the log-files, as well as their analysis.

### Representations, Models and Clusterings

Modeling starts with the choice of a data representation. That is the structure given to the raw data which emphasizes specific characteristics of the data. The model itself can then be described as a set of assumptions between these characteristics. For example, logs can be represented as chronologically ordered sequences of events. A common model using such a representation assumes that past events influence the future ones. Note that representation and models are not always dissociated, since the later usually implies the former. We chose to stress the distinction since several models use the same type of representation.

Naive session models, such as *bag-of-words* [247], omit the sequential nature of the data and only consider the frequencies of events separately or in tuples. This is troublesome because shuffling data does not change the frequency of events, but changes the latent behaviors. In any case, probabilistic approaches are preferable. They put distance between the model and the data and, therefore, better generalize.

Regarding sequence models, there are two points of view: sessions can be modeled either as ongoing processes or as complete realizations thereof. The first approach qualified as *local*, focuses on the transitions between events. The relevance of an observation depends only on the previous ones: if a user opens a new page, she will likely start scrolling through it. As a result, the probability of observing an entire session is proportional to that of the last event. In contrast, for *global* approaches the probability of an observation also depends on the future events: if a user opens a gallery, she has likely clicked and scrolled several times in the past. Note that these two points of view generally yield different results. One shortcoming of sequence-based models is that a single model governs the entire sequence. This is equivalent to assuming that users consistently maintain the same behavior throughout the session, which is not necessarily true.

Grouping or *clustering* similar behaviors allows to further generalize the analysis. To do this automatically, one needs a measure of similarity. However, there is as yet no unequivocal definition. In the case of web-sessions, it is not even clear how to compare them. It could based on their duration, the content viewed, or the actions. A standard solution is to embed the sessions into a vector space where the distances are well studied and defined. However, learning such an embedding can be tedious.

Another approach is to rely on mixtures of probabilistic models. The idea is to assume that sessions are realizations of several of models. That way optimizing the model also returns a clustering.

## 1.2 The "mBook" Project

Most of the works that we present here have been done in the context of the "mBook" project [225]. The cooperation between educational specialists for History, educational psychologists, and computer scientists spread between the academic years 2013-2014 and 2016-2017. The objective of this pioneering project in educational science in Germany was to evaluate the impact of an electronic textbook, called *mBook*, on the learning process of secondary school pupils.

Traditional evaluations of textbooks include interviews and analyses of in-class video recordings [102]. Log-files allow for less intrusive evaluation methods that reduce the bias induced by the presence of a researcher in the classroom and guarantee the anonymity of the users. Nevertheless, the analysis of the log-files does not replace usual protocols but complements them. For example, it is almost impossible to decide from the logs alone if there is more than one pupil in front of the screen.

#### An Electronic Textbook

The structure of *mBook* resembles to its paper-counterparts. The page graph of the website is a tree with extra edges to allow a linear reading. Five chapters cover Antiquity, Middle Age, Renaissance, 19th century, the 20th and 21st centuries together, plus a chapter on methods.

Figure 1.1 shows a mock-up of a generic page of the mBook. It consists of a succession of text, galleries, audios, videos, and interactive information boxes. Galleries comprise pictures related to the text. Some audio or video files are directly integrated into the web-page and can be visualized from there. Expandable information boxes provide additional information or exercises. A navigation bar is always visible at the bottom of the screen. Users can reach the previous or next page (like in a paper-based book), or jump to the summary of the current section (central button). The bar also doubles as a toolbar for adding notes or highlighting parts of the text.

#### The Data-set

Throughout the thesis, we restrict the study of the mBook to a single period ranging from January 31st to July 11th 2017. This corresponds to 2,197 sessions from 400


Figure 1.1: Mock-up of the mBook.

users, of whom 195 are pupils (537 sessions) who have passed a standardized test in July 2017 that evaluates five psychometric factors: competencies, knowledge, and motivation for History, as well as access and skills with information and communication technologies (ICT). The competencies and knowledge scores [152] are estimated using a 1-PL model [79]. The last three are inferred from MCQ tests. The distributions of the scores are displayed in Figure 1.2.

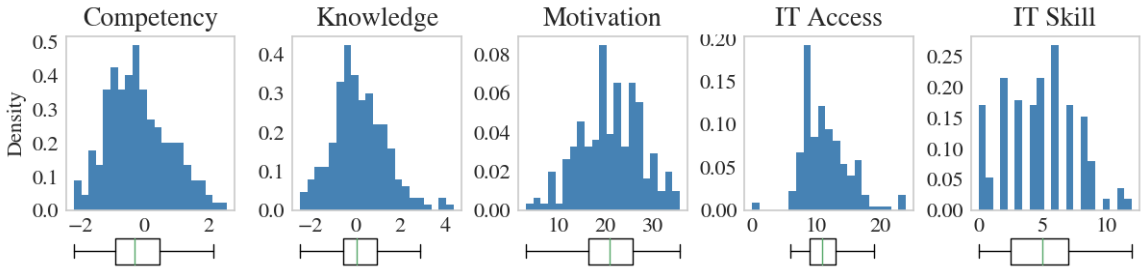



Figure 1.2: Distributions of the psychometric scores and their respective box-plots.

## 1.3 Literature Review

The analysis of online behavior has accompanied the spread of the Internet and is part of the broader field of pattern mining. One of the first models called *sequential pattern mining* is based on the *apriori* algorithm [5] and identifies the most frequent tuples of observations. It has a wide range of applications in the field of web mining [69, 186, 232, 190, 135, 3, 212]. Association rules [4] condition the frequency of the co-occurrences on the frequency of one element of the pair, introducing that a way some causality. The confidence of the rule $A \to B$ is equal to the frequency of the tuple $(A, B)$ divided by that of the singleton $A$. A famous application of this model is the prediction of shopping carts [47, 266, 74, 75]. Rule construction extends to more than two objects, resulting in chains of association rules. Markov chains can be considered as their probabilistic equivalents [54]: the expected transition probability between two observations mirrors the confidence an association rule.

Markov chains are standard to model user behaviors [54, 183, 81, 38]. The hidden Markov model (HMM) [21] does not model observations but latent, or hidden, states that in turn emit the observations. It is used to solve many problems [63, 154, 189, 182], including for behavior analysis [260, 138, 103]. Several models derive from it. For example, the infinite hidden Markov model [20, 242] integrates a Dirichlet process to let the optimization find the most appropriate number of hidden states given the

data. The hidden semi-Markov model [198, 261] combines two layers of HMM: one governs the latent states; the other one governs the number of observations emitted by these states. To some extent, we can consider that recurrent neural networks [221] generalize Markov models. In particular, the memory of the long-term memory cells (LSTM) [128] or recurrent gated units (GRU) [66] resembles a dynamic order of a Markov chain but that evolves with the data.

We identify three phases, throughout the history of online behavior analysis, In its early days, the focus was on *personalizing* the user experience. This implies acting on the layout of the website [54, 184, 191, 118] or on the results of the search engines [69, 119]. The latter has opened a second phase under the drive of recommender systems [64, 158, 227]. Although collaborative filtering remains the standard approach in the field [45, 201], recent methods make greater use of context [265] and neural networks [90, 125, 249]. Nonetheless, the benefits of these so-called *neural recommendations* are being questioned [72]. In its most recent developments, research in OBA focuses on the analysis of social networks [252, 25, 111], and in particular, on the detection of bots [76, 244].

Use-cases of OBA in educational research reflect "regular" applications. Although frequentist approaches are still frequently used [78, 39], advanced machine learning and data mining techniques are increasingly being employed for EDM [217, 15].

One specificity of educational sciences is the opportunity to have high-quality labeled of the data as well as information outside of the logs [147, 136, 259]. For example, the analysis of student's curricula, represented as sequences of courses and exams taken by students, has been used to predict grades or to customize future curricula. Computerized adaptive testing raises issues that are reminiscent of general recommender systems [173]. The first is to predict a student's success based on sequences of assigned or selected questions [120]. Another is the customization of these sequences to improve performance or influence the learning process [27, 151]. Lastly, the problem of robot detection finds an equivalent in the analysis of learning behaviors in massive open online courses (MOOC) [96, 103], especially to detect and explain dropouts [39].

## 1.4 Outline and Contributions

The "mBook" project is an ideal framework for the transfer of classical methods from OBA to EDM. Nevertheless, we also develop and improve data-mining techniques for

behavior analysis in general. The work presented here makes thus several genuine theoretical contributions in the fields of machine learning and data-mining.

Chapter 2 contains a definition of general dissimilarity measure and an introduction to mixture models. The remaining chapters are organized according to an increasing complexity of the representation of the sessions. In each, we take a different stand on the data and propose a model accompanied with an application using the *mBook* data. Chapter 3 is dedicated to archetypal analysis where sessions correspond to points on a simplex. In Chapters 4 and 5, we propose two Markov chains models that can leverage different context and level of granularity of the data. A new perspective on user behaviors is investigated in Chapter 6 where sessions are represented as spatio-temporal trajectories within the page graph. Finally, Chapter 7 serves as an opening. We present there two neural networks architectures for centroid-based clustering which constitute our first step toward a deep learning approach for online behavior analysis.

In the following, we give a short summary of each chapter and of their contributions.

**Archetypal Analysis and Content Analysis**

Previous works have measured motivation based on the actions of online learners [126, 143]. We go further with regard to two aspects. Firstly, we do not focus on events but on the visibility times of certain contents, which are only accessible through a careful pre-processing of the data. Secondly, we show how to use archetypal analysis (AA) in this context. A naive factor analysis leads to a single statistically significant correlation between time spent in galleries and motivation. It is not much. Archetypal analysis extracts three more correlations.

The idea behind archetypal analysis is to represent data-points as convex combinations of factors, or *archetypes*, lying on the convex hull of the data-set. This allows for a straightforward interpretation of the factorization, but at the cost of inefficient calculations. To alleviate this issue, we show that the factorization can be efficiently computed by a quadratic program, namely the active-set method of Lawson and Hanson [161]. We prove that this algorithm also identifies the set of points of the convex hull, called the *frame*, which is considered a complex problem. In an effort to improve the scalability of AA, we propose an approximation restricting the whole optimization to the frame only. The heuristic is that a good approximation of the latter also gives a good approximation of the data. On the downside, the non-unique and sparse solutions returned by the active-set algorithm might hinder the interpretation of the

reconstruction weights. We propose, thus, a method to compute dense representations. Empirical evaluations of the novel method yield similar reconstruction errors as baseline competitors while being faster to compute. This speed-up is particularly beneficial for model selection.

### Markov Chain and Periodic Behaviors

Markov models are standard methods in behavior analysis [54] due to their interpretatibility. The underlying idea exploits the sequential nature of user behaviors and translates user sessions into Markov processes, i.e., observations depend on their predecessors. Regrettably, most of the approaches based on Markov chains focus on the pure sequence of events, without taking into account contextual information. Haider et al. [117] include temporal dependencies like daily or weekly periodicity. We refine their approach by using truly periodic distributions and conditioning the observation of a page on the chapter. We derive an Expectation-Maximization-based algorithm (EM) [80] to cluster users and their sessions according to their behavior. While $k$-means produces trivial and insignificant groups, our methodology successfully discovers the main navigation patterns. The analysis of the user clustering over a week suggests that behaviors may also be influenced by the teachers.

### Bayesian Markov Chains and Scrolling Behaviors

The EM falls short on two aspects. Firstly, as greedy optimization strategy it may lead to poor local optima, requiring several random initializations. Second, a model selection based on information criteria fails if the model is too complex or when the number of instances is too low, which is common in educational science. We, therefore, develop the infinite mixtures of Markov chains (iMMC) to avoid these shortcomings. Our model extends the hierarchical Dirichlet process (HDP) to Markov processes: one Dirichlet process governs the cluster assignments and another one models the Markov transitions. Computations are eased thanks to a degree $k$-weak limit approximation [133]. Our empirical study of scrolling patterns within the mBook, used as a model for reading style, reveals correlations with psychometric scores.

### Trajectories for Online Behaviors

Since the same vocabulary can describe movements in a museum and online, we propose to apply the same methods for both. To the best of our knowledge, this is the first time that such an approach has been adopted. Spatio-temporal data is ubiquitous, but the theory is often application dependent. For example, there is no prevailing

definition of a similarity measure between trajectories. We propose a formalization of the study of spatio-temporal trajectories as well as an unambiguous classification of their similarity measures. We obtain theoretically grounded properties, which are never all satisfied by existing measures. To fill this gap, we devise a novel measure based on the Laplace distributions and the Kullback-Leibler divergence. It is equivalent to the normalized point-wise distance with a penalty for different duration of the trajectories. Empirically, we observe that our measure performs better or on par with state-of-the-art competitors while having a linear time complexity and being robust to sampling rates and time units. We further derive behavior indicators from trajectory clusters that characterize the dynamic within a class group and the teaching style.

**Deep Clustering as a Unifying Method**

The temptation to use deep learning is great particularly to leverage its ability to learn efficient representations for multiple tasks. With this contribution, we take the first step to merge modeling and clustering of online behavior, reducing many choices to that of an architecture. We present an end-to-end (deep) clustering network. The network in its simplest form consists of a two layer autoencoder (AE), where the loss function is derived from Gaussian mixture models (GMM). A deep variant can be obtained by adding more layers for expressivity so that the loss function sums up the reconstruction losses of an AE and a relaxation of the GMM-based loss. On average, our models empirically outperform traditional clustering techniques like $k$-means and GMMs and also perform better or equal to existing (deep) clustering architectures while being less reliant on pre-training.

## 1.5 Previously Published Work

Some works presented in this thesis have already been published or are under review. In the following we list them and give a brief summary of the respective contributions to the papers.

(1) A. Boubekki, U. Kröhne, F. Goldhammer, W. Schreiber, and U. Brefeld. Toward Data-Driven Analyses of Electronic Text Books. *Proceedings of the International Conference on Educational Data Mining*, 2015.

(2) A. Boubekki, U. Kröhne, F. Goldhammer, W. Schreiber, and U. Brefeld. Data-driven analyses of electronic text books. *In Solving Large Scale Learning Tasks. Challenges and Algorithms*, 2016.

These works build upon a previous work of Haider et al. [118]. I extended it with the true periodic Gaussian based distribution and adapted it to the chapter/page/gallery structure of the mBook. I also provided the implementation and carried out the experiment.

(3) S. Mair, A. Boubekki, and U. Brefeld. Frame-based Data Factorizations. *Proceedings of the International Conference on Machine Learning*, 2017.

Together with Sebastian Mair, we investigated archetypal analysis. We tested several approaches that never came into fruition, before Seb looked into NNLS. I reviewed the main proof and helped with the experiments.

(4) J. Reubold, A. Boubekki, T. Strufe, and U. Brefeld. Infinite Mixtures of Markov Chains. *New Frontiers in Mining Complex Patterns*. 2018.

Jan Reubold had some previous work on mixture models and Dirichlet processes. For this work, I helped with the formalization, the implementation, and provided the application on the mBook.

(5) A. Boubekki, S. Jain, and U. Brefeld. Mining User Trajectories in Electronic Text Books. *Proceedings of the International Conference on Educational Data Mining*, 2018.

(6) A. Boubekki and U. Brefeld. Mining Trajectories. Submitted to *Data Mining and Knowledge Discovery*, 2019.

I supervised Shailee Jain during her bachelor thesis investigating my idea to represent online behaviors as trajectories. In (5), we worked together to list requirements that a measure should satisfy to obtain interpretable clusters. In (6), I revamped the idea of the trajectories and built the theory of trajectory measures from scratch. The development of the theory made clear the need for a new classification scheme of trajectory measures. I also derived a new measure that extends the one used in (5) and provided larger set of applications.

(7) A. Boubekki, M. Kampffmeyer, R. Jenssen, and U. Brefeld. Theoretically Grounded Centroid-based Deep Clustering. Submitted to the *International Conference on Machine Learning*, 2020.

I uncovered the relation between GMM and autoencoder. Michael helped me to formalize the presentation in terms common to the deep learning community.

# Chapter 2

# Preliminaries

In the first section, we provide a definition for a dissimilarity measure on a set and introduce the notion of a point at infinity. The subsequent sections are dedicated to the mixtures of Markov chains (MMC) and their inference. Section 2.2 recalls standard definitions. In Section 2.3, we introduce the mixtures of Markov chains and discuss their optimization based on the Expectation-Maximization algorithm [80]. In the following Section 2.4, we take a Bayesian perspective on the MMC, discuss the Gibbs sampler [104] as an inference algorithm for Bayesian mixtures, and its similarities with the EM. Section 2.5 reviews Dirichlet processes and their integration into mixture models. Using an explicit indexation of a Dirichlet process, we show that the usual transition from a finite mixture to an infinite mixture [23] is degenerate. The chapter ends with the theorem of Ishwaran and Zarepour [133], which shows that Bayesian mixing models can be used to approximate Dirichlet process based mixture models.

Note that the sections dealing with the mixtures of Markov chains are in purpose written in a less formal tone. They are designed to serve as lecture notes and target students in Master with basic knowledge in Probability [35]. Each section starts with a paragraph describing motivations and heuristics involved. It is followed by a more rigorous presentation punctuated with remarks.

## 2.1 Dissimilarity Measures and Point at Infinity

To cluster automatically objects, we need to estimate their similarity. Although the task is omnipresent in data mining, there is no consensus on the definition. In this section, we fix the definition of a dis/similarity measure valid throughout the manuscript. Moreover, we introduce the notion of point at infinity for a measure, that will prove handy to model missing data or the end of a sequence.

Throughout, the section, $(\Omega, d)$ is the metric space of all the possible observations, equipped with a distance $d$. The space can be discrete (e.g., the set of the nodes of a graph and the shortest path distance) or a real vector space with the euclidean distance.

**Dissimilarities**

We base our definitions on the metric axiomatic [229].

**Definition 2.1** (Dissimilarity)**.** *A* dissimilarity measure*, or* semi-metric*, on* $\Omega$ *is a bivariate function* $d : \Omega^2 \to \mathbb{R}_{\geq 0}$ *satisfying the following conditions for any elements* $x$*,* $y$*,* $z$ *of* $\Omega$*:*

$$\begin{aligned} &d(x,y) \geq 0 \quad \textit{and} \quad d(x,x) = 0 && \textit{(non-negativity)},\\ &d(x,y) = d(y,x) && \textit{(symmetry)},\\ &d(x,y) = 0 \Leftrightarrow x = y && \textit{(identity of indiscernibles)}. \end{aligned}$$

*If* $d$ *also satisfies the triangle inequality, it is called a* distance *or* metric*:*

$$d(x,z) \leq d(x,y) + d(y,z) \qquad \textit{(triangle inequality)}.$$

From a semantic point of view, dissimilarities and similarities have an inverse behavior: analogous objects have a small dissimilarity but a high similarity. We formalize here this intuition and bound similarity measures between 0 and 1.

**Definition 2.2** (Similarity)**.** *A* similarity measure *on* $\Omega$ *is a bivariate symmetric function* $s : \Omega^2 \to [0\,,1]$ *such that:*

$$\forall (x,y) \in \Omega^2,\ s(x,y) = 1 \Leftrightarrow x = y.$$

In accordance with the generalization function of Shepard [229], the inverse of the exponential of a dissimilarity is a similarity.

**Lemma 2.1** ([229])**.** *If* $d$ *is a dissimilarity measure on* $\Omega$*,* $\exp(-d)$ *is a similarity measure on* $\Omega$*.*

The conversed is also true.

**Lemma 2.2.** *If* $s$ *is a similarity measure on* $\Omega$*,* $-\log(s)$ *is a dissimilarity measure on* $\Omega$*.*

It is often handy to add an auxiliary element to $\Omega$ to account for missing or corrupted data. This state is virtually inaccessible from normal point, hence we model it as a *point at infinity*.

**Definition 2.3** (Point at Infinity)**.** *A dissimilarity on $\Omega$ can be extended to $\mathbb{R}_{\geq 0}^{+\infty}$ by adding a point to $\Omega$, called a* point at infinity *or an* infinite point*, noted $\infty$, such that for any element $x$ of $\Omega$:*

$$x \neq \infty \Leftrightarrow d(x, \infty) = +\infty \quad \textit{and} \quad d(\infty, \infty) = 0.$$

*Consequentially, for a similarity $s$ it holds: $\forall x \in \Omega,\ x \neq \infty,\ s(x, \infty) = 0$.*

Note that, the value of the point at infinity depends on the measure. The formalization of the notion of dissimilarity if further studied in Chapter 6, where we propose a new classification of dissimilarity measures for trajectories.

## 2.2 Mixture Models and Markov Chains

Consider the task of prolonging the sequences of the following data-set:

$$\mathcal{S} = \left\{ \begin{array}{ll} \boldsymbol{s}_1 = AABAABAAA, & \boldsymbol{s}_2 = AAABAABAA, \\ \boldsymbol{s}_3 = BBABBABBB, & \boldsymbol{s}_4 = BBBABBABB \end{array} \right\}$$

A naive approach consists of drawing randomly the new events using the transition's frequencies between elements. The frequency contingency table $\mathrm{Cont}(\mathcal{S})$ organizes row-wise these frequencies: a cell gives the transition's frequency in $\mathcal{S}$ from the element indexing the row to the one indexing the column:

$$\mathrm{Cont}(\mathcal{S}) = \begin{array}{|c||c|c|} \hline & A & B \\ \hline\hline A & .5 & .5 \\ \hline B & .5 & .5 \\ \hline \end{array}$$

The table is uninformative, although a pattern is clearly apparent in the data: the two first and two last sequences are mirrored. This observed pattern is well captured by the contingency tables for each pair:

$$\mathrm{Cont}(\{\boldsymbol{s}_1, \boldsymbol{s}_2\}) = \begin{array}{|c||c|c|} \hline & A & B \\ \hline\hline A & .67 & .33 \\ \hline B & 1 & 0 \\ \hline \end{array} \qquad \mathrm{Cont}(\{\boldsymbol{s}_3, \boldsymbol{s}_4\}) = \begin{array}{|c||c|c|} \hline & A & B \\ \hline\hline A & 0 & 1 \\ \hline B & 33 & .67 \\ \hline \end{array}$$

Therefore, although $\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \boldsymbol{s}_4$ appear together in the same data-set, we would use different contingency tables to augment the two first and two last sequences. In other

word, we assume that the data was generated by two different processes. This idea is the basis of *mixture models*, and in particular, since we deal with sequences, *mixtures of Markov chains* (MMC).

Optimizing, or *learning*, a mixture of models, amounts to determining its different *generative processes*, or *models*. *Maximum likelihood estimation* (MLE) is a basic solution: it maximizes the *likelihood* that the data was generated by the mixture. The calculations are generally simple and the *Expectation-Maximization* (EM) algorithm [80] provides an efficient optimization scheme. However, the method does not yield any statistical guarantee of the results. As a remedy, it can be assumed that the models are governed by parameters drawn from a priori known distributions. The optimization problem becomes thus to learn the *prior distributions* whose expected values are the parameters that maximize the likelihood of the data. For a careful choice of prior distributions, the calculations are straightforward and the expected value and variance have closed forms. Such a view on the problem is said *Bayesian*, while the associated optimization scheme is called *maximum a posteriori* estimation (MAP).

One aspect has not yet been raised: the number of models, or *components*. In the example, two components seem to be a reasonable choice. However, four is also a good answer. To find the best value, a grid search combined with specific criteria does the job, but it is ineffective. An advanced solution involves some stochasticity via the use of *Dirichlet processes*.

Although mixture models are often used for clustering, this is not their raison d'être. Their purpose is to explain how the data were generated. Groups of instances most likely generated by the same model do indeed constitute a clustering of data, but this is a by-product.

In the following, we first give formal definitions of a stochastic process, a Markov chain, and a mixture model. Next, we describe the mixture of Markov chains, the Bayesian approach, and their respective inference using MLE and MAP. Finally, we discuss the use of Dirichlet processes. Note that similar explanations and constructions can be used for any mixture, e.g., Gaussian mixture models.

### 2.2.1 Definitions

To avoid ambiguities, we recall a list of basic definitions.

**Definition 2.4** (Random Variable)**.** *A* random variable *(RV) is a function from a probability space [208] into a measurable space.*

*The* realization *of a random variable, also referred to as an* observation*, is the outcome of a random draw of a possible value of the RV with respect to its probability distribution.*

Although, the Bayes' rule is usually considered as a theorem, we use it as the name of a formula. That way, we include it in a Definition and do not present a proof [35].

**Definition 2.5** (Bayes' rule)**.** *Let $\mathcal{X}$ be a set of observations (data-set) and $\Theta$ a set of parameters (of a model), the following formula is called the* Bayes' rule*:*

$$p(\Theta|\mathcal{X}) = \frac{p(\mathcal{X}|\Theta)\, p(\Theta)}{p(\mathcal{X})}. \tag{2.1}$$

*The terms in the formula have specific names.*

- $p(\mathcal{X}|\Theta)$*:* likelihood *(of the data-set given the model),*
- $p(\Theta|\mathcal{X})$ *:* posterior *probability of* $\Theta$*,*
- $p(\Theta)$ *:* prior *probability of* $\Theta$*,*
- $p(\mathcal{X})$ *:* evidence*.*

*More generally, the adjectives* posterior *and* prior *indicate if the probability is conditioned on $\mathcal{X}$ or not.*

Just like the multinomial distribution generalizes the binomial distribution, the Dirichlet distribution generalizes the Beta distribution.

**Definition 2.6** (Dirichlet distribution)**.** *A Dirichlet distribution of order $K > 1$ with parameter $\boldsymbol{\alpha} \in \mathbb{R}^K_{\geq 0} \setminus \{\mathbf{0}_K\}$, is noted* $\mathrm{Dir}(\boldsymbol{\alpha})$*. The pdf on $\boldsymbol{x} \in \mathbb{S}^K$ has for value*

$$\mathrm{Dir}(\boldsymbol{\alpha})(\boldsymbol{x}) = \frac{1}{\mathrm{Beta}(\boldsymbol{\alpha})} \prod_{k=1}^{K} x_k^{\alpha_k - 1}, \tag{2.2}$$

*where* $\mathrm{Beta}(\cdot)$ *is the Beta function [35]. If $\boldsymbol{\alpha} = \alpha \mathbf{1}_K$, the distribution is said* symmetric *and noted* $\mathrm{Dir}(\alpha)$*.*

The aggregation property is a key feature of the Dirichlet distribution.

**Lemma 2.3** (Aggregation)**.**

$$\langle X_1, X_2, X_3 \rangle \sim \mathrm{Dir}(\alpha_1, \alpha_2, \alpha_3) \Rightarrow \langle X_1, X_2 + X_3 \rangle \sim \mathrm{Dir}(\alpha_1, \alpha_2 + \alpha_3). \tag{2.3}$$

This property generalizes to any order.

### 2.2.2 Markov Chains

Formally, the term "Markov chain" designates a *discrete stochastic process* satisfying the *Markov condition*. Let us define each of these terms.

**Definition 2.7.** *Let $(\mathcal{U}, \mathcal{A}, p)$ be a probability space, $\mathcal{I}$ an arbitrary set called* index set*, and $\mathcal{M}$ a measurable set called* sample set*. A* stochastic process SP *indexed by $\mathcal{I}$ with value in $\mathcal{M}$ is the set of random variables:*

$$SP = \{X_t : \mathcal{U} \to \mathcal{M},\ \forall t \in \mathcal{I}\}. \tag{2.4}$$

*If $\mathcal{I}$ is discrete, the process is said* discrete*. If the random variables follow the same distribution, the latter gives the process its name.*

**Definition 2.8.** *A* Markov Chain of order $n$ *is a discrete stochastic process that satisfies the Markov property, i.e., without loss of generality $\mathcal{I} \subset \mathbb{N}$ and :*

$$\forall t \geq n,\ p(X_{t+1}|X_t, \ldots, X_1) = p(X_{t+1}|X_t, \ldots, X_{t+1-n}). \qquad \textit{(Markov Property)}$$

*A chain is fully defined by the initial probability $p(X_1)$ and the* transition *probabilities $p(X_{t+1}|X_t, \ldots, X_{t+1-n})$. If the latter is constant with respect to $t$, the Markov chain is said* homogeneous*.*

Remark that the Markov property induces an order on the process' random variables, such that they actually form a sequence. Therefore, a sequence of observations is the realization of all the process' random variables. In online behavior analysis, we usually make the following assumptions:

- The process is indexed by the positive integers: $\mathcal{I} = \mathbb{N}$;
- The sample set $\mathcal{M}$, or *alphabet*, is included in $\Omega$ and is finite with cardinal $M$;
- The alphabet contains a point noted $\infty$ that marks the end of an observed sequence, which leads to the notion of length:
  $$\text{A sequence } \langle X_i \rangle_{\mathbb{N}} \text{ is of length } T \in \mathbb{N}, \text{ if } \forall t \leq T,\ X_t \neq \infty;$$
- The Markov chain is of order one:
  $$\forall t \in \mathbb{N},\ p(X_{t+1}|X_t, \ldots, X_1) = p(X_{t+1}|X_t);$$
- The Markov chain is homogeneous:
  $$\forall (t, \tau) \in \mathbb{N}^2,\ p(X_{\tau+1}|X_\tau) = p(X_{t+1}|X_t);$$

- The transitions follow categorical distributions with parameter the row-vectors of the row-stochastic matrix $\theta$:

$$\forall a \in \mathcal{M},\ \theta(m, \cdot) \in \mathbb{S}^M,$$
$$\forall (t, a) \in \mathbb{N} \times \mathcal{M},\ (X_{t+1} | X_i = a) \sim \mathrm{Cat}\left(\langle \theta(a, m) \rangle_{\mathcal{M}}\right) = \mathrm{Cat}\left(\theta(a, \cdot)\right).$$

We add two assumptions of our own:

(1) Sequences are prepended with $\infty$;

(2) No sequence is empty, i.e., of length 0.

These two choices have broader consequences. On the bright side, sequences always start the same way, hence the initial probability of any element of $\mathcal{M}$ is null except for $\infty$ for which it is 1. The first real observation is thus obtain by transitioning from the infinite point, i.e., the first observation is drawn from a categorical distribution with parameters $\theta(\infty, \cdot)$. This assumption has the practical benefit that only the transition probability matrix, $\langle \theta(m, m') \rangle_{\mathcal{M} \times \mathcal{M}}$, needs to be estimated. On the other hand, it theoretically challenges the notion of length of a sequence. Indeed, assumption (3) implies that the self-transition probability of $\infty$ is zero. Therefore, once an $\infty$ marking the end of a sequence is drawn, the next event is necessarily different from $\infty$ and another sequence starts. Consequently, sequences never end. That is why we defined the length of a sequences as the index of the last non infinite observation.

If our assumptions are theoretically troublesome, their practical advantages, be it on legibility and implementation, largely compensate.

### 2.2.3 Mixture Models

A mixture of models is a *hierarchical generative model* that develops as follows: First, select a component/mixture; Second, Generate an observation from the its model.

Let us consider a finite mixture with $K \in \mathbb{N}$ components whose models are parameterized on a space $\Xi$, with parameters $\boldsymbol{\theta} = \langle \theta_k \rangle_K \in \Xi^K$. The random variable $z$, modeling the assignment to a component, follows a categorical distribution with parameter

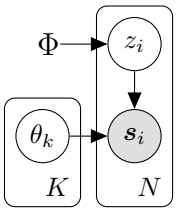



Figure 2.1: Graphical model of a general mixture model.

$\Phi = \langle \phi_k \rangle_K \in \mathbb{S}^K$. The latter, also called *mixture weights*, represent the prior probabilities of the components. The generation of an observation $s$ is described as follows:

$$\begin{aligned} z &\sim \mathrm{Cat}(\Phi), \\ s &\sim \mathrm{Model}(\theta_z). \end{aligned} \tag{2.5}$$

A plate diagram is given in Figure 2.1. Fitting a mixture model to a data-set boils down to learning the mixture weights and the parameters of each model, $\Theta = \{\Phi, \boldsymbol{\theta}\}$. If necessary, a clustering can then be derived from the assignments.

## 2.3 Mixture of Markov Chains

Assuming previous notation, the generation of a sequence $\boldsymbol{s} = \langle s_t \rangle_{\mathbb{N}}$ from a mixture of Markov chains (MMC) is described as follows:

$$\begin{aligned} z &\sim \mathrm{Cat}\Big(\Phi\Big), \\ s_1|\infty &\sim \mathrm{Cat}\Big(\theta_z(\infty, \cdot)\Big), \\ s_t|s_{t-1} &\sim \mathrm{Cat}\Big(\theta_z(s_{t-1}, \cdot)\Big). \end{aligned} \tag{2.6}$$

Figure 2.2 depicts a plate diagram of the model. We introduce now more notation to define the model from its likelihood.

Let $\mathcal{S} = \{\boldsymbol{s}_i\}_N$ be a set of $N$ iid sequences. Each sequence $\boldsymbol{s}_i = \langle s_t^{(i)} \rangle_{1 \leq t \leq T_i}$ of length $T_i \in \mathbb{N}, T_i > 0$ is defined over a finite alphabet $\mathcal{M}$ (including $\infty$) of cardinality $M$. The random variables of the sequences' assignments are regrouped in $\mathcal{Z} = \{z_i\}_N$. The likelihood of a sequence $\boldsymbol{s}$ is:

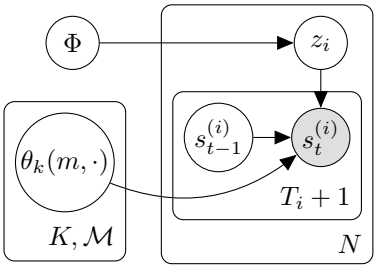



Figure 2.2: Graphical model of a mixture of Markov chains.

$$\mathcal{L}(\boldsymbol{s}; \Theta) = p(\boldsymbol{s}|\Theta) = \sum_{k=1}^{K} p(z = k|\Phi) \prod_{t=1}^{T+1} p(s_t|s_{t-1}, z = k, \boldsymbol{\theta}) = \sum_{k=1}^{K} \phi_k \prod_{t=1}^{T+1} \theta_k(s_{t-1}, s_t), \tag{2.7}$$

where $\Theta = \{\Phi, \boldsymbol{\theta}\}$. This formula is an abuse of notation, because $\boldsymbol{s}$ does not appear as a random variable. Recall that an observed sequence is a realization of a Markov chain. Thus, a rigorous way to write the likelihood is:

$$\mathcal{L}(\boldsymbol{s}; \Theta) = p(X_1 = s_1, \ldots, X_{T_i} = s_{T_i}, X_{T_i+1} = \infty, \ldots |\Theta). \tag{2.8}$$

Because it is too cumbersome, we avoid the proper notation and use the imperfect one.

### 2.3.1 Expectation-Maximization

The *maximum likelihood estimation* (MLE) of a mixture model lends itself well to the *Expectation-Maximization* (EM) algorithm [80, 253], that we describe in this section. Let us recall first *Jensen's inequality* as it will be crucial in the following derivations.

**Lemma 2.4** (Jensen's inequality)**.** *Let $f$ be a real concave function, $\langle a_k \rangle_K \in \mathbb{S}^K$ a stochastic vector, for any $K$ points $x_1, \ldots, x_K \in \mathbb{R}$:*

$$\sum_{k=1}^{K} a_k f(x_k) \leq f\Big( \sum_{k=1}^{K} a_k x_k \Big), \tag{2.9}$$

*with equality if $x_1 = \ldots = x_K$ or $f$ is linear.*

#### $\mathcal{Q}$-function

An MLE aims to find the set of parameters $\Theta$ maximizing the log-likelihood:

$$\ell\left(\mathcal{S}; \Theta\right) = \log p\left(\mathcal{S}|\Theta\right) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} p\left(\boldsymbol{s}_i, z_i = k|\Theta\right) \tag{2.10}$$

Unfortunately, the sum inside the log makes the problem intractable. It can be taken out using Jensen's inequality at the cost of having a lower-bound instead. To achieve this, we introduce the line stochastic matrix $\Gamma \in \mathbb{R}^{N \times K} = \langle \gamma_{ik} \rangle_{\substack{1 \leq i \leq N \\ 1 \leq k \leq K}}$ such that for any $i \in [1 \,..\, N]$, $\langle \gamma_{ik} \rangle_K \in \mathbb{S}^K$. Jensen' inequality implies that:

$$\begin{aligned} \ell\left(\mathcal{S}; \Theta\right) &= \sum_{i=1}^{N} \log \sum_{k=1}^{K} \gamma_{ik} \frac{p\Big(\boldsymbol{s}_i, z_i = k|\Theta\Big)}{\gamma_{ik}} \\ &\geq \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \log \frac{p\Big(\boldsymbol{s}_i, z_i = k|\Theta\Big)}{\gamma_{ik}} = \mathcal{Q}(\Theta, \Gamma). \end{aligned} \tag{2.11}$$

This inequality holds for any choice of $\Gamma$. Therefore, for a fixed set of parameters $\Theta^*$, the function $\Gamma \mapsto \mathcal{Q}(\Theta^*, \Gamma)$ can be maximized using the Lagrange multipliers [35]. Using the stochasticity of $\Gamma$, we recognize that the maximum is reached when the $\hat{\gamma}_{ik}$ are equal to the posteriors of the $z_i$:

$$\hat{\gamma}_{ik} = \frac{p(\boldsymbol{s}_i, z_i = k|\Theta)}{\sum_{l=1}^{K} p(\boldsymbol{s}_i, z_i = l|\Theta)} = p(z_i = k|\boldsymbol{s}_i, \Theta). \tag{2.12}$$

For this choice, $\mathcal{Q}(\Theta^*, \hat{\Gamma})$ is constant with respect to $k$, and thus, equal to the likelihood of the model:

$$\begin{aligned}\mathcal{Q}(\Theta^*, \hat{\Gamma}) &= \sum_{i=1}^{N}\sum_{k=1}^{K} \hat{\gamma}_{ik} \log \frac{p\Big(\boldsymbol{s}_i, z_i = k|\Theta^*\Big)}{p(z_i = k|\boldsymbol{s}_i, \Theta^*)} \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} \hat{\gamma}_{ik} \log p(\boldsymbol{s}_i|\Theta^*) \\ &= \sum_{i=1}^{N} \log p(\boldsymbol{s}_i|\Theta^*) \\ &= \ell\left(\mathcal{S}; \Theta^*\right).\end{aligned} \tag{2.13}$$

In turn, if we fix $\Gamma^*$, a Lagrange optimization of $\Theta \mapsto \mathcal{Q}(\Theta, \Gamma^*)$ yields the following solutions:

$$\begin{aligned}\hat{\phi}_k &= \frac{\sum_{i=1}^{N} \gamma_{ik}}{\sum_{l=1}^{K}\sum_{j=1}^{N} \gamma_{jl}} = \frac{1}{N}\sum_{i=1}^{N} \gamma_{ik}, \\ \hat{\theta}_k(u, v) &= \frac{\sum_{i=1}^{N} \gamma_{ik}\eta_{uv}(\boldsymbol{s}_i)}{\sum_{l=1}^{K}\sum_{j=1}^{N} \gamma_{jl}\eta_{uv}(\boldsymbol{s}_j)},\end{aligned} \tag{2.14}$$

where $(u, v) \in \mathcal{M}^2$ and $\eta_{uv}(\boldsymbol{s}_i)$ is the number of transitions from $u$ to $v$ in $\boldsymbol{s}_i$.

### Implementation

The previous derivations sketch the EM algorithm: alternatively maximizing $\mathcal{Q}$ with respect to $\Gamma$ and $\Theta$. Algorithm 1 describes the different steps. Note that the E and M steps are also referred to as Expectation and Maximization steps, respectively.

**Algorithm 1** EM for a mixture model.

**Require:** $\mathcal{S} = \{\boldsymbol{s}_i\}_N$: input sequences
1: Initialize randomly $\Theta^{(0)}$ and $\Gamma^{(0)}$
2: **repeat**
3: **E-step**: Compute $\Gamma^{(t)} = \text{argmax}_\Gamma\ \mathcal{Q}(\Theta^{(t-1)}, \Gamma)$ (Equation 2.12)
4: **M-step**: Compute $\Theta^{(t)} = \text{argmax}_\Theta\ \mathcal{Q}(\Theta, \Gamma^{(t)})$ (Equations 2.14)
5: **until** convergence
Let $\hat{\Theta}$ be the set of parameters at convergence.
6: Compute $\hat{\Gamma} = \text{argmax}_\Gamma\ \mathcal{Q}(\hat{\Theta}, \Gamma)$ (Equation 2.12)
7: For all $i$, compute $z_i = \text{argmax}_k\ \hat{\gamma}_{ik}$
8: **return** $\mathcal{Z} = \{z_i\}_N$

Dempster et al. [80], later extended by Wu et al. [253], gave a proof that for some mixtures satisfying certain conditions, the sequence $\langle \Theta^{(t)}, \Gamma^{(t)} \rangle_{\mathbb{N}}$ produced by the algorithm surely becomes stationary, i.e., the algorithm converges. In the particular case of a mixture of Markov chain, the proof is simpler.

**Theorem 2.1** ([80]). *The log-likelihood increases after each iteration of the EM described in Algorithm 1, thus converges toward a local maximum.*

*Proof.* Each step of the algorithm solves a maximization problem, hence the value of $\mathcal{Q}$ increases after each E and M-step. In particular, after each E-step (Equation 2.13), $\mathcal{Q}(\Theta^{(t-1)}, \Gamma^{(t)}) = \ell\left(\mathcal{S}; \Theta^{(t-1)}\right)$. Hence, between two successive E-steps, the log-likelihood increases: $\ell\left(\mathcal{S}; \Theta^{(t-1)}\right) \leq \ell\left(\mathcal{S}; \Theta^{(t)}\right)$.

Since the parameter space $(\mathbb{S}^K \times (\mathbb{S}^M)^M)$ is a compact sub-space of a real vector space, the log-likelihood is upper-bounded. Given that $\ell$ is non-positive by definition and that its value increases after every two E-steps , the algorithm necessarily converges toward a local optimum. □

In practice, convergence is declared when the difference between two successive iteration is smaller than a threshold. However, there is no guarantee that the solution is the global maximum. The algorithm is thus usually run several times with different random initializations, to find an optimal solution.

### Clustering

There are two points of view on how to compute a clustering. A first approach is to use the components' likelihood, which groups together sequences that are most likely generated by the same model. Another stand is to use the posteriors, $p(z_i = k|\boldsymbol{s}_i) = \gamma_{ik}$. This second method is more common and it is the one used in Algorithm 1. Accordingly, $\gamma_{ik}$ are also called the *clusters' responsibilities* [54].

### Related Works

To avoid sub-optimal solutions, Broniatowski et al. [49] proposed the stochastic EM (SEM). Between the E and M steps, the assignments are drawn from the posteriors, such that the $\boldsymbol{\gamma}_i$ are one-hot vectors. These sampling might decrease the number of active clusters, which can rapidly lead to the trivial solution. To avoid such a situation, the algorithm re-samples the assignments until the number of clusters is stable, and only then the M-step starts. The SEM algorithm is also guaranteed to converge to a local maximum, although it might present erratic behaviors. As a remedy, Celeux and Diebolt [58] proposed to simulate an annealing [149] from SEM to EM. This way, the algorithm benefits in its early stage from the stochasticity of SEM to escape from sub-optimal regions. Later, the progressive shift toward EM accelerates the convergence.

In the case of a large data-set, the estimation of the posteriors may be expensive. A solution is to use mini-batches [17], or to update the model after each instance: an approach called incremental EM [202] (iEM). The latter is guaranteed to converge, but not necessarily to the same optimum as EM [114]. On the other hand, the one-at-a-time strategy makes it particularly suitable for online (on-the-fly) applications [170]. Offline, the order the data-set is browsed can also be modified, for example, to favor instances with high perplexity.

## 2.4 Bayesian Mixture Model

Taking a Bayesian perspective on a model consists of assuming that the parameters are random variables The inference scheme aims thus to find the distributions for which the parameters maximizing the likelihood of the data are the most likely. This is can be done by maximizing the posterior distribution, which gives the optimization strategy its name: *Maximum a posteriori* estimation (MAP).

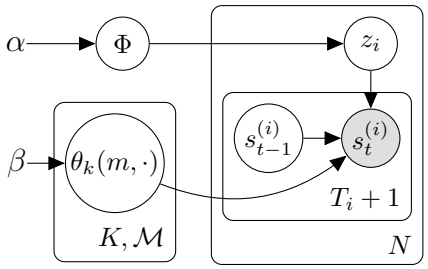



Figure 2.3: Graphical model of a Bayesian mixture of Markov chains.

To follow this line of thinking, we need to define the distributions governing the parameters, also called the *prior distributions*. A Bayesian mixture of Markov chains (BMMC) requires two priors whose characteristics are defined by some hyper-parameters $\alpha$ and $\beta$. The plate diagram of Figure 2.3 describes such a setting, that is summarized as follows:

$$\begin{aligned} \Phi &\sim \text{Prior}_{\Phi}(\alpha), \\ \theta_k &\sim \text{Prior}_{\theta}(\beta), \\ z &\sim \text{Cat}(\Phi), \\ s &\sim \text{Markov Chain}(\theta_z). \end{aligned} \tag{2.15}$$

The posterior probability of the mixture's parameters factorizes as follows:

$$p(\Phi, \boldsymbol{\Theta}, \mathcal{Z}|\mathcal{S}) \propto p(\mathcal{S}|\Phi, \boldsymbol{\theta}, \mathcal{Z}) p(\Phi, \boldsymbol{\theta}, \mathcal{Z}). \tag{2.16}$$

Note that, since $\mathcal{Z}$ is unknown, it is also considered a parameter. This formula is not computable yet, as the prior distributions are not explicitly provided by the model.

### 2.4.1 Conjugate Distribution

There is no restriction on the choice of prior distributions. However, for some choices, the optimization problem is easier to compute. Raiffa and Schlaifer [214] introduced the notion of a *conjugate* distribution.

**Definition 2.9** ([214])**.** *A prior distribution is* conjugate *to its posterior if*

1. *the resulting posterior is easy to calculate and to sample,*
2. *the expectations of some utility functions (e.g., expected value) have a closed-form,*
3. *the posterior and the prior belong to the same family of distributions.*

Remark that at this step a parallel can be draw with the construction of the $\mathcal{Q}$-function for MLE. Both inference schemes rely on an arbitrary choice to make the optimization tractable: the $\Gamma$ for MLE, the conjugate distribution for MAP.

#### Mixture Weights

Let us consider the case of the mixture weights as an example of how to choose a *proper* conjugate prior distribution. According to Bayes' rule and independence between variables, the posterior and prior of $\Phi$ are proportional:

$$p(\Phi|\mathcal{Z}) \propto p(\mathcal{Z}|\Phi)p(\Phi). \tag{2.17}$$

If we suppose that the assignments are known:

$$p(\mathcal{Z}|\Phi) = \prod_{i=1}^{N} \phi_{z_i} = \prod_{k=1}^{K} \phi_k^{N_k}, \tag{2.18}$$

where $N_k = \sum_{i=1}^{N} \delta_{z_i k}$ is the number of data-points assigned to component $k$. The right term looks like the pdf of a Dirichlet distribution (Equation 2.6) with parameter $\langle N_k + 1 \rangle_K$. The Dirichlet distribution seems, *therefore*, to be a good candidate for the prior. As a matter of fact, if the prior distribution of $\Phi$ is a symmetric Dirichlet distribution with hyper-parameter $\alpha > 0$, the posterior becomes proportional to a Dirichlet distribution with parameters $\langle N_k + \alpha \rangle_K$:

$$p(\Phi|\mathcal{Z}) \propto p(\mathcal{Z}|\Phi)p(\Phi) = \prod_{k=1}^{K} \phi_k^{N_k} \prod_{k=1}^{K} \phi_k^{\alpha-1} = \prod_{k=1}^{K} \phi_k^{N_k+\alpha-1} \tag{2.19}$$

To summarize: If the prior follows a symmetric Dirichlet distribution, the posterior is easy to calculate; the posterior and the prior are both from the same family of distributions; the expected value of the posterior has a simple closed form:

$$\mathbb{E}[\phi_k|\mathcal{Z}] \propto \frac{N_k + \alpha}{N + K\alpha}. \tag{2.20}$$

Consequently, the posterior and the prior are *conjugate* in the sense of Definition 2.9 Note that the Dirichlet prior does not need to be symmetric, however, without this assumption the formulas are more complex.

#### Transition Probabilities

The transitions from $u \in \mathcal{M}$ follows a categorical distribution. Hence, the same reasoning as for $\Phi$ applies, such that:

$$\forall k \in [1 \,..\, K],\ \forall v \in \mathcal{M},\ \mathbb{E}[\theta_k(u,v)|\mathcal{Z}] = \frac{\sum_{i=1}^{N} \delta_{z_i k} \eta_{uv}(\boldsymbol{s}_i) + \beta}{\sum_{w \in \mathcal{M}} \sum_{i=1}^{N} \delta_{z_i k} \eta_{uw}(\boldsymbol{s}_i) + M\beta}, \tag{2.21}$$

where $\eta_{uv}(\boldsymbol{s}_i)$ is the number of transitions from $u$ to $v$ in $\boldsymbol{s}_i$, and $\beta$ a hyper-parameter.

### 2.4.2 Gibbs Sampler

Recall that MAP aims to learn the prior distributions for which the expected values of the parameters maximize the likelihood. Often, the computations of the expected value are intractable. However, if all but one parameter are known, one can sample the missing parameter multiple times from its posterior and obtain a good enough approximation of its expected value. This heuristic is at the core of Markov Chain Monte Carlo methods [188], such as Metropolis-Hastings algorithm [122] and the Gibbs sampler [104, 57]. While the former rejects samples that do not pass some tests, the latter accepts them all. In this text, we focus on the latter.

#### Implementation

The idea of the Gibbs sampler is to draw each parameter alternatively, given all the others. After a random initialization, the operation is repeated until the sample means are statistically significant. The repetition creates a Markov chain as described in Algorithm 2. The sampling of $\phi_k^{(t)}$ depends on $\phi_{k+1}^{(t-1)}$, that itself depends on $\phi_k^{(t-1)}$. Per transitivity, the sequence $\langle \phi_k^{(t)} \rangle_{\mathbb{N}}$ satisfies the Markov property. The algorithm is proven to let the empirical distributions of each parameter converge toward their respective true posteriors [57] since these are stationary points of their respective Markov chain.

**Algorithm 2** Gibbs sampler for a mixture model.

**Require:** $\mathcal{S} = \{\boldsymbol{s}_i\}_N$: input data, $T$: number of iterations, $B$: burn-in period

Initialize randomly $\Theta^{(0)} = (\Phi^{(0)}, \boldsymbol{\theta}^{(0)}, \mathcal{Z}^{(0)})$
**for** $t = 1 \dots T$ **do**
  **for** $k = 1 \dots K$ **do**
    Sample $\phi_k^{(t)}$ from $\phi_k | \Phi_{:k-1}^{(t)}, \Phi_{k+1:}^{(t-1)}, \boldsymbol{\theta}^{(t-1)}, \mathcal{Z}^{(t-1)}, \mathcal{S}$
  **end for**
  **for** $k = 1 \dots K$ **do**
    Sample $\theta_k^{(t)}$ from $\theta_k | \Phi^{(t)}, \boldsymbol{\theta}_{:k}^{(t)}, \boldsymbol{\theta}_{k+1:}^{(t-1)}, \mathcal{Z}^{(t-1)}, \mathcal{S}$
  **end for**
  **for** $i = 1 \dots N$ **do**
    Sample $z_i^{(t)}$ from $z_i | \Phi^{(t)}, \boldsymbol{\theta}^{(t)}, \mathcal{Z}_{:i-1}^{(t)}, \mathcal{Z}_{i+1:}^{(t-1)}, \mathcal{S}$
  **end for**
**end for**
**return** $\hat{\Theta} = \frac{1}{T-B} \sum_{t=B}^{T} \Theta^{(t)}$

In contrast to EM, the estimates at convergence are less dependent on the initialization. On the other hand, the first ones maintain a strong dependence with the initial states. Therefore, the final estimation does not include them (*burn-in* period).

### Auto-correlation

The Markov chain described by $\langle \Theta^{(t)} \rangle_{\mathbb{N}}$ in Algorithm 2 induces an auto-correlation between samples of successive iterations. Consequently, the algorithm may not exhaustively explore the parameters' space, leading to a biased estimation of the posterior. The *blocked Gibbs sampler* reduces this influence by sampling several variables at the same time, e.g., sample $\Phi$ as a whole from a Dirichlet distribution instead of each $\phi_k$ from univariate Beta distributions. The *collapsed Gibbs sampler* integrates out some variables, if the resulting formula has a closed-form. Let us consider the reassignment of $z_i$, noted $\hat{z}_i$. Using the independence between variables (Figure 2.3), the update of Algorithm 2 is as follows:

$$\begin{aligned} p(\hat{z}_i = k | \Phi, \boldsymbol{\theta}, \mathcal{Z}_{-i}, \mathcal{S}) &= p(\hat{z}_i = k | \Phi, \mathcal{Z}_{-i}, \mathcal{S}) \\ &\propto p(\hat{z}_i = k, \boldsymbol{s}_i | \Phi, \mathcal{Z}_{-i}) \\ &= p(\hat{z}_i = k | \Phi, \mathcal{Z}_{-i})\, p(\boldsymbol{s}_i | \mathcal{Z}_{-i}, \hat{z}_i = k) \end{aligned} \tag{2.22}$$

where $\mathcal{Z}_{-i} = \mathcal{Z} \setminus \{z_i\}$ and the superscript $\cdot^{(t)}$ are omitted. To sample the new $\hat{z}_i$ independently from $\Phi$, a collapsed Gibbs sampler integrates out $\Phi$ from $p(\hat{z}_i = k | \Phi, \mathcal{Z}_{-i})$, i.e., it is summed over all the possible values of $\Phi$. Since the prior of $\Phi$ is

a Dirichlet distribution, we obtain:

$$\begin{aligned} p(\hat{z}_i = k|\mathcal{Z}_{-i}) &= \int_\Phi p(\hat{z}_i = k, \Phi|\mathcal{Z}_{-i})\partial\Phi \\ &= \int_\Phi p(\hat{z}_i = k|\Phi, \mathcal{Z}_{-i})\, p(\Phi|\mathcal{Z}_{-i})\partial\Phi \\ &= \mathbb{E}[\hat{\phi}_k|\mathcal{Z}_{-i}] = \frac{N_{-i,k} + \alpha}{N + K\alpha}, \end{aligned} \tag{2.23}$$

with $N_{-i,k} = N_k - \delta_{z_i,k}$. The last line derives from Equations 2.19 and 2.20. The assignments can now be sampled from a simple formula, which highlights the benefits of the collapsed Gibbs sampler.

**Relation with EM**

A Gibbs sampler and an incremental stochastic EM (iEM [202] + sEM [49]) are similar. Both can be split into three steps repeated until convergence: for each datapoint (i) compute the assignment's posterior probabilities, (ii) draw a new assignment, (iii) maximize the other parameters. The difference is that the estimations of the posteriors in iEM do not exclude the current point.

## 2.5 Dirichlet Processes and Mixture Models

We have seen how to infer the parameters of a mixture of models, except for the number of models, $K$. The standard approach remains the grid search which implies learning the model for several values of $K$. The best value is then chosen using the *elbow rule* [218] of the curve of the likelihood at convergence, or some criteria from information Theory such as the Akaike information criterion (AIC) [6, 53] or the Bayesian information criterion (BIC) [226]. The approach we present here let the optimization *learn* the number of components.

The rationale is to let $K$ vary more or less randomly. However, this induces some theoretical challenges: when the number of mixtures changes, the dimension of $\Phi$ changes, and thus the order of its Dirichlet prior distribution. A first solution is to consider that there is an infinite number of components. At first sight, this may contradict the definition of a Dirichlet distribution which relies on a simplex of finite dimension, but the aggregation property provides a workaround. Nevertheless, this strategy reaches its limit when it comes to updating the values of the prior. A more robust approach requires a change of paradigm. The idea is to consider vector $\Phi^*$ as a realization of a Dirichlet process (DP) *somehow* indexed by the number of non-empty

components plus one. That is, $\Phi^*$ is a random variable that follows a Dirichlet distribution of order $K+1$ with parameters depending on the index. The extra dimension serves to model the creation of a new component in the mixture, i.e., an increase of the prior's order. Formally, any update of the Dirichlet distribution's parameters or order corresponds to a change of index of the stochastic process.

In the following, we formalize this heuristic and consider the case of a symmetric Dirichlet process. In practice DPs are cumbersome to implement since the size of some tables may vary. We discuss thus, in the last paragraph, a finite approximation.

### 2.5.1 Dirichlet Processes

Let us first give the original definition of Ferguson [98].

**Definition 2.10** ([98])**.** *Let $(\mathcal{U}, \mathcal{A})$ be a measurable set with a finite measure $\alpha$, a* Dirichlet process *(DP) with parameter $\alpha$ is indexed by the set of the measurable partitions of $\mathcal{U}$. For every partition $\langle E_k \rangle_K$ with $K > 1$ and $E_k \in \mathcal{A}$, $\mathrm{DP}_\alpha(\langle E_k \rangle_K)$ is a random variable on the set of the probability mass functions on $(\Omega, \mathcal{A})$, $P$, such that $\langle P(E_k) \rangle_K$ is the Dirichlet distribution $\mathrm{Dir}(\langle \alpha(E_k) \rangle_K)$.*

The random vector $\langle P(E_k) \rangle_K$ can as well be seen as a vector of $\mathbb{S}^K$, which would be in line with Definition 2.6 of the Dirichlet distribution. Therefore, we propose an alternative definition using simplices instead of probability mass functions. Moreover, the finite distribution is replaced by a scaled probability.

**Definition 2.11.** *Let $(\mathcal{U}, \mathcal{A})$ be a measurable set with a probability $H$ and $\alpha > 0$, a* Dirichlet process *(DP) with parameter $\alpha H$ is indexed by the set of the measurable partitions of $\mathcal{U}$.*

*For a finite, measurable partition $\langle E_k \rangle_K$ of size $K > 1$,*

$$z \sim \mathrm{DP}_{\alpha H}(\langle E_k \rangle_K) \Leftrightarrow z \in \mathbb{S}^K \text{ and } z \sim \mathrm{Dir}(\langle \alpha H(E_k) \rangle_K). \tag{2.24}$$

*For a countably infinite, measurable partition, $\langle E_k \rangle_{\mathbb{N}}$,*

$$\begin{gathered} z \sim \mathrm{DP}_{\alpha H}(\langle E_k \rangle_{\mathbb{N}}) \Leftrightarrow z \in \mathbb{S}^\infty \text{ and} \\ \forall k \geq 1, (z_1, \ldots, z_k, \textstyle\sum_{l=k+1}^{\infty} z_l) \sim \mathrm{Dir}\left(\alpha H(E_1), \ldots, \alpha H(E_k), \alpha - \alpha H\left(\cup_{l=1}^{k} E_l\right)\right) \end{gathered} \tag{2.25}$$

This last equation is a consequence of the aggregation property (Lemma 2.3) and of the $\sigma$-additivity of the measure $H$.

Figure 2.4 depicts a plate diagram of the Dirichlet process based mixture of Markov chains (DP-MMC), which can be summarized as follows:

$$\begin{aligned} \Phi^* &\sim \mathrm{DP}_{\alpha H}(E) \\ z &\sim \mathrm{Cat}(\Phi^*) \\ s &\sim \text{Markov Chain}(\theta_z). \end{aligned} \tag{2.26}$$

where $E$ is a partition of the index set of the DP.

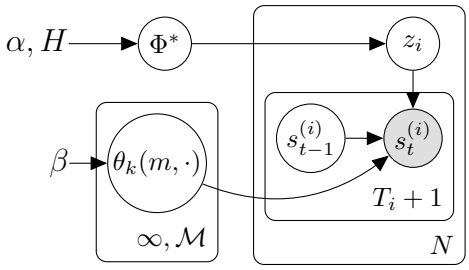



Figure 2.4: Graphical model of a Dirichlet process based mixture of Markov chains.

### 2.5.2 Symmetric DP

In order to have a better insight into DPs, we exhibit a family of partitions of $\mathbb{R}_{\geq 0}$ that induces symmetric Dirichlet distributions of any order $K > 1$.

**Lemma 2.5.** *There exists a measurable set $(\mathcal{U}, \mathcal{A})$, a probability $H$, and a countable collection of measurable partitions of $\mathcal{U}$, $\langle E_K \rangle_{K \in \mathbb{N}}$, with $E_K = \langle e_l^{(K)} \rangle_{1 \leq l \leq K+1}$, such that for any $k \in \mathbb{N}$ and $l \in [1 \,..\, K+1]$, $H\left(e_l^{(K)}\right) = \frac{1}{K+1}$.*

*Proof.* A solution is realized for $\mathcal{U} = \mathbb{R}_{\geq 0}$, $\mathcal{A}$ its Borel set, $H = \mathrm{Exp}(1)$[1], and the intervals defined for any $K \in \mathbb{N}$ and $l \in [1 \,..\, K+1]$ as follows:

$$e_l^{(K)} = \left[ -\log\left(\frac{K+2-l}{K+1}\right), -\log\left(\frac{K+1-l}{K+1}\right)\right). \tag{2.27}$$

□

When the DP is indexed by $E_K$, the model is equiavlent to a Bayesian mixture model with $K+1$ components (Equation 2.15). The extra dimension of $\Phi^*$ models the opportunity to increase the order. If $z_i$ is reassigned to an already opened component, the model's parameters are updated accordingly to the optimization scheme of a

[1] pdf of the exponential distribution with parameter 1.

BMMC. If the draw returns $z_i = K+1$, the index of $\mathrm{DP}_{\alpha H}$ is set to $E_{K+1}$ and the mixture has now $K+2$ components. Furthermore, $\Phi^*$ gains one more dimension, $z_i$ is set equal to $K+1$, and the model's parameters are updated as in a BMMC with $K+2$ components. According to Equations 2.23, the expected posterior proabilities of $\Phi$ are:

$$\begin{aligned} \text{for } 1 \leq k \leq K, \quad \mathbb{E}[\phi_k^* | \mathcal{Z}] &= \frac{N_k + \frac{\alpha}{K+1}}{N+\alpha}, \\ \mathbb{E}[\phi_{K+1}^* | \mathcal{Z}] &= \frac{\frac{\alpha}{K+1}}{N+\alpha}. \end{aligned} \tag{2.28}$$

**Simplification**

The symmetry allows for a major simplification: a fixed family of partitions. According to the theory, two DPs with $K$ components may be indexed by different partitions with different $H$-measure. However, since the symmetry constraint ensures a constant measure for a specific $K$, these partitions are all equivalent.

**Degenerative Symmetrization**

Beal et al. [23] introduce symmetric DPs without including the probability of opening a new component inside $\Phi$. Using the aggregation property, they split the assignment phase in two steps. First, it is decided whether a new cluster is open with a fixed probability proportional to $\alpha$. If not, the point is assigned to an already opened cluster using $\Phi$. To remove the dependency of this last draw on the size of the mixture, they let $K$ tend to infinity. We claim that this presentation is degenerate. First, if $K$ is infinite, the the first step is irrelevant. Second, at the limit, the prior of $\Phi$ is a Dirichlet distribution with all weights null, which is not well defined. Nevertheless, we admit that once the components are populated, that second issue disappears.

An equivalent but non-degenerate construction relies on $\Phi^*$, $H = \mathrm{Exp}(1)$ and a DP indexed with the following collection of measurable partitions of $\mathbb{R}$, $\langle F_K \rangle_{K \in \mathbb{N}}$:

$$\forall K \geq 1, \quad F_K = \{u_l^{(K)}\}_{1 \leq l \leq K+1},$$

$$\forall l \in [1 \,..\, K], \quad u_l^{(K)} = [a_l, b_l^{(K)}] \quad \text{and} \quad u_{K+1}^{(K)} = \mathbb{R} \setminus \{u_l^{(K)}\}_{1 \leq l \leq K}.$$

$$a_1 = 0, \quad a_l = 1 - \log(e - \sum_{j=1}^{l-1} e^{-j}) \quad \text{and} \quad b_l^{(K)} = 1 - \log(e - e^{-l-K}).$$

For a given $K \geq 1$, $H(u_K^{(l)}) = e^{-K+1}$ for any $l \in [1 \,..\, K]$ and $H(u_{K+1}^{(K)}) = 1 - Ke^{-K+1}$. As $K$ tends toward infinity, $b_l^{(K)}$ becomes closer to $a_l$, and the intervals $u_l^{(K)}$ converge toward the singletons and $H(u_K^{(l)}) \to 0$. On the other hand, the complementary set

expands until its measure becomes 1. At the limit, we obtain the following infinite partition:

$$\begin{aligned} F_\infty &= \{a_l\}_{\mathbb{N}} \cup (\mathbb{R} \setminus \mathbb{N}) = \{u_l\}_{\mathbb{N}} \cup u_\infty, \\ \forall l \in \mathbb{N},\, \alpha H(u_l) &= 0, \quad \text{and} \quad \alpha H(u_\infty) = \alpha. \end{aligned} \tag{2.29}$$

Here again, if a reassignment opens a new cluster, the index of the DP may change for another partition, but equivalent to $F_\infty$. Hence, we can assume that the indexing partition remains the same. Using the aggregation property of the Dirichlet distribution and Equation 2.25, the prior of $\Phi^*$ is well defined:

$$\forall K \geq 1,\ \Phi^* \sim \text{Dir}(\underbrace{0, \ldots, 0}_{K}, \alpha). \tag{2.30}$$

In the construction of Beal et al. [23], the $\Phi$ does not have the extra dimension, hence its prior Dirichlet distribution is $\text{Dir}(\mathbf{0}_K)$, which is not well defined.

### 2.5.3 Approximation

If an infinite number of clusters is theoretically attractive, in practice it is problematic. The value of $K$ changing, the size of the table storing the components' parameters can not be fixed in advance, which hinders computationally efficiency of any implementation. Regarding the interpretability, an infinite or even a potentially large number of clusters prohibit any analysis of the groupings. Lastly, in an offline scenario, the number of clusters is anyway upper-bounded by the number of instances in the data-set.

In a landmark paper Ishwaran and Zarepour [133] show that for $K$ big enough, Bayesian mixture models are good approximations of DP mixture models. Their approximation is called a *degree k-weak limit approximation.*

**Theorem 2.2.** *[133] Let $\mathcal{Z}^{(K)}$ and $\mathcal{Z}^{\infty}$ be two sets of assignments obtained from a Bayesian mixture model with $K$ components and a DP mixture model, respectively. Let $D_K$ and $D_\infty$ equal the number of distinct values in $\mathcal{Z}^{(K)}$ and $\mathcal{Z}^{\infty}$, respectively. If $H$ is nonatomic, then*

$$\frac{K!}{K^l(K-l)!} \leq \frac{p(D_K = l)}{p(D_\infty = l)} \leq N^{\frac{\alpha l}{K}}, \quad \textit{for } l = 1, \ldots, \min(N, K).$$

*Both bounds converge to 1 as $K \to \infty$.*

Formally, the previous theorem guarantees that the clusterings obtained from a Bayesian mixture model tend (up to a permutation) to the distribution of indices of clusterings obtain with a DP mixture model, when the number of components increases. This is equivalent to say that the groupings are equivalent at infinity.

# Chapter 3

# Archetypal Analysis and Content Analysis

Several studies have shown that the time spent reading a page or completing exercises is an indicator of the level of pupils's motivation [126, 143]. In this chapter, we follow this line of research but from a content perspective. We choose to model behaviors using visibility times of different types of content and look for correlations with pupils' motivation. Naively representing a session using only the shares of each object in the total duration is sub-optimal. As we shall see, in this case, some correlations cancel each other out. Instead, we reconstruct the time distributions from factors close to the data, and in greater number than the number of dimensions. This way, we obtain a better understanding of the relationships between the objects themselves as well as with motivation.

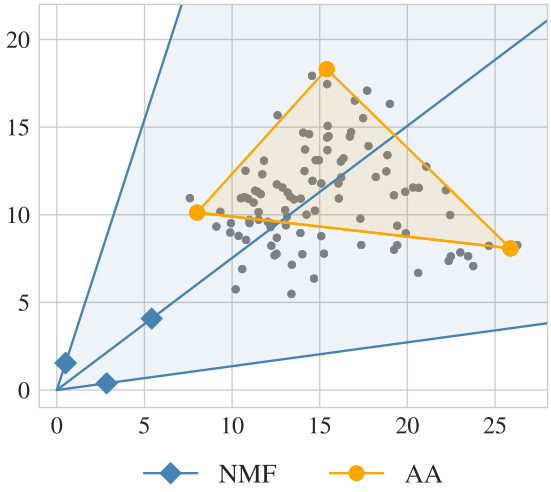



Figure 3.1: The factors of NMF (blue squares) are far from the data. Besides, only two a necessary to define the enveloping cone. On the other hand, Archetypal Analysis makes the est out of three factors (orange circles) and form a polyhedron that follows the distribution of the data.

Centroid-based methods provide factors close to the data, but cluster weights are designed to indicate the closest centroid, not to reconstruct points from them. Matrix factorization techniques lend themselves better to this problem. Two major methods are singular value decomposition (SVD) [233], and non-negative matrix factorization (NMF) [206, 163]. The former uses the eigenvectors of the data-matrix as factors,

while the second looks for an enveloping cone (blue in Figure 3.1). These methods give good reconstruction losses, but their factors lack interpretability. First, the factors may be far from the data. Stochastic NMF for clustering [11, 12] or with convex constraints [85] improve this aspect, though. Moreover, the number of relevant factors is limited to the number of dimensions (see Figure 3.1).

Cutler and Breiman [71] studied the problem of calculating interpretable factors. Their idea is to define an enveloping polyhedron using points, or *archetypes*, on the convex hull of the data-set. This way, any point within the polyhedron can be represented without loss as a convex combination of the archetypes. To ensure that the factors are on the boundary, the archetypal analysis (AA) calculates them as convex combinations of the data-points. The reconstruction loss then pushes them towards the boundary. Several variants have been recently proposed [197, 65, 228].

Because it relies on two stochastic matrices, AA can be very heavy to calculate. Since the archetypes lie on the convex hull, a simple idea is to reduce their search to the convex combinations of the *frame*, i.e., the vertices of the convex hull. We propose to go one step further and approach AA by only reconstructing the frame. Indeed, any data-points can be expressed losslessly as a convex combination of points of the frame. So a good approximation of the latter would contain enough points to yield a reduced reconstruction loss. The idea is similar to that of Thurau et al. [239], who propose to approximate the frame using two-dimensional projections. However, they add an unnecessary approximation layer to the problem.

At first glance, the calculation of the exact frame is the solution to the described problem. However, it is a very demanding task. Standard approaches like Quickhull [18] are infeasible in high dimensions because of dispensable triangulations. Discarding the triangulation leads to linear programming (LP)-based solutions [88, 204, 89] that test whether a point at-hand is included in the convex-hull: much ado for only a single point. In addition, duplicates in the data cause false negatives to duplicated extreme points. In the following, we (i) show that the exact frame can be computed by a quadratic program (QP), (ii) reduce the optimization to an existing algorithm, and (iii) provide theoretical and empirical justifications for the developed method.

The remainder is structured as follows. Section 3.1 reviews archetypal analysis and Section 3.2 contains the main contributions: a new computational method for finding the frame and a frame-based matrix factorization. Section 3.3 reports on our empirical results. In Section 3.3.3, we use our method to analyze and interpret behaviors from the mBook.

## 3.1 Archetypal Analysis

Let $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ be a set of $N$ points of $\mathbb{R}^M$ summarized by matrix $\boldsymbol{X} \in \mathbb{R}^{N \times M}$, the goal of archetypal analysis (AA) [71] is to find a factorization of the data with $K \in \mathbb{N}$ factors, such that

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{B}\boldsymbol{X} = \boldsymbol{A}\boldsymbol{Z}, \tag{3.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{N \times K}$ and $\boldsymbol{B} \in \mathbb{R}^{K \times N}$ are row-stochastic, and the column-vectors of $\boldsymbol{Z} \in \mathbb{R}^{K \times M}$ correspond the factors or *archetypes*, $\langle \boldsymbol{z}_k \rangle_K$. The formula says that the data-points are convex combinations of the archetypes, while these are themselves convex combinations of the data-points:

$$\forall i \in [1 \,..\, N], \quad \boldsymbol{a}_{i\cdot} \in \mathbb{S}^K, \quad \boldsymbol{x}_i = \sum_{k=1}^{K} a_{ik} \boldsymbol{z}_k,$$
$$\forall k \in [1 \,..\, K], \quad \boldsymbol{b}_{k\cdot} \in \mathbb{S}^N, \quad \boldsymbol{z}_k = \sum_{i=1}^{N} b_{ki} \boldsymbol{x}_i.$$

The factorization can be obtained by minimizing the residual sum of squares (RSS)

$$\min \ \mathrm{RSS}(k) = \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{Z}\|_F^2 = \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}\boldsymbol{X}\|_F^2.$$

The optimization problem is non-convex in $\boldsymbol{A}$ and $\boldsymbol{B}$, but convex if one matrix is fixed. It can be solved by alternatively computing $\boldsymbol{A}$ and $\boldsymbol{B}$ as outlined in Algorithm 3. Cutler and Breiman [71] proved that the archetypes lie on the boundary of the convex-hull of the data $\mathcal{X}$ for $1 < K < N$. From a geometrical point of view, AA yields an approximation of the convex-hull with $K$ vertices. Points inside the polyhedron formed by the archetypes are reconstructed in a lossless way, while those outside are approximated by their orthogonal projection onto this polyhedron. Thus, minimizing the RSS also minimizes the quantity of these projections.

### 3.1.1 Convex Hull and Frame

The convex-hull of a set of $\mathbb{R}^M$ can be defined in various ways [29]. We give here, two definitions.

**Definition 3.1.** *Let $\mathcal{X}$ be a set of $\mathbb{R}^M$, its* convex-hull $\mathrm{Conv}(\mathcal{X})$ *is:*

1. *The intersection of all convex sets containing $\mathcal{X}$.*
2. *The set of all the convex combinations of points in $\mathcal{X}$.*

**Algorithm 3** Archetypal Analysis (AA)

**Input:** data matrix $\boldsymbol{X}$, number of archetypes $K$
**Output:** factor matrices $\boldsymbol{A}, \boldsymbol{Z}$
$\boldsymbol{Z}$ = initial guess of archetypes on $\boldsymbol{X}$
**while** not converged **do**
  **for** $i = 1, 2, \ldots, N$ **do**
    $\boldsymbol{a}_{i\cdot} = \underset{\boldsymbol{a} \in \mathbb{S}^K}{\operatorname{argmin}} \|\boldsymbol{Z}^\top a - \boldsymbol{x}_i\|_2^2$
  **end for**
  $\boldsymbol{Z} = (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top \mathbf{X}$
  **for** $l = 1, 2, \ldots, K$ **do**
    $\boldsymbol{b}_{l\cdot} = \underset{\boldsymbol{b} \in \mathbb{S}^N}{\operatorname{argmin}} \|\boldsymbol{X}^\top b - \boldsymbol{z}_l\|_2^2$
  **end for**
  $\boldsymbol{Z} = \mathbf{B}\mathbf{X}$
**end while**

In the case of a discrete set, we distinguish the set of the vertices of the convex-hull.

**Definition 3.2.** *The* frame *of* $\mathcal{X}$, $\mathrm{Frame}(\mathcal{X})$*, is the set of the points of the boundary of* $\mathcal{X}$*,* $\partial\mathcal{X}$*. The proportion of points in* $\mathcal{X}$ *belonging to the frame is called the* frame density*.*

The frame consists thus of the extreme points of $\mathcal{X}$. Those points cannot be represented as convex combinations of other points rather than themselves. Note that a set and its frame share the same convex-hull, i.e. $\mathrm{Conv}(\mathrm{Frame}(\mathcal{X})) = \mathrm{Conv}(\mathcal{X})$.

We state now two straightforward properties that will become handy in the remainder. First, given a point, the maximizer of the inner-product relatively to that point is an extreme point, i.e., it belongs to the frame of the set.

**Lemma 3.1.** *Let* $\mathcal{X}$ *be a finite discrete set of* $\mathbb{R}^M$*, then*

$$\forall \boldsymbol{x} \in \mathcal{X}, \quad \underset{\boldsymbol{x}' \in \mathcal{X}}{\operatorname{argmax}} \left( \boldsymbol{x}^\top \boldsymbol{x}' \right) \in \mathrm{Frame}(\mathcal{X})$$

*Proof.* Continuity, and convexity of the inner-product imply that its maximum on a compact set is realized by an extreme point of the domain. Since the domain $\mathcal{X}$ is finite, it is compact. Therefore, the maximum belongs to the frame of $\mathcal{X}$. □

Furthermore, every point of the domain lies in the convex span of some points on the frame.

**Proposition 3.1.** *Every point* $\boldsymbol{x}$ *of a finite discrete set* $\mathcal{X} \subset \mathbb{R}^M$ *can be written as a convex combination of at most* $M + 1$ *points of* $\mathrm{Frame}(\mathcal{X})$*.*

*Proof.* See Brondsted [48]. □

In the remainder, we assume $N > M$ and note $\mathcal{F}$ the frame of $\mathcal{X}$. It is of cardinal $P$ and summarized by matrix $\boldsymbol{F} \in \mathbb{R}^{P \times M}$.

## 3.2 Frame-AA

The optimization pushes archetypes toward the convex-hull of the data [71]. We use this property to approximate and speed up AA.

### 3.2.1 Motivation

Given that the archetypes lie on the convex-hull, it is possible to restrict the search of the archetypes to the frame $\mathcal{F}$. We intend to go one step further. Since the frame $\mathcal{F}$ and the data $\mathcal{X}$ yield the same convex-hull, the better the archetypes approximate the frame, the lower is the loss. The idea is depicted in Figure 3.2. Although archetypal analysis is only computed on the frame, as seen in the right subplot, it yields almost identical archetypes as AA computed on the whole data-set (left). Moreover, the reduced number of points yields a faster convergence.

The construction develops as follows. First, we factorize the frame using AA:

$$\boldsymbol{F} = \boldsymbol{A}_F \boldsymbol{B}_F \boldsymbol{F} = \boldsymbol{A}_F \boldsymbol{Z}, \tag{3.2}$$

where $\boldsymbol{A}_F \in \mathbb{R}^{P \times K}$ and $\boldsymbol{B}_F \in \mathbb{R}^{K \times P}$ are both row-stochastic. Then, the archetypes $\boldsymbol{Z}$ are used to compute the weight matrix $\boldsymbol{A} \in \mathbb{R}^{N \times K}$ for all the data-points. This procedure is called Frame-AA and is presented in Algorithm 4. Note that the idea does

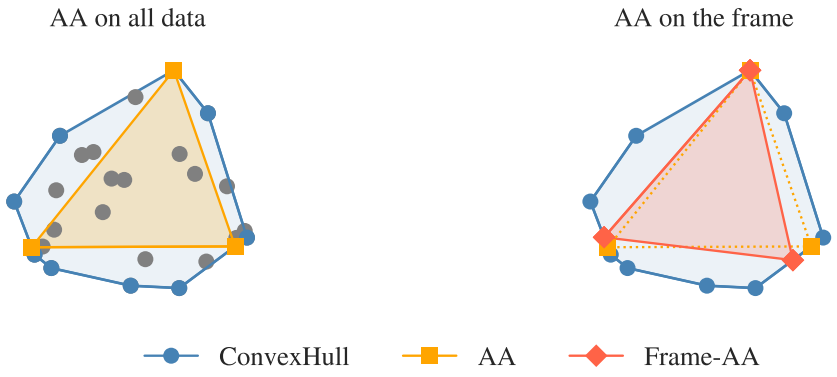



Figure 3.2: Computing the archetypes from the whole data-set (left) or from the frame (right) yields very similar solutions.

**Algorithm 4** Frame-AA

**Input:** Data-matrix $\boldsymbol{X}$, number of archetypes $K \in \mathbb{N}$
**Output:** Factor matrices $\boldsymbol{A}, \boldsymbol{Z}$
$\boldsymbol{F} = \text{Frame}(\boldsymbol{X})$
$\boldsymbol{A}_F, \boldsymbol{Z} = \text{ArchetypalAnalysis}(\boldsymbol{F}, K)$
$\boldsymbol{A} = \boldsymbol{0}_{N \times K}$
**for** $i = 1, 2, \ldots, N$ **do**
  $\boldsymbol{a}_{i\cdot} = \underset{\boldsymbol{a} \in \mathbb{S}^K}{\text{argmin}} \; \|\boldsymbol{Z}^\top a - \boldsymbol{x}_i\|_2^2$
**end for**

not only apply to standard archetypal analysis as presented in Cutler and Breiman [71] but also to all variants thereof [197, 65, 19].

Assuming a low frame density, i.e., $P \ll N$, the polyhedron of a sufficient approximation of the frame, and therefore of the convex-hull, includes most of the interior points. On the other hand, the problem tends toward a standard AA and the speed up vanishes as the frame density increases, i.e., $P \to N$. However, based on the nature of the problem, we claim that AA makes no sense in high frame density scenarios. In such a case, almost all points are projected, unless $K$ is close to $N$, which is usually not the case.

In Algorithm 4, we assume that the frame $\mathcal{F}$ or equivalently the frame matrix $\boldsymbol{F}$ is already given. In the following section, we present a novel algorithm for efficiently computing the frame of a discrete data-set. Before that, we show now that AA can be solved efficiently using a standard non-negative least squares algorithm.

### 3.2.2 Representation

Archetypal analysis aims to represent any point of $\mathcal{X}$ as a convex combination of points from the frame. Formally, for any point $\boldsymbol{y} \in \mathcal{X}$, AA looks for $\boldsymbol{a} \in \mathbb{R}^N$ such that:

$$\begin{aligned} \boldsymbol{X}^\top \boldsymbol{a} &= \boldsymbol{y} \\ \boldsymbol{a} \in \mathbb{S}^N = 1 \;\wedge\; a_i \neq 0 &\Rightarrow \boldsymbol{x}_i \in \mathcal{F}. \end{aligned} \tag{3.3}$$

This problem can be reduced to a least-squares problem.

**Theorem 3.1.** *Let $\hat{\boldsymbol{X}} \in \mathbb{R}^{N \times (M+1)}$ and $\hat{\boldsymbol{y}} \in \mathbb{R}^{M+1}$ be the augmentation of $\boldsymbol{X}$ and $\boldsymbol{y}$:*

$$\forall i, j \in [1 \,..\, N] \times [1 \,..\, M], \quad \hat{x}_{ij} = x_{ij},\ \hat{x}_{i(M+1)} = 1, \quad \hat{y}_j = y_j,\ \hat{y}_{M+1} = 1.$$

*For $\boldsymbol{y} \in \mathcal{X}$, the following problems are equivalent.*

*(i)* $\boldsymbol{X}^\top \boldsymbol{a} = \boldsymbol{y} \;:\; \boldsymbol{a} \in \mathbb{S}^N \wedge\; a_i \neq 0 \Rightarrow \boldsymbol{x}_i \in \mathcal{F}.$

*(ii)* $\hat{\boldsymbol{X}}^\top \boldsymbol{a} = \hat{\boldsymbol{y}} \ : \ \boldsymbol{a} \geq 0 \wedge \ a_i \neq 0 \Rightarrow \boldsymbol{x}_i \in \mathcal{F}.$

*(iii)* $\operatorname{argmin}_{\boldsymbol{a}\geq 0} \ \frac{1}{2}\|\hat{\boldsymbol{X}}^\top \boldsymbol{a} - \hat{\boldsymbol{y}}\|_2^2 \ : \ a_i \neq 0 \Rightarrow \boldsymbol{x}_i \in \mathcal{F}.$

*Proof.* (i) is equivalent to (ii) as it integrates the stochasticicty constraint into the system of linear equations. Proposition 3.1 assures that (i) has a solution. Hence, $\min \frac{1}{2}\|\boldsymbol{X}^\top \boldsymbol{a} - \hat{\boldsymbol{y}}\|_2^2 = 0$. Meaning that a solution of (iii) is also a solution to (ii) and hence (i). □

Without the frame condition on $\boldsymbol{a}$, problem (iii) of Theorem 3.1 is equivalent to the non-negative least squares (NNLS) problem which is a special case of a quadratic problem (QP). The active-set method from Lawson and Hanson [161] is proven to yield a least-squares estimate of this unconstrained problem (Algorithm 5). We prove that it also provides a solution to the problems of Theorem 3.1.

**Theorem 3.2.** *The active-set method from Lawson and Hanson [161] solves the problems of Theorem 3.1.*

*Proof.* Let $\boldsymbol{a}$ be the solution returned for the problem:

$$\underset{\boldsymbol{a}\geq 0}{\operatorname{argmin}} \ \frac{1}{2}\|\hat{\boldsymbol{X}}^\top \boldsymbol{a} - \hat{\boldsymbol{y}}\|_2^2.$$

Accordingly to Algorithm 5, if $a_j \neq 0$, at some iteration $t$, $w_j^{(t)}$ was the greatest coefficient of $\boldsymbol{w}^{(t)}$, i.e.,

$$j = \underset{\{i \ | \ a_i^{(t)}=0\}}{\operatorname{argmax}} \{\boldsymbol{w}_i^{(t)}\} = \underset{\{i \ | \ a_i^{(t)}=0\}}{\operatorname{argmax}} \{\hat{\boldsymbol{x}}_i\Big(\hat{\boldsymbol{y}} - \hat{\boldsymbol{X}}^\top \boldsymbol{a}\Big)\}$$

Lemma 3.1 implies that $\boldsymbol{x}_j$ belongs to the frame. Therefore, $\boldsymbol{a}$ is also a solution of the three equivalent problems of Theorem 3.1. Algorithm 5 can hence be used in archetypal analysis to compute matrix $\boldsymbol{A}$. □

**Algorithm 5** Lawson and Hanson's active set algorithm

**Input:** Augmented matrix $\hat{\boldsymbol{X}}$ and augmented data-point $\hat{\boldsymbol{y}}$
**Output:** Solution $\boldsymbol{a}$ with $a_i \geq 0$
$\mathcal{P} = \emptyset$
$\mathcal{Z} = [1 \,..\, N]$
$\boldsymbol{a} = \mathbf{0}_N$
$\boldsymbol{w} = -\nabla f(\boldsymbol{a}) = \hat{\boldsymbol{X}}(\hat{\boldsymbol{y}} - \hat{\boldsymbol{X}}^\top \boldsymbol{a})$
**while** $\mathcal{Z} \neq \emptyset$ and $\exists\, \boldsymbol{w}[\mathcal{Z}] \geq \varepsilon$ **do**
  $l = \text{argmax}_j\{\, \boldsymbol{w}_j \mid j \in \mathcal{Z} \,\}$
  $\mathcal{Z} = \mathcal{Z} \setminus \{l\}$
  $\mathcal{P} = \mathcal{P} \cup \{l\}$
  $\hat{\boldsymbol{X}}_\mathcal{P} = \hat{\boldsymbol{X}}[:, \mathcal{P}] \in \mathbb{R}^{(M+1)\times|\mathcal{P}|}$
  $\boldsymbol{z} = \text{argmin}_{\boldsymbol{z}} \|\hat{\boldsymbol{X}}_\mathcal{P}^\top \boldsymbol{z} - \hat{\boldsymbol{y}}\|_2^2$
  $\boldsymbol{z}[\mathcal{Z}] = 0$
  $\mathcal{J} = \{\, j \in \mathcal{P} \mid \boldsymbol{z}_j \leq \varepsilon \,\}$
  **while** $|\mathcal{J}| > 0$ **do**
    $\alpha = \min\{\, \frac{s_j}{s_j - z_j} \mid j \in \mathcal{J} \,\}$
    $\boldsymbol{a} = \boldsymbol{a} + \alpha \cdot (\boldsymbol{z} - \boldsymbol{a})$
    **for** $i = 1, 2, \ldots, N$ **do**
      **if** $i \in \mathcal{P}$ and $|a_i| \leq \varepsilon$ **then**
        $\mathcal{P} = \mathcal{P} \setminus \{i\}$
        $\mathcal{Z} = \mathcal{Z} \cup \{i\}$
        $\hat{\boldsymbol{X}}_\mathcal{P}[:, i] = \mathbf{0}$
      **end if**
    **end for**
    $\boldsymbol{z} = \text{argmin}_{\boldsymbol{z}} \|\hat{\boldsymbol{X}}_\mathcal{P}^\top \boldsymbol{z} - \hat{\boldsymbol{y}}\|_2^2$
    $\boldsymbol{z}[\mathcal{Z}] = 0$
  **end while**
  $\boldsymbol{a} = \boldsymbol{z}$
  $\boldsymbol{w} = -\nabla f(\boldsymbol{a}) = \hat{\boldsymbol{X}}(\hat{\boldsymbol{y}} - \hat{\boldsymbol{X}}^\top \boldsymbol{a})$
**end while**

### Densification

Consider the example given in Figure 3.3: the point inside the quadrilateral is a convex combination of two combinations of three extreme points. However, to analyze the archetypes weights ($\boldsymbol{A}$) and their distribution, it is necessary that these do not depend on the order of the data-points. The solutions returned by NNLS do depend on the order and have up to $M+1$ non null coefficients, which is sparse if $K > M+1$. We propose to build an unequivocal and dense solution out of it.

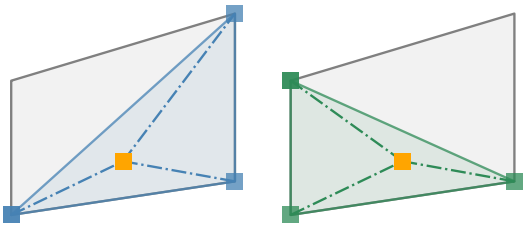


Figure 3.3: The point inside a quadrilateral is the convex combination of two different sets of three extreme points.

If a point is on, or is projected onto, a face of the archetypal polyhedron (AP), the solution of NNLS is unique (faces are sub-spaces of dimension up to $M-1$). The problem is thus concentrated on the data-points inside the polyhedron. Recall that, these can be reconstructed losslessly from the archetypes. An unequivocal representation is the mean of all lossless representations. This solution has also the property to be dense.

**Proposition 3.2.** *For any data-point $\boldsymbol{x}$ inside the archetypal polyhedron formed by $\boldsymbol{Z}$:*

*(i) The set of vectors $a$ of $\mathbb{S}^K$ such that $\boldsymbol{Z}^\top \boldsymbol{a} = \boldsymbol{x}$ is finite and non empty.*

*(ii) For any archetype $\boldsymbol{z}_k$, $k \in [1 \,..\, K]$, there exists a vector $\boldsymbol{a} \in \mathbb{S}^K$ such that $\boldsymbol{Z}^\top \boldsymbol{a} = \boldsymbol{x}$ and $a_k ! = 0$.*

*Proof.* (i) Let $\mathcal{C}$ be the set of vectors reconstructing losslessly $\boldsymbol{x}$ from the archetypes. Since NNLS returns such a solution, $\mathcal{C}$ is non empty. It is also finite, because any reconstruction relies on a combination of the $K$ archetypes that are in finite numbers.

(ii) Let $z_k$, $k \in [1 \,..\, K]$, be an archetype and consider the line $L$ passing through it and $\boldsymbol{x}$. Since $\boldsymbol{x}$ is inside the archetypal polyhedron and that the latter is the convex hull of the archetypes, $L$ intersects AP twice: in $\boldsymbol{z}_k$ and $\check{\boldsymbol{z}}_k$. The data-point $\boldsymbol{x}$ being inside the segment connecting these two points, it is a convex combination of them. The point $\check{\boldsymbol{z}}_k$ being on the border of the AP, it can be expressed as a convex combinations of the vertices (also archetypes) of the face supporting it. Hence $\boldsymbol{x}$ can be reconstructed in a lossless manner using a combination of archetypes, including $\boldsymbol{z}_k$. □

**Algorithm 6** Dense Archetypal Analysis

**Input:** Data-matrix $\boldsymbol{X}$, number of archetypes $K$
**Output:** Factor matrices $\boldsymbol{A}, \boldsymbol{Z}$
$\boldsymbol{A}, \boldsymbol{Z} = \text{ArchetypalAnalysis}(\boldsymbol{X}, K)$ (using NNLS)
$\mathcal{P}_{\boldsymbol{Z}} = \{$ all the combinations of $M+1$ column-vectors of $\boldsymbol{Z}$ $\}$
**for** $i = 1, 2, \ldots, N$ **do**
  **if** $\#\{\boldsymbol{a}_{il} > 0,\ l \in [1\,..\,K]\} = M+1$ **then**
    $\boldsymbol{a}_{i\cdot}, c = \boldsymbol{0}_K, 0$
    **for** $Z' \in \mathcal{P}_{\boldsymbol{Z}}$ **do**
      $\boldsymbol{a} = \text{NNLS}(\boldsymbol{Z}', \hat{\boldsymbol{x}}_i)$
      **if** $\|\boldsymbol{Z}^\top \boldsymbol{a} - \hat{\boldsymbol{x}}_i\|_2^2 < \epsilon$ **then**
        $\boldsymbol{a}_{i\cdot} = \boldsymbol{a}_{i\cdot} + \boldsymbol{a}$
        $c = c + 1$
      **end if**
    **end for**
    $\boldsymbol{a}_{i\cdot} = \frac{1}{c}\boldsymbol{a}_{i\cdot}$
  **end if**
**end for**

Algorithm 6 computes a dense solution from that of AA by computing the mean representations. The initial values of $\boldsymbol{A}$ and $\boldsymbol{Z}$ are computed using AA or Frame-AA and NNLS. If a point inside the AP is recognized by the number of non null coefficients of its reconstruction. For such a point, NNLS is run with all the $M+1$ combinations of archetypes, and the average of the solutions reconstructing exactly the data-point stored in the matrix weight $\boldsymbol{A}$. Note that if $K \leq M$, the archetypal polyhedron is flat, its interior is empty, thus all the data-points are considered on the border.

### 3.2.3 Generalization

In the previous section, we represented a single point as a convex combination of points of the frame. By doing so for every point in the data-set, we obtain the frame $\mathcal{F}$ of $\mathcal{X}$. Algorithm 7 summarizes this procedure, called *NNLS-Frame*, and Corollary 3.1 proves this claim.

**Corollary 3.1.** *Algorithm 7 computes the frame $\mathcal{F}$ of $\mathcal{X}$.*

*Proof.* According to Theorem 3.2, the indices of the positive coefficients of each vector $\boldsymbol{a}_i$ ($i \in [1\,..\,N]$) refers to points on the frame. Besides, every extreme point can only be defined as a convex combination of itself: the associated weight vector has a single positive coefficient. Therefore, the union of these indices yields the indices of the elements of the frame. □

**Algorithm 7** NNLS-Frame

**Input:** Augmented matrix $\hat{\boldsymbol{X}}$
**Output:** Indices of ext. points $\mathcal{F}$
$\mathcal{F} = \emptyset$
**for** $i = 1, 2, \ldots, N$ **do**
  $\boldsymbol{a} = \text{NNLS}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{x}}_i)$
  $\mathcal{F} = \mathcal{F} \cup \{\ j \in [1\ ..\ N]\ :\ a_j > 0\ \}$
**end for**

### Complexity Analysis

There are two main computations inside Algorithm 5: the computations of the negative gradient $\boldsymbol{w}$ and the resolution of the unconstrained least-squares problem

$$\boldsymbol{z} = \underset{\boldsymbol{z}}{\operatorname{argmin}} \|\hat{\boldsymbol{X}}_{\mathcal{P}}^{\top}\boldsymbol{z} - \hat{\boldsymbol{x}}\|_2^2.$$

The latter can be rewritten as $\boldsymbol{z}[\mathcal{P}] = (\hat{\boldsymbol{X}}_{\mathcal{P}}\hat{\boldsymbol{X}}_{\mathcal{P}}^{\top})^{-1}\hat{\boldsymbol{X}}_{\mathcal{P}}\hat{\boldsymbol{x}}$. The complexity of the unconstrained least-squares step is $O((M+1)\#(\mathcal{P})^2)$ and dominates the first main computation. Since no more than $M+1$ points are sufficient for the solution (Theorem 3.2), the outer `while`-loop is executed at least $M+1$ times and the average size of $\mathcal{P}$ is $\frac{1}{2}(M+1)$. Therefore, the complexity of the NNLS method is $O(\frac{1}{4}(M+1)^4)$ on average. Hence, the complexity of NNLS-Frame presented in Algorithm 7 does not exceed $O\Big(\frac{N}{4}(M+1)^4\Big)$.

### Computing the Frame efficiently

One way to speed up the frame computation is a divide-and-conquer approach. The underlying principle is stated in the following lemma.

**Lemma 3.2.** *Let $\mathcal{A}$ and $\mathcal{B}$ be non-empty discrete sets, it holds that:*

$$\text{Conv}(\mathcal{A} \cup \mathcal{B}) = \text{Conv}\Big(\text{Conv}(\mathcal{A}) \cup \text{Conv}(\mathcal{B})\Big).$$

The idea is as follows. Let $\mathcal{X}^{(1)} \cup \ldots \cup \mathcal{X}^{(K)}$ be a partition of $\mathcal{X}$ such that the cardinal $N_k$ of every subset $\mathcal{X}^{(k)}$ should be significantly smaller than that of $\mathcal{X}$, i.e., $N_k \ll N$. The assumption of having a pairwise disjunction is not necessary but reasonable. Instead of the whole data-set $\mathcal{X}$, Algorithm 7 is now executed on every subset $\mathcal{X}^{(k)}$ for $k = 1, 2, \ldots, K$. Finally, the frame of $\mathcal{X}$ is obtained by merging the frames of every subset and run our approach again on it. The procedure is summarized in Algorithm 8. Note that the `for`-loops of Algorithms 4, 7, and 8 can be parallelized.

**Algorithm 8** Divide-and-Conquer strategy of NNLS-Frame

**Input:** data $\mathcal{X}$, number of splits $K \in \mathbb{N}$
**Output:** indices of extreme points $\mathcal{F}$
$\mathcal{X} = \bigcup_{k=1}^{K} \mathcal{X}^{(k)}$ with $\mathcal{X}^{(k)} \cap \mathcal{X}^{(l)} = \emptyset$ for $k \neq l$
$\mathcal{F} = \emptyset$
**for** $k = 1, 2, \ldots, K$ **do**
  $\mathcal{F}_k = \text{NNLS-Frame}(\mathcal{X}^{(k)})$
  $\mathcal{F} = \mathcal{F} \cup \mathcal{F}_k$
**end for**
$\mathcal{F} = \text{NNLS-Frame}(\mathcal{X}_{\mathcal{F}})$

## 3.3 Experiments

### 3.3.1 Computing the Frame

In this section, we study the computation of the frame. The experiments rely on the same synthetic data-set[1] as in Dulá and López [89], which was generated according to a procedure described in López [175]. The data-sets consist of $N = 2{,}500, 5{,}000, 7{,}500, 10{,}000$ data-points with $M = 5, 10, 15, 20$ dimensions with a frame density of $1, 15, 25, 50, 75$ percent, respectively.

LP-based approaches [88, 204] discover up to one extreme point per iteration. On the other, accordingly to Theorem 3.2, NNLS-frame finds up to $M + 1$ extreme points. This behavior is illustrated in Figure 3.4. The graph shows the percentage of discovered extreme points against the percentage of iterations conducted on a synthetic data-set with $N = 2{,}500$ points in 5 dimensions with various frame densities.

[1] `http://www.people.vcu.edu/~jdula/FramesAlgorithms/SyntheticData/`

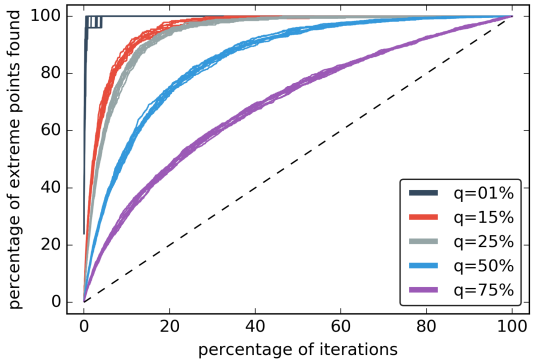



Figure 3.4: Percentage of discovered extreme points versus percentage of iterations on synthetic data-set with $N = 2{,}500$ data-points in $\mathbb{R}^5$.

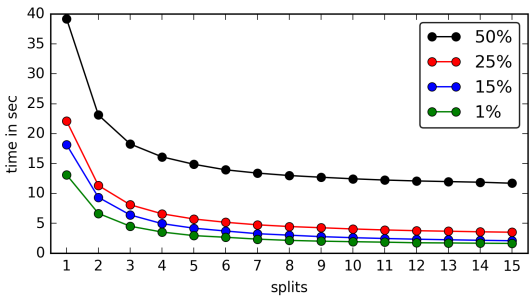



Figure 3.5: Timing results for the divide-and-conquer approach on synthetic data with $N = 10{,}000$ points with 5 dimensions.

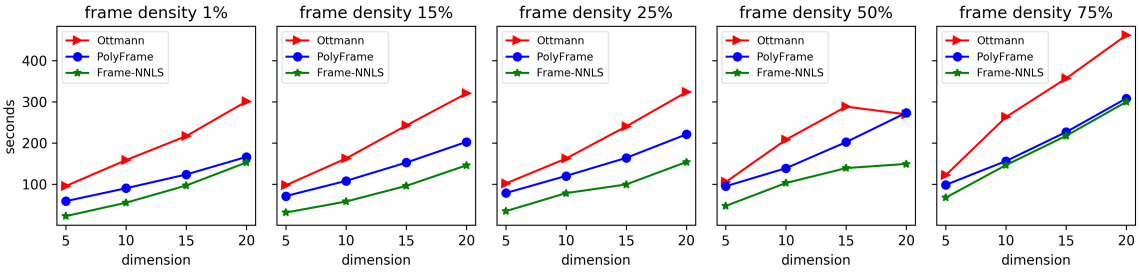



Figure 3.6: Timing results on synthetic data with $N = 10{,}000$ points.

The lower the frame density the faster the frame is being discovered. Even for a frame density of 75%, the discovery is faster than the linear growth (dashed line) expected from a LP-based approach. Hence, if an approximation of the frame is sufficient an early termination is possible.

In Figure 3.6, we compare NNLS-Frame to two LP-based baselines proposed by Ottmann et al. [204] and Dulá and Helgason [88]. The code is taken from Dulá and López [89]. For compatibility, we implemented our approach in the same programming language. The plot shows that for $N = 10{,}000$ and every configuration our approach is always faster than the baselines. We obtain similar results on the remaining data-set ($N = 2{,}500, 5{,}000, 7{,}500$).

The divide-and-conquer approach with three partitions cuts the run-time in half for each frame density configuration evaluated (Figure 3.5). Hence, even a small number of partitions leads to a substantial speed up. Note that, Although it is not the case here, the partitions can be processed in parallel.

### 3.3.2 Matrix Factorization

We compare now Frame-AA to several baselines including standard archetypal analysis (AA) [71], ConvexHull-NMF (CH-NMF) [239], Convex-NMF (C-NMF) [85] and standard NMF [163]. Frame-AA and AA are implemented in Python. For CH-NMF and C-NMF we use *pymf* [2], and *scikit-learn* [3] for NMF. Table 3.1 depicts the real world data-sets used for this experiment. The frame sizes and the frame densities are computed with our NNLS-Frame.

Table 3.2 reports on the results for $K = 6$ in terms of reconstruction error measured with the Frobenius norm. We obtain similar results for $K = 8, 10, 12$. The

[2] `http://pypi.python.org/pypi/PyMF`
[3] `http://scikit-learn.org`

Table 3.1: Real world data-sets sorted with respect to their frame density.

| Data-set | $N$ | $M$ | $P$ | frame density |
|---|---|---|---|---|
| Banking2 | 12456 | 8 | 715 | 5.74% |
| Banking1 | 4971 | 7 | 345 | 6.94% |
| USAFSurvey | 2420 | 6 | 368 | 15.21% |
| yeast | 1484 | 8 | 242 | 16.31% |
| Banking3 | 19939 | 11 | 4960 | 24.88% |
| SpanishSurvey | 600 | 5 | 150 | 25.00% |
| swiss-heads | 200 | 6 | 115 | 57.50% |
| skel2 | 507 | 10 | 431 | 85.01% |
| ozone | 330 | 10 | 308 | 93.33% |

number of iterations executed per algorithm is fixed to 100 in order to obtain fair results. We use random initializations for all algorithms and report averages over 36 repetitions. Our method Frame-AA yields similar error as AA. The lowest errors are achieved with NMF. The most comparable baseline CH-NMF, which approximates the frame instead of computing it exactly, returns an error approximately 20% larger. The worse results are returned by C-NMF. In summary, Frame-AA performs similarly to standard AA and much better than CH-NMF and C-NMF.

Usually, it is a priori not known how to choose the latent dimensionality $K$. A standard approach, the so-called *elbow rule*, requires several runs for different values of $K$ to be executed. In such a scenario, our approximative approach is particularly beneficial. Frame-AA requires the computation of the frame $\mathcal{F}$ before archetypes can be located. However, once the frame is complete, it can be used for testing any number of archetypes. It is, hence, interesting to see the cumulative time taken by the methods when evaluating several configurations, say $K = 4, 6, 8, \ldots, 16$.

We use the USAFSurvey data-set to illustrate this scenario (see Figure 3.7).

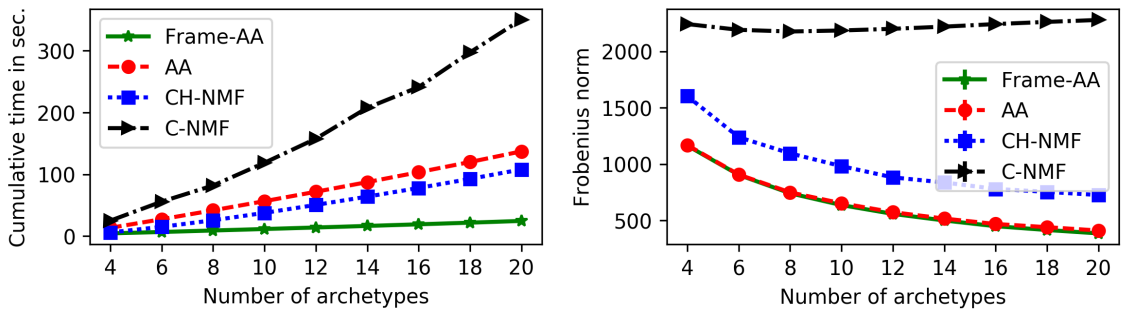



Figure 3.7: Cumulative time for several evaluations of $K$ as well as reconstruction error for USAFSurvey data-set.

Table 3.2: Average Frobenius norm reported on 36 repetitions with random initializations for $K = 6$. "-" indicates failure of the method due to negative data.

| Data-set | Frame-AA | AA | CH-NMF | C-NMF | NMF |
|---|---|---|---|---|---|
| Banking2 | 90.08 | 72.35 | - | - | - |
| Banking1 | 67.20 | 58.97 | - | - | - |
| USAFSurvey | 904.22 | 902.07 | 1239.67 | 2192.30 | 688.35 |
| yeast | 5.43 | 5.02 | 9.18 | 7.04 | 3.52 |
| Banking3 | 134.30 | 131.81 | - | - | - |
| SpanishSurvey | 94.84 | 93.51 | 117.91 | 254.92 | 32.14 |
| swiss-heads | 75.05 | 74.67 | 87.06 | 116.07 | 47.16 |
| skel2 | 64.84 | 64.87 | 77.78 | 101.73 | 51.75 |
| ozone | 1532.12 | 1669.70 | - | - | - |

Frame-AA is the fastest method for $K = 4$ despite that this first evaluation includes the computation of the frame (Figure 3.7 left). Since the frame is static for a data-set, it is reused in all the remaining computations. Note that we can obtain even faster computations for Frame-AA using the divide-and-conquer strategy as outlined in Algorithm 8.

The reconstruction error is shown in the right plot of Figure 3.7. Once again, Frame-AA yields errors similar to those of standard AA, which is computed on the whole data-set. The other baselines, C-NMF and CH-NMF, perform much worse.

### 3.3.3 Behavioral Archetypes

We study here the relationships between the amount of time the pupils see certain type of objects on the mBook and their motivation to study History. We consider the five most informative class of objects: text, text with a picture (Text/Pic), text with a picture linked to a gallerie (Text/Gal), galleries (always full screen), and expandable information boxes (Boxes). To compare sessions of various duration, we measure the ratio between the visibility time of each object in a session and the total duration. Since, there are more than five classes of objects (e.g. tile, loss of focus, etc.), the ratios do not sum up to 1.

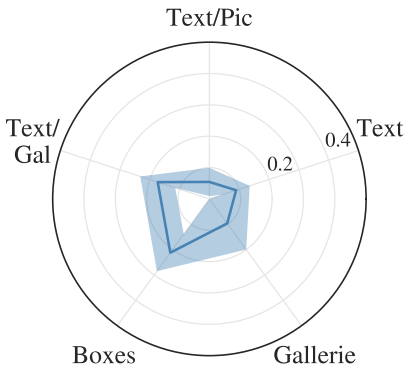



Figure 3.8: Average visibility time-ratio of the five most informative contents.

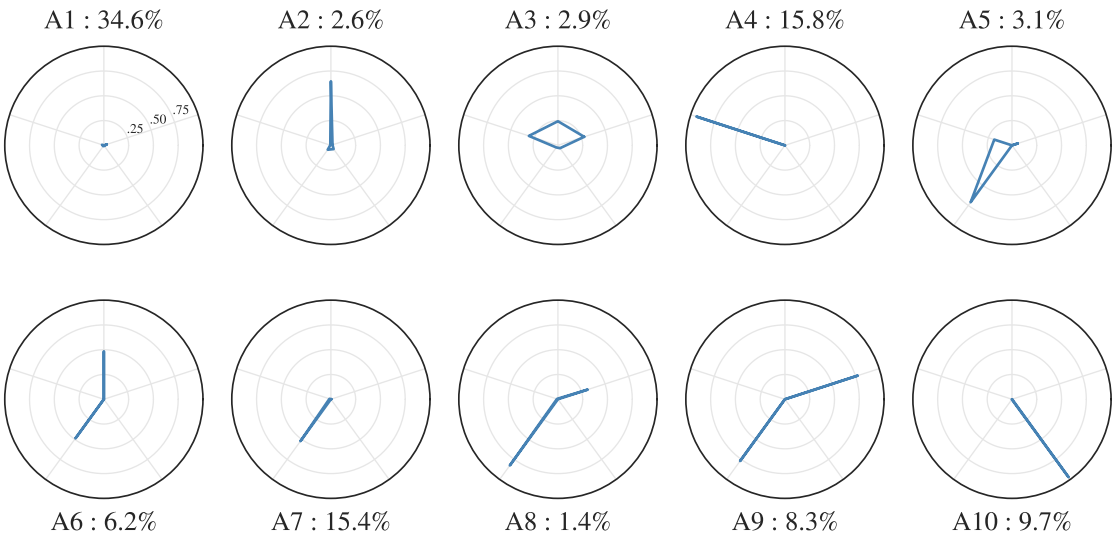



Figure 3.9: The ten archetypes found by Frame-AA. For legibility the labels are omitted (see Figure 3.8).

### Archetypes

We analyze here the 537 sessions from January 31st to July 11th 2017. After pre-processing, including the removal of sessions where none of the five contents were visible, 354 sessions remain. The average time distribution of the five selected objects is shown in Figure 3.8. The archetypes of the best solution out of 30 runs of Frame-AA are depicted in Figure 3.9, where the labels are not repeated for legibility. We compare only dense solutions computed using Algorithm 6.

The most frequent archetype A1 models users spending most of their time on non informative content. In contrast, for A3 each of the three text-based contents are visible during 25% a session. Archetypes A5, to A9 describe different ways of using the information-boxes.

### Psychometric Correlations

The naive approach to this analysis is to use the visibility distribution raw (Figure 3.8). The Pearson's $r$ correlation coefficients between the visibility ratio for each object and the motivation score are reported in Table 3.3. The only statistically significant relationship links the motivation to the time spent on galleries. The same analysis using the archetypal representation yields four significant correlations (Table 3.4). The significant correlation of the naive approach is here associated to A10 with a similar coefficient. These figures show that a high motivation implies more

Table 3.3: Pearson's $r$ coefficients between visibility ratio of five objects and the motivation score. Statistical significance ($p < 0.05$) is indicated in bold.

| | Text | Text/Pic | Text/Pic+ | Boxes | Gallerie |
|---|---|---|---|---|---|
| Motivation | -0.032 | 0.071 | 0.03 | -0.072 | **0.17** |

Table 3.4: Pearson's $r$ coefficients between archetypes' weight and the motivation score. Statistical significance ($p < 0.05$) is indicated in bold.

| A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 |
|---|---|---|---|---|---|---|---|---|---|
| **-0.143** | -0.01 | **0.107** | 0.016 | -0.038 | 0.098 | -0.038 | **-0.156** | -0.019 | **0.169** |

time spent on rich content (A3), and less time on less informative ones (A0). Paradoxically, there is a negative correlation between boxes and motivation. The very imbalance distribution of A8 suggests something unexpected: users see more boxes than texts. A deeper analysis of sessions close to A8 indicates that the visibility time does not itself cause this negative influence, but the number of clicks on boxes. These pupils tend to open and close a lot of these, without necessarily taking the time to read them. Consequently, information-boxes are artificially more often visible.

## 3.4 Conclusion

In this chapter we proposed a novel method for computing the frame of a data-set, i.e., the vertices of the convex-hull. While standard approaches like QuickHull are infeasible for data with more than three dimensions, we computed the frame by leveraging the well known active-set method for non-negative least squares problems, called NNLS. We provided a theoretical underpinning for our approach and conducted a series of experiments to compare the computation of the frame with two LP-based approaches to show our competitiveness.

We proposed an approximation of archetypal analysis, called Frame-AA, that restricts the optimization to the frame. Our heuristic is that a good approximation of the frame would reconstruct losslessly many of the data-points. Empirically, Frame-AA returned on par residual errors with standard archetypal analysis, while being much faster once the frame is known. This is a crucial characteristic as the optimal number of archetypes is generally not given a priori. Although, the computation of the frame may slow down the first evaluation, every subsequent evaluation saves time as the frame is then already known.

The archetypal analysis of the mBook data yielded insights inaccessible from a simpler approach. It highlighted correlations between usage of informative contents and the motivation of the pupils. However, the method has its limit. It could not explain the paradoxical negative correlation between motivation and information-boxes. Only an analysis of the sequences could reveal an abusive usage.

# Chapter 4

# Markov Chains and Periodic Behaviors

In this chapter, we investigate how users interact with contents over a week, especially between the different categories of pages: summaries, text pages, galleries. History lessons recurring weekly, they hence surely induce periodic behaviors. Several authors were involved, each focusing on the period in which they specialize. The chapters thus differ in their structure and content, i.e., galleries, information boxes, links. Besides, the nature of the behaviors sought calls for a sequential approach. We propose thus a Markov chain based model that captures periodic behaviors conditioned on the chapter.

Sequence-based model for log file analysis are common in computer science and are widely used to understand how web users navigate [135, 3]. These methods serve to detect navigation patterns that are indicative of future events [212, 232, 81] or user interests [10]. Patterns in sequences of page views have been studied using a variety of techniques, including relational models [8], association rules [74, 75], and k-nearest neighbors [34].

Previous works aiming at interpretable models have focused on modeling navigation sequences using Markov processes [54, 183, 81, 38, 260]. The underlying assumption is that navigational behaviors are memoryless and transitions to a state depend only on the precedent. Existing approaches focus mainly on the pure sequence of page views or categories without taking into account any contextual information. We argue that context, such as whether or not pupils are in school, is essential for drawing conclusions about any specific session or user. The model presented by Haider et al. [118] is a first attempt in this direction. They combine a mixture of Markov chains (MMC) with a model for connection times.

Our contribution goes further. We represent user sessions as fully observed Markov processes that are enriched by context variables: connection timestamps, chapters, and page category. We derive an Expectation-Maximization (EM) [80] algorithm to cluster sessions according to the learners' behavior when using the textbook. To highlight the expressiveness of our approach, we compare the results to a standard solution based on $k$-means [174]. While the latter yields trivial and insignificant groups, our approach groups sessions or users into clusters that can be easily visualized and interpreted.

The remainder is organized as follows. We present our probabilistic model in Section 4.1 and report on empirical results in Section 4.2. Section 4.3 presents a discussion of the results and concludes.

## 4.1 Time Nested Markov Chains

Let $\mathcal{S} = \{\boldsymbol{s}_i\}_N$ denote a set of $N$ iid user-sessions and $\mathcal{Z} = \{z_i\}_N$ their assignments to one of $K$ clusters. A session is a triplet $\boldsymbol{s}_i = (t^{(i)}, \mathbf{x}^{(i)}, \mathbf{c}^{(i)})$ defined by its connection time $t^{(i)} \in \mathbb{R}_{\geq 0}$, and the sequences of the chapter and category of the visited pages, noted, respectively, $\mathbf{x}^{(i)} = \langle x_j^{(i)} \rangle_{T_i}$ and $\mathbf{c}^{(i)} = \langle c_j^{(i)} \rangle_{T_i}$. Each sequence is prepended and appended by the auxiliary symbol $\infty$ capturing the initial and terminal events: $x_0 = x_{T_i+1} = c_0 = c_{T_i+1} = \infty$. The six chapters of the book together with the *homepage*, an *out* chapter that encodes the external pages, and $\infty$, form 9 possible realizations for every visited page. There are six different categories: *summary*, *text*, *gallery*, plus three auxiliary categories for the *homepage*, the external pages (*out*), and $\infty$.

### 4.1.1 Representation

We model the realization of a session $\boldsymbol{s}$ using a mixture model with $K$ components. The likelihood of $\boldsymbol{s}$ in the $k$-th component is given by

$$p(\boldsymbol{s}|z = k, \Theta) = p(t|\theta_k^t)p(\mathbf{x}|\theta_k^x)p(\mathbf{c}|\mathbf{x}, \theta_k^c). \tag{4.1}$$

The browsing process through chapters and categories are modeled by first-order Markov chains:

$$p(\mathbf{x}|\theta_k^x) = \prod_{t=1}^{T+1} \theta_k^x(x_{t-1}, x_t), \tag{4.2}$$

$$p(\mathbf{c}|\mathbf{x}, \theta_k^c) = \prod_{t=1}^{T+1} \theta_k^c(x_{t-1}, c_{t-1}, c_t). \tag{4.3}$$

The functions $\theta^x$ and $\theta^c$ are the transition probabilities of each Markov chain, (with omitted cluster's indicator):

$$\begin{aligned} \theta^{x,tr}(x_{t-1}, x_t) &= p(x_t|x_{t-1}, \theta^x), \\ \theta^{c,tr}(x_{t-1}, c_{t-1}, c_t) &= p(c_t|c_{t-1}, x_{t-1}, \theta^c). \end{aligned} \tag{4.4}$$

The category model depends on the chapters as we aim to capture different behaviors in each chapter. For example, some chapters may have their galleries visited more frequently than others.

### 4.1.2 Time Model

The model for the connection times is inspired by the approach described in [118]: a Gaussian mixture model with fixed number of components and parameters (mean and standard deviation), such that only the mixture weights are left to be optimized. Since we aim to extract periodic behaviors, we combine the Gaussian pdf with the tangent function to have a periodic distribution over a period $T \in \mathbb{R}_+$. For legibility, cluster indices are omitted.

**Lemma 4.1.** *For $T \in \mathbb{R}_+$, $(\mu, \sigma) \in \mathbb{R}^2$ the following function is $T$-periodic and defines a distribution over $\left[-\frac{T}{2}, \frac{T}{2}\right)$:*

$$\forall t \in \mathbb{R},\ p_T(t|\mu, \sigma) = \frac{1}{\operatorname{erfc}(\frac{1}{\sigma})T} \exp\left(-\frac{1 + \tan^2\left(\frac{\pi}{T}(t-\mu)\right)}{\sigma^2}\right). \tag{4.5}$$

*Proof.* The periodicity is inherited from the tangent. The function defines a distribution over $[-\frac{T}{2} + \mu, \frac{T}{2} + \mu)$, if the integral over this interval equals 1. This result can be deduced from Formula 7.5.11 of [1] (page 302):

$$\int_0^{+\infty} \frac{e^{-\frac{u^2}{\sigma^2}}}{1+u^2} = \frac{\pi}{2} \exp\left(\frac{1}{\sigma^2}\right) \operatorname{erfc}(\frac{1}{\sigma}), \quad \text{with } \sigma > 0.$$

□

We focus on daily and weekly behaviors to have two levels of granularity. Days and weeks are split into 48 and 42 slots of 30 minutes and 4 hours, respectively. This partition of the week is synchronized with schools' working hours: a morning slice between 08:00 and 12:00, one the afternoon between 12:00 and 16:00, and another one the evening between 16:00 and 20:00. The probability distribution of each time component is centered in its time slot. The standard deviations are chosen such that the probability at the extremities of the component is half the mode, i.e., $\sigma_T =$

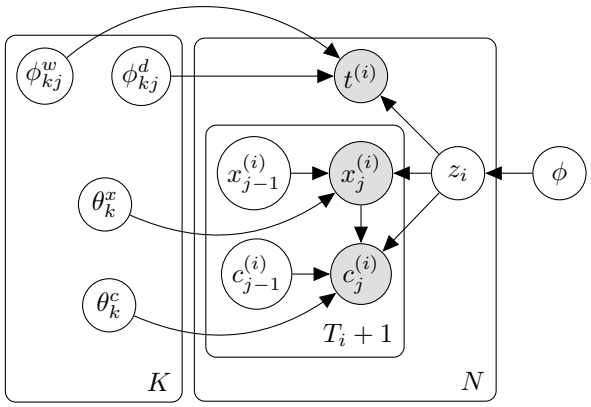



Figure 4.1: Graphical model of our nested mixture of Markov chains model; $N$ is the number of session and $T_i$ is the length of session $\boldsymbol{s}_i$.

$\tan(\frac{\pi}{2T})/\sqrt{\log(2)}$. Timestamps, $t$, are converted into the numbers of minutes since the beginning of the on-going week. The indices of the daily ($t_d$) and weekly ($t_w$) components associated to a timestamp $t$ are given, respectively, by:

$$t_d = \left\lfloor \frac{t}{30} \right\rfloor \pmod{48} + 1 \quad \text{and} \quad t_w = \left\lfloor \frac{t}{240} \right\rfloor \pmod{42} + 1. \tag{4.6}$$

The pdf of the $j$-th daily and weekly components are expressed as follows:

$$\begin{aligned} p^d_j(t_d) &= p_{48}\Big(t_d | \mu_j = j + 1/2, \sigma_{48}\Big) \\ &= \frac{1}{48\,\mathrm{erfc}(\frac{1}{\sigma_{48}})} \exp\left( -\frac{1 + \tan^2\left(\frac{\pi}{48}(t - \mu_j)\right)}{\sigma^2_{48}} \right), \end{aligned} \tag{4.7}$$

$$\begin{aligned} p^w_j(t_w) &= p_{42}\Big(t_w | \mu_j = j + 1/2, \sigma_{42}\Big) \\ &= \frac{1}{48\,\mathrm{erfc}(\frac{1}{\sigma_{42}})} \exp\left( -\frac{1 + \tan^2\left(\frac{\pi}{48}(t - \mu_j)\right)}{\sigma^2_{42}} \right), \end{aligned} \tag{4.8}$$

where $\sigma_{48} \approx 0.039$ and $\sigma_{42} \approx 0.045$. The daily and weekly mixture weights are denoted $\phi^d$ and $\phi^w$, respectively. The set of parameters of the time model is $\theta^t = \{\phi^d, \phi^w\}$ Finally, the likelihood that a user initializes a session at a time $t$ is given by:

$$p(t|\theta^t) = p_d(t|\phi^d) + p_w(t|\phi^w) = \sum_{j=1}^{48} \phi^d_j p^d_j(t_d) + \sum_{j=1}^{42} \phi^w_j p^w_j(t_w). \tag{4.9}$$

### 4.1.3 Optimization

Our nested mixture model is summarized by the plate diagram of Figure 4.1. Assuming independence of the user-sessions, the $\mathcal{Q}$-function (Section 2.3.1) is given by

$$\mathcal{Q}(\Theta, \Gamma) = \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \log\left( \phi_k p(t^{(i)}|\theta^t_k) p(\mathbf{x}^{(i)}|\theta^x_k) p(\mathbf{c}^{(i)}|\mathbf{x}^{(i)}, \theta^c_k) \right). \tag{4.10}$$

Recall that $\gamma_{ik} = p(z_i = k|s^{(i)}, \Theta)$ maximizes the function $\mathcal{Q}(\Theta, \cdot)$. The time model being also a mixture model itself, the maximization of $p(t^{(i)}|\theta_k^t)$ also requires known assignments for its optimization. We note $z_{ik}^d$ and $z_{ik}^w$ the assignments of session $i$ to the daily and weekly assignments in mixture $k$. The time mixture responsibilities are noted $\gamma_{ikj}^d$ and $\gamma_{ikj}^w$, respectively. The $\mathcal{Q}$-function for the daily time model of the $k$-th mixture is:

$$\mathcal{Q}(\theta_k^d, \Gamma_k^d) = \sum_{i=1}^{N} \sum_{j=1}^{48} \gamma_{ik} \gamma_{ikj}^d \log \left( \phi_{kj}^d p_j^d(t_d^{(i)}) \right). \tag{4.11}$$

We develop an EM-like algorithm [80, 54] to optimize the model. The main mixture and time mixture responsibilities are updated during the expectation phase, such that:

$$\begin{aligned} \hat{\gamma}_{ik} &= \frac{\phi_k \, p(\boldsymbol{s}_i|\Theta_k)}{\sum_{k'=1}^{K} \phi_{k'} \, p(\boldsymbol{s}_i|\Theta_{k'})}, \\ \hat{\gamma}_{ikj}^d &= \frac{\phi_{kj}^d \, p_j^d(t_d^{(i)})}{\sum_{j'=1}^{48} \phi_{kj'}^d \, p_{j'}^d(t_d^{(i)})}, \\ \hat{\gamma}_{ikj}^w &= \frac{\phi_{kj}^w \, p_j^w(t_w^{(i)})}{\sum_{j'=1}^{42} \phi_{kj'}^w \, p_{j'}^w(t_w^{(i)})}. \end{aligned} \tag{4.12}$$

We give the maximization update formulas of four of the parameters. They can be easily adapted to the other parameters.

$$\begin{aligned} \hat{\phi}_{kj}^d &= \frac{\sum_{i=1}^{N} \gamma_{ik} \, \gamma_{ikj}^d}{\sum_{j'=1}^{48} \sum_{i=1}^{N} \gamma_{ik} \, \gamma_{ikj'}^d}, \\ \hat{\theta}_k^x(g,h) &= \frac{\sum_{i=1}^{N} \gamma_{ik} \, \eta_{gh}(\boldsymbol{x}^{(i)})}{\sum_{h'=1}^{6} \sum_{i=1}^{N} \gamma_{ik} \, \eta_{gh'}(\boldsymbol{x}^{(i)})}. \end{aligned} \tag{4.13}$$

The function $\eta_{gh}(\boldsymbol{x}^{(i)})$ returns the number of transitions between chapters $g$ and $h$ in sequence $\boldsymbol{x}^{(i)}$.

### 4.1.4 User Model

Since we study user behaviors, we also need to model the users themselves. For this purpose, we consider the average model of the user's sessions. The likelihood of a user $u$ with $N_u$ sessions, $\{\boldsymbol{s}_1^u, \dots \boldsymbol{s}_{N_u}^u\}$, is given by:

$$p(u|\Theta) = \frac{1}{N_u} \sum_{i=1}^{N_u} p(\boldsymbol{s}_i^u|\Theta). \tag{4.14}$$

The formulas of the other parameters derive directly from the likelihood, and are similar to Equations 4.12 and 4.13. To not clutter the presentation, we omit them.

## 4.2 Experiments

To ease the discussion, we need to fix the number of clusters, $K$. In Figure 4.2, we show the evolution of three information criteria with the number of clusters: Akaike information criterion (AIC) [6], bayesian information criterion (BIC) [226], and AIC corrected for small sample size (AICc) [53]. A model selection based on these criteria fails. The two first criteria do not present a minimum. AICc may have one but for a too large number of clusters to be easily interpreted.

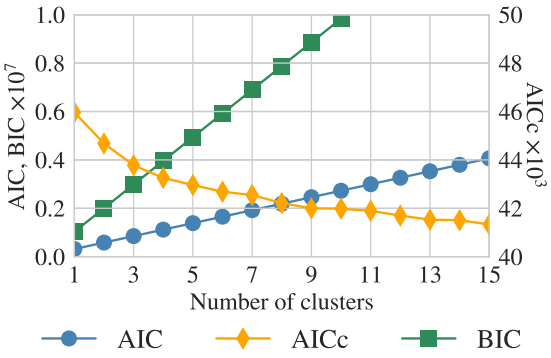



Figure 4.2: Evolution of AIC, BIC, and AICc with the number of clusters. None present a minimum within a reasonable range.

In the remainder, we fix the number of clusters to $K = 8$ as a trade-off between expressiveness and interpretability.

### 4.2.1 Comparison with $k$-Means

The first experiment demonstrates the expressiveness of our approach. We compare our probabilistic solution with a $k$-means [174] clustering. Since the latter operates on vector spaces, user-sessions are represented as vectors with fixed dimension. Reference to the null chapter and category are appended to match the length of the longest trajectory ($\max_{i \in [1..N]} T_i = 102$). The two corresponding sequences are then concatenated, augmented by two extra dimensions for $t_d$ and $t_w$. All in all, the dataset consists of 1,485 vectors with 206 dimensions. Our mixture model is trained 30 times; the best in terms of likelihood is kept. Clusters are named with respect to the size, i.e., the first is the largest.

We compare here the cluster distributions of the daily components (Figure 4.3). The solution returned by $k$-means (Figure 4.3 left ) splits the day in four periods: the early morning is shared between C3 and C6; C1 joins the mixture between 8:00 and 12:00; clusters C2 and C4 gather the majority of the afternoon activity, and the evening is the domain of C5, C7, and C8. The more complex coloring between 12:00 and 16:00 indicates the influence of more variables than the connection time during that period. Nevertheless, the simplicity of the result is clearly inappropriate for further processing or interpretation. It is unlikely that users present completely

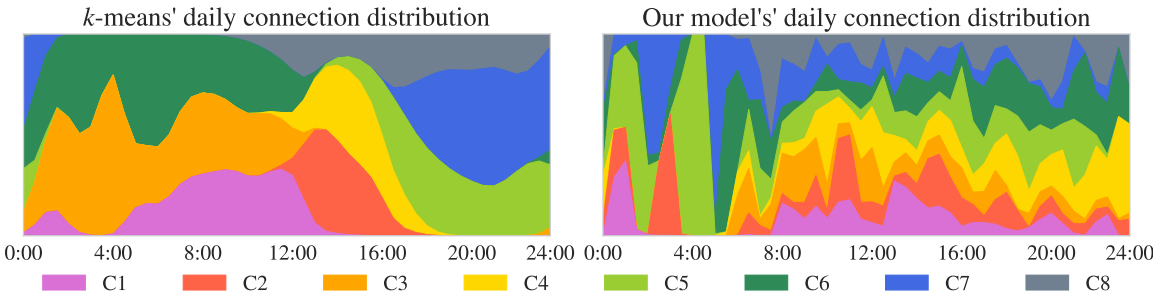



Figure 4.3: Daily distribution of clusters for $k$-means (left) and our model (right).

different behaviors in the morning and afternoon. Furthermore, an analysis of the transition matrices between chapters ($\theta^{c,tr}_{\cdot}$) reveals that $k$-means' clusters captured the same average behavior.

Figure 4.3 (right) shows the corresponding results for our probabilistic approach. The distribution of the clusters is more interesting since balanced across the day. The weekly distribution (Figure 4.4 top) changes dramatically when it is combined with the daily distribution (Figure 4.4 bottom). For example, according to the weekly distribution, C3 models a significant share of the sessions on Tuesdays early morning. However, when combined with the daily model, this share vanishes at the benefit of C5 and C6. Hence, we prefer to analyze the combined distribution.

The clusters' distribution presents peaks at different moments of the week. For example, C1 gathers a large share of the sessions on Monday and Saturday morning, C3 on Monday and Tuesday evening, and C5 over the weekends.

### 4.2.2 Session-based View

In this section, we discuss the behaviors captured by our model applied to sessions. Figure 4.5 shows for each cluster the distribution of the the main chapter, i.e., the most visited chapter during a session. Each heat-map of Figure 4.6 represents the transition matrix of selected clusters inside the Renaissance chapter between three categories: summaries (S), text (T), and galleries (G). Since the other categories are omitted, the row probabilities do not sum up to 1.

Except for C6 and C8, more than 60% of any cluster's sessions share the same main chapter. Five out of the eight groups have at least 26% of their sessions studying the Renaissance. This distribution is interesting and highlights the strength of our model. Firstly, although some clusters share the same topic, they appear at different moments of the week (see Figure 4.4). For example, C2 represents a large share of the sessions on Tuesdays, while Wednesday mornings are dominated by C7, but

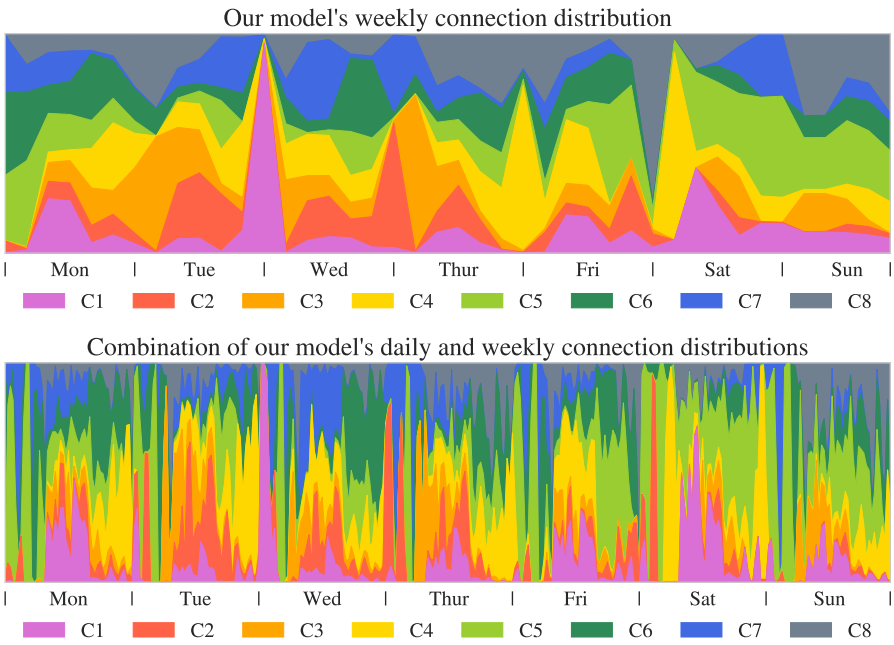



Figure 4.4: Weekly (top) and combined (bottom) distributions of clusters of our session-based model.

both have Renaissance as main chapter. Secondly, each cluster represent different browsing behaviors as shown in Figure 4.6. The five clusters with at least 25% of sessions studying the Renaissance, returned five different behaviors. These make up for all the possible patterns found, even those extracted for any other combinations of clusters and chapters. The sessions of C1 use the most the galleries. The others transition matrices present vanishing probabilities between text and galleries. Given that this is the only way to reach a gallery, we can deduce that sessions in these

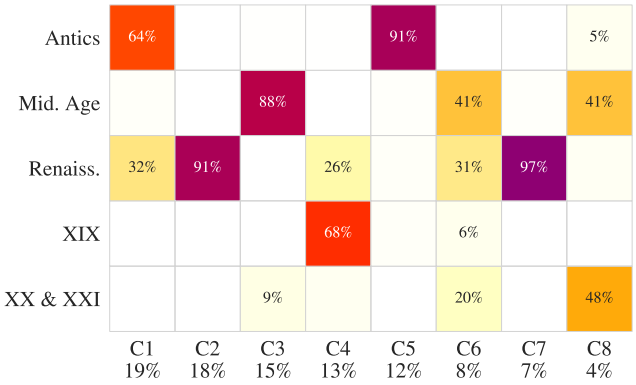



Figure 4.5: Main chapter distribution per cluster, with cluster's frequencies as labels. Percentages smaller than 5% are omitted.

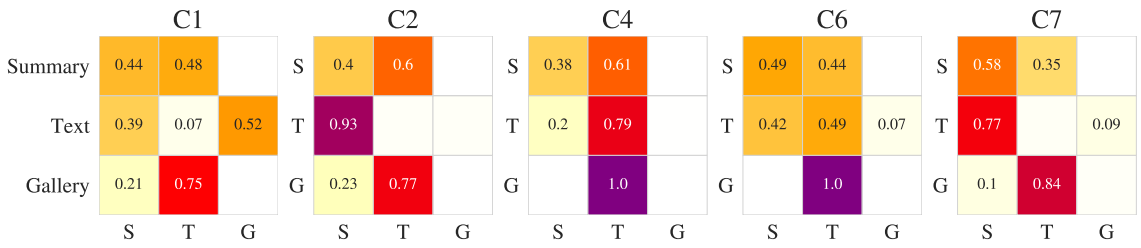



Figure 4.6: Transition matrices ($\theta^{tr}_{\cdot}$) restricted to summaries (S), text (T), and galleries (G), in the Renaissance chapter. Probabilities lower than 0.05 are omitted. Other categories being omitted, the probabilities per line do not sum up to 1.

clusters hardly opened any. The relatively high self-transition between text pages in C4 and C6 indicates a usage of the navigation bar to change the page. Based on these observations, we can state that our nested mixture model successfully distributed sessions with respect to the behavior and connection times.

### 4.2.3 User-based View

The clusters of the user model (Section 4.1.4) encode similar users rather than similar sessions, as in the previous section. Figure 4.7 shows the weekly distribution of the connection time for each cluster of users. Note that the distribution is here less balanced than in Figure 4.4. Cluster U2 represents a large majority of the connections on Fridays and Saturdays. Sessions on Tuesday and Thursday mornings are almost exclusively initiated by users of cluster U3. The group U8 dominates several evenings but represents approximately 1% of the users, i.e., four users.

The four largest clusters represent 85% of the population, and present an evident main chapter. The transition matrices between the categories in the main chapter of the four largest clusters are displayed in Figure 4.9 We recognize the behaviors captured by the session-based model C1, C4, and C7 (Figure 4.6) in U4, U2, and U1,

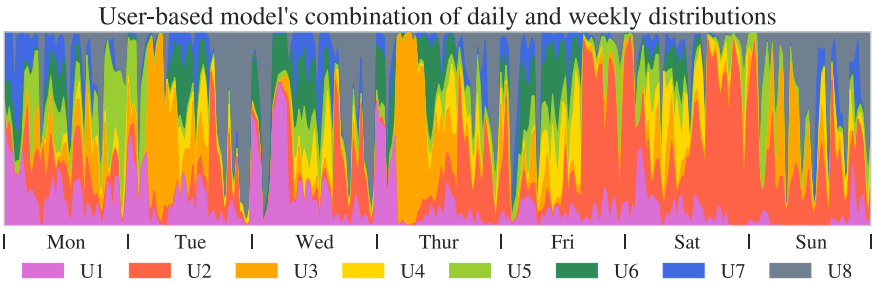



Figure 4.7: Combined weekly and daily distribution of clusters our user-based model.

respectively. The pattern of U3 is new. These users never visit galleries and use only the summaries to navigate between pages. Recall that the sessions of U3 also have a singular weekly distribution: they appear on Tuesday and Wednesday morning. We can conjecture that the behaviors of the pupils in U3 are influenced by their teacher.

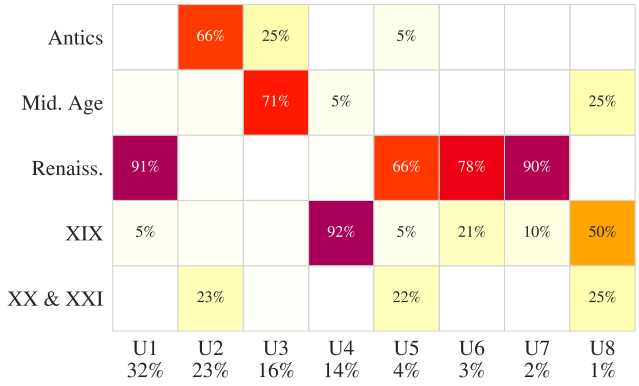



Figure 4.8: Main chapter distribution per cluster for the user-based model, with cluster's frequencies as labels. Percentages smaller than 5% are omitted.

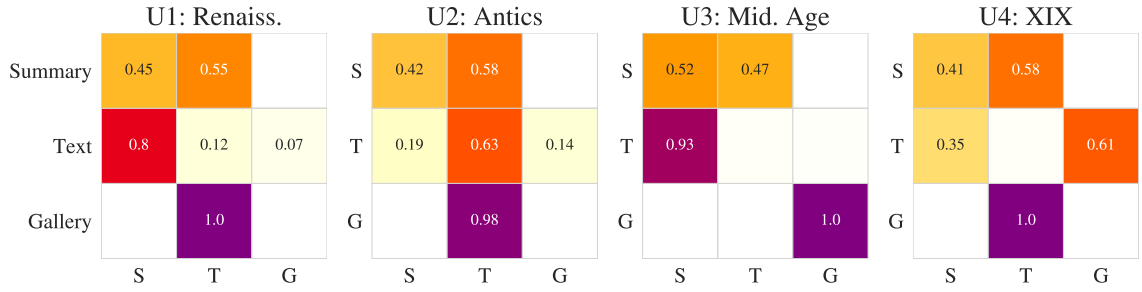



Figure 4.9: Transition matrices ($\theta^{tr}_{\cdot}$) restricted to summaries (S), text (T), and galleries (G) in the main chapter of the four biggest clusters of the user-based model. Probabilities lower than 0.05 are omitted.

## 4.3 Conclusion

We presented a context-aware mixture of Markov chains to represent user-sessions and proposed an Expectation-Maximization algorithm for optimization. We applied our approach to clustering user-sessions of the mBook. However, although the model rely on a coarse representation of the data (pages), a model selection based on information criteria fails. A solution is proposed in the next chapter. Our results are easy to interpret and visualize. There analysis suggested a possible influence of the teacher on the behaviors of the pupils. This line of research is further explored in Chapter 6.

# Chapter 5

# Bayesian Markov Chains and Scrolling Behaviors

Related studies reveal that time-on-page and cursor trajectories often serve as indicators for student engagement [68, 222]. However, in our case, the e-textbook is designed to be used on tablets in class and, hence, cursors or eye tracking are not available. We aim, though, to identify alternative indicators that are precise enough to capture characteristic traits of different behaviors.

One way to refine Markov models is to increase the order. However, the number of parameters grows exponentially with the order of the Markov chain and the number of states. Assuming a Dirichlet prior on the chain's order [192] yields inefficient computations and results that are difficult to interpret. Generally, approaches refining the Markov assumption tend to require unreasonably large data-sets [52, 24], which are rare in educational mining, where small data-sets are usual the norm.

Comforted by the results of our nested MMC (Chapter 4), we stick to first-order Markov chains. Clusterings obtained using Expectation-Maximization [80] (EM) successfully tell apart different types of users and behaviors. However, this optimization scheme falls short on two aspects. First, the greedy optimization strategy requires several random initializations. Second, a model selection based on information criteria (e.g., [6, 226]) is always more likely to fail as we refine the granularity of the analysis and add events.

One solution is to rely on a Bayesian interpretation [98, 112, 106]. Moreover, the number of mixtures can be learned by modeling the assignments as a realization of a Dirichlet process [215, 23, 238]. A remarkable example is the hierarchical Dirichlet process (HDP) [238] that generalizes the latent Dirichlet allocation [36]. HDP combines two Dirichlet processes to model a possibly infinite number of topics,

each defined using a variable amount of words. Several inference methods have been developed for it [242, 141, 248, 264].

We introduce here the infinite mixtures of Markov chains (iMMC) that aims to avoid the shortcomings of an EM-based approach. First, we embrace a Bayesian inference to be more robust against local maxima. Secondly, to avoid fixing the number of clusters, we let the mixture weights arise from a Dirichlet process. That way, our model is flexible enough to process users' behavior at the event level. Unfortunately, often in practice, the complete list of events is not necessarily accessible to the practitioner (if to anyone). Hence, we nest a second Dirichlet process to govern the prior distribution of the events. We present an analysis of the scrolling patterns from the mBook using iMMC. The conclusions constitute novel insights that may impact future developments and design decisions of electronic textbooks.

The chapter is organized as follows. In Section 5.1, we introduce our model and a computable approximation. Section 5.2 contains two experiments. The first one compares iMMC to baselines on a synthetic data-set. Then, we analyze scrolling behaviors in the mBook and draw conclusions on the correlations with the pupils' performance in history.

## 5.1 Infinite Mixture of Markov Chains

### 5.1.1 Description

The infinite mixtures of Markov chains (iMMC) is defined as follows. The mixtures prior $\boldsymbol{\phi}^*$ is a realization of a symmetric DP with base measure $\alpha H_0$, $\alpha > 0$. An assignment is sampled from a categorical distribution with $\boldsymbol{\phi}^*$ as a parameter. Each cluster has its own alphabet $\mathcal{M}_k$ of events. The prior distribution of the latter in cluster $k$, $\boldsymbol{\psi}_k^*$, is a realization of another symmetric DP with base measure $\beta H_1$, $\beta > 0$. The transition probabilities' distribution from an event $m \in \mathcal{M}_k$ is a Dirichlet distribution with parameter $\lambda \boldsymbol{\psi}_k^*$, where the hyper-parameter $\lambda > 0$ controls the variance of the distribution. In summary:

$$\begin{aligned} \boldsymbol{\phi}^* &\sim \mathrm{DP}_{\alpha H_0} \\ \boldsymbol{\psi}_k^* &\sim \mathrm{DP}_{\beta H_1} \\ z &\sim \mathrm{Cat}(\boldsymbol{\phi}^*) \\ \theta_k(m, \cdot) &\sim \mathrm{Dir}(\lambda \boldsymbol{\psi}_k^*) \\ \boldsymbol{s}, z &\sim \phi_z \prod_{t=1}^{T+1} \theta_z(s_{t-1}, s_t) \end{aligned} \tag{5.1}$$

Since, we assume both DPs symmetric, the indices of the stochastic processes are omitted. A plate diagram of the model is depicted in Figure 5.1. The structure is similar to the hierarchical Dirichlet process (HDP) [238]: topics and words are replaced by clusters and events, respectively. The documents/sessions are modeled here as realizations of Markov chains instead of bag-of-words.

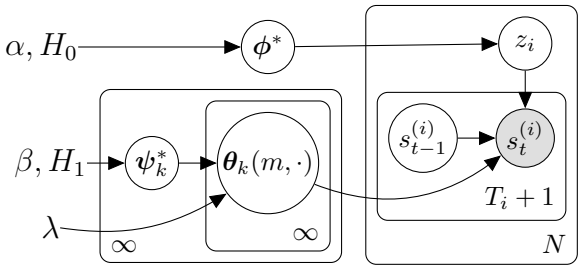



Figure 5.1: Graphical representation of the infinite mixtures of Markov chain with three hyper-parameters $\alpha$, $\beta$, and $\lambda$. The two DPs correspond to the two arrows originating from $\alpha, H_0$ and $\beta, H_1$.

**Algorithm 9** Blocked-Gibbs sampler for iMMC

**Require:** $\mathcal{S} = \{\boldsymbol{s}_i\}_N$: input data, $K, M, \alpha, \beta, \lambda$: hyper-parameters.

Initialize $\boldsymbol{n} = \mathbf{0}_N, \boldsymbol{m} = \mathbf{0}_{K\times M}, \boldsymbol{w} = \mathbf{0}_{K\times M\times M}$
**repeat**
  Sample $\phi \sim \mathrm{Dir}(\boldsymbol{n} + \frac{\alpha}{K}\mathbf{1}_K)$.
  **for** $k$ in $1 \dots K$ **do**
    Sample $\boldsymbol{\psi}_k \sim \mathrm{Dir}(\boldsymbol{m} + \frac{\beta}{K}\mathbf{1}_K)$.
    **for** $m$ in $1 \dots M$ **do**
      Sample $\theta_k(m,\cdot) \sim \mathrm{Dir}(\boldsymbol{w} + \lambda\,\boldsymbol{\psi}_k)$.
    **end for**
  **end for**
  Reset $\boldsymbol{n} = \mathbf{0}_N, \boldsymbol{m} = \mathbf{0}_{K\times M}, \boldsymbol{w} = \mathbf{0}_{K\times M\times M}$
  **for** $i$ in $1 \dots N$ **do**
    Compute and normalize $\boldsymbol{l}_i = \langle \phi_k p(\boldsymbol{s}_i|\boldsymbol{\theta}_k)\rangle_K$.
    Sample $z_i \sim \mathrm{Cat}(\boldsymbol{l}_i)$.
    Increment $\boldsymbol{n}(z_i) \leftarrow +1$.
    **for** $t$ in $1 \dots T_i$ **do**
      Increment $\boldsymbol{m}(z_i, s_t^{(i)}) \leftarrow +1$.
      Increment $\boldsymbol{w}(z_i, s_{t-1}^{(i)}, s_t^{(i)}) \leftarrow +1$.
    **end for**
    Increment $\boldsymbol{w}(z_i, s_{T_i}^{(i)}, s_{T_i+1}^{(i)}) \leftarrow +1$.
  **end for**
**until** convergence
**return** Averages over the *last* iterations of $\boldsymbol{\phi}$, $\boldsymbol{\psi}$, $\boldsymbol{\theta}$.

### 5.1.2 Inference

We make use of a computationally efficient approximation that is the $k$-weak limit approximation [133]: the (symmetric) Dirichlet processes are replaced by (symmetric) Dirichlet distributions. The model is thus equivalent to a Bayesian mixture model with two Dirichlet distributions: one for the mixtures of order $K >> 1$ and one for the events' of order $\#\mathcal{M} = M >> 1$. Moreover, we assume a single alphabet $\#\mathcal{M} = M$ shared among all the clusters. Remember that one of the elements of $\mathcal{M}$ represents the auxiliary event $\infty$. The approximated model is as follows:

$$\begin{aligned} \boldsymbol{\phi} &\sim \mathrm{Dir}(\alpha) \\ \boldsymbol{\psi}_k &\sim \mathrm{Dir}(\beta) \\ z &\sim \mathrm{Cat}(\boldsymbol{\phi}) \\ \theta_k(m,\cdot) &\sim \mathrm{Dir}\left(\lambda\boldsymbol{\psi}_k\right) \\ \boldsymbol{s}|z &\sim \phi_z \prod_{t=1}^{T+1} \theta_z(s_{t-1}, s_t) \end{aligned} \tag{5.2}$$

A maximum a posteriori estimation (MAP) of the model is learned using the collapsed and blocked-Gibbs sampler detailed in Algorithm 9. The sampler is *blocked* because the parameters are sampled as vectors, instead of individually. The expectancies of the events represented by each coefficient of each parameter are given as follows, for $k \in [1\,..\,K]$ and $(u,v) \in [1\,..\,M]^2$:

$$\begin{aligned} \mathbb{E}[\phi_k|\boldsymbol{\Theta},\mathcal{Z},\mathcal{S}] &\propto \boldsymbol{n}(k) + \alpha \\ \mathbb{E}[\psi_k(u)|\boldsymbol{\Theta},\mathcal{Z},\mathcal{S}] &\propto \boldsymbol{m}(k,u) + \beta \\ \mathbb{E}[\boldsymbol{\theta}_k(u,v)|\boldsymbol{\Theta},\mathcal{Z},\mathcal{S}] &\propto \boldsymbol{w}(k,u,v) + \lambda\,\psi_k(u) \end{aligned} \tag{5.3}$$

where we used the notations of Algorithm 9: $\boldsymbol{n}(k)$ is the number of session assigned to cluster $k$; $\boldsymbol{m}(k,u)$ is the count of event $u$ in the sessions assigned to cluster $k$; $\boldsymbol{w}(k,u,v)$ is similar to the latter but counts the transitions from $u$ to $v$.

## 5.2 Experiments

We evaluate first the clustering performance of our model in controlled scenarios to understand its effectiveness and to shed light on extreme cases. Then, we apply iMMC to the mBook and show that some scrolling patterns correlate with the success of the pupils.

### 5.2.1 Synthetic Data-set

In this section we compare the clustering performance of iMMC to the traditional mixture of Markov chains (MMC). We pick the latent Dirichlet allocation (LDA) [36] as an additional baseline to assess the importance of the sequential information contained in the observations.

We generate two synthetic scenarios to generate different sets of clusters. Scenario I is made of two clusters/behaviors with disjoint sets of nodes. Scenario II adds to Scenario I a variation of it: same nodes but different graphs. The four behaviors are displayed in Figure 5.2, where the first column corresponds to Scenario I.

For each scenario, we evaluate the algorithms on data-sets consisting of 50 to 1,000 sessions generated as follows. First, a cluster is selected uniformly at random. Then, the generative process is repeated until the desired number of sequences is reached. We use a single set of hyper-parameters ($\alpha = \beta = \lambda = 2$) and set the upper-bound $K$ to ten times the true number of clusters. MMC and LDA optimized with the correct number of clusters. We report on clustering performance in terms of the averaged adjusted Rand index [131] (ARI) over 30 runs. The evolution of the ARI with size of the data-sets is displayed on Figure 5.3.

First, LDA lags in every setting. It never reports an average ARI higher than 0.5. Even though MMC is trained with the correct number of clusters, and iMMC has to

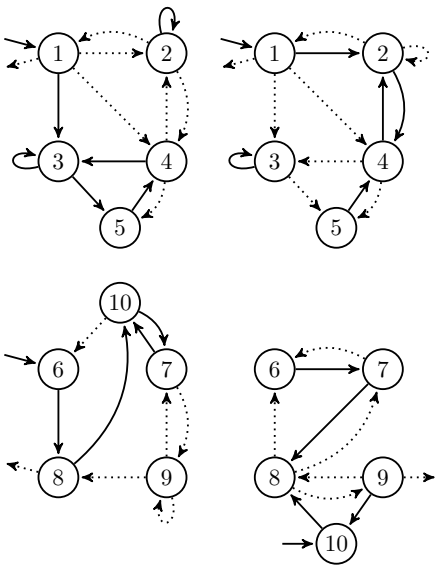



Figure 5.2: Generative processes of scenarios I (left column) and II (all four). Dotted arrows represent low transition probabilities ($< .5$).

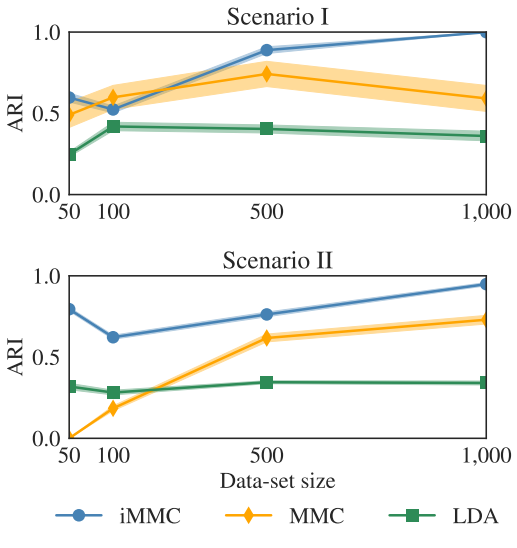



Figure 5.3: Averaged ARI and standard errors of the clusterings returned by the three methods on data-sets based on both scenarios.

adjust it to the data, the latter returns on average better ARI than MMC. In the case of small data-sets based on Scenario I, MMC and iMMC are equivalent. However, iMMC is more stable. In all the experiments, the standard error (shade) of iMMC stays small, while for Scenario I, MMC shows an unsteady behavior.

### 5.2.2 Scrolling Patterns

In this section, we present insights on the pupils's usage of the mBook. We show that identified usage patterns correlate with psychometric scores.

We define and differentiate 75 atomic events that a user can trigger, ranging from pressing a button to various scrolling performances. The latter are further divided into 9 events : *direction.duration*. The direction can be *up*, *down*, or *fix* if the movement is of less than 10 pixels. The duration can be *fast*, *medium* or *slow* for event duration of respectively less than 1 second, between 1 and 3 seconds, and more than 3 seconds. In the following, node names are abbreviated using only the first letter. For example *down.fast* is reduced to *d.f.*

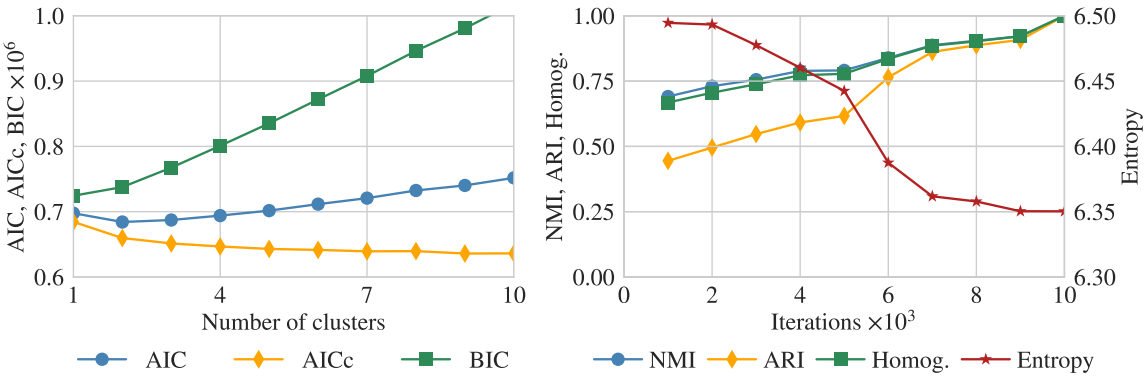



Figure 5.4: Evolution of several model selection criteria. Left: BIC, AIC and AICc for MMC. Right: Relative NMI, ARI, homogeneity and entropy for iMMC.

A model selection using information criteria for MMC fails, as shown in Figure 5.4 (left). This is not surprising as they are known to perform poorly when the order of magnitude of the sample size is not greater than that of the number of parameters [107]. The evolution of three information criteria AIC [6], AICc [53], and BIC [226] is depicted for different numbers of clusters. Every point corresponds to the best clustering in term of likelihood out of 30 repetitions. Theoretically, the minima of these curves give the optimal solutions. Here, the criteria grow almost linearly with $K$. The AIC curves does reach a mininum for two clusters, but this is not a relevant solution. Thus information criteria do not allow to draw conclusion.

By contrast, our iMMC approach successfully clusters the data using $\alpha = 2$, $\beta = 1.5$, $\lambda = 2.4$, $K = 100$ and 10,000 iterations. After every 1,000 iterations, an intermediate clustering is computed as the average of the last 1,000 iterations. After the first 1,000 iterations, 34 clusters are open. The final solution settles on 32 clusters. The evolution of the optimization is shown in Figure 5.4 (right). The blue line (left scale) represents the evolution of the normalized mutual information (NMI, blue), the adjusted Rand index (ARI, orange) [131], and the homogeneity score [219] (green), all relatively to their final value. The red line (right scale) refers to the entropy of the clustering for the actual iteration. After 7,000 iterations, the NMI indicates that the clustering is already 90% similar to the final one. The decrease in entropy and increase of ARI reveal that the algorithm merges clusters. The plateau after 7,000 iterations indicates fine granular changes of cluster memberships.

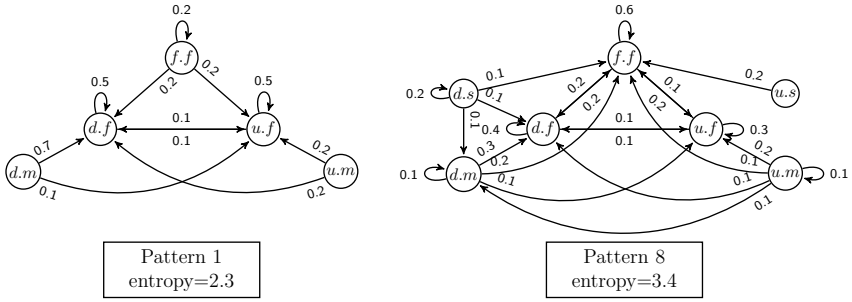



Figure 5.5: Two remarkable scrolling patterns extracted from the mBook.

The solution contains eight clusters with at least 20 sessions. We focus on their scrolling patterns. Figure 5.5 displays two patterns realizing the smallest and highest entropy, respectively. Note that the weights do not sum up to one, as we ignore outgoing edges to non-scroll events in this analysis. In Pattern 1, *fix.fast* cannot be reached from any other type of scroll. It either starts a scrolling sequence, or it indicates a misusage or hesitation of the user. In both patterns, users tend to not transit to slower scrolls. This behavior may result from "long" scrolls corrected by fast ones. This is a typical behavior of users who scroll while reading. In Pattern 8, the high self-transition probabilities of *down.slow* and *fix.fast* support this interpretation. The emission of *fix.fast* probably comes from a finger held on the screen after a scroll.

### 5.2.3 Psychometric Correlations

To correlate the psychometric scores with our clustering, we represent clusters using the average scores of the pupils who have sessions in the cluster. The distribution is

shown in the first row of Figure 5.6. The clusters are organized from top to bottom according to the entropy of their pattern. Patterns 1 and 8 (see Fig. 5.5) are extracted from clusters 1 and 8, respectively. Both patterns are often observed among pupils with high competencies in history. Therefore, these patterns may serve as behavioral indicators for a user's competency. Seemingly, knowledgeable pupils prefer simpler scrolling patterns (top of the plot). Cluster 2 contains highly knowledgeable and motivated pupils that possess high computer skills. The pupils in cluster 6 are also motivated but do not possess such high ICT skills.

We compute Pearson's correlation coefficients adjusted for small sample sizes [203], between the 81 possible transition probabilities (between scrolls) and the five scores of the eight largest clusters. The maximum and minimum correlations for the assessed variables are reported in Table 5.1. Except for motivation, every highly, positively correlated transition changes the direction to a *up.fast* . Knowledge correlates almost perfectly with *down.medium* $\rightarrow$ *up.fast* and *down.slow* $\rightarrow$ *down.medium*. A finer analysis shows that only Pattern 8 contains these two edges, but their influences cancel out. Indeed, as we can see in the first row of Figure 5.6, users in cluster 8 present a relatively lower knowledge score in comparison to pupils in clusters with simpler scrolling pattern.

The second row of Figure 5.6 reports on the transition probabilities in each cluster of the most strongly correlated transition with the column's score, as reported in

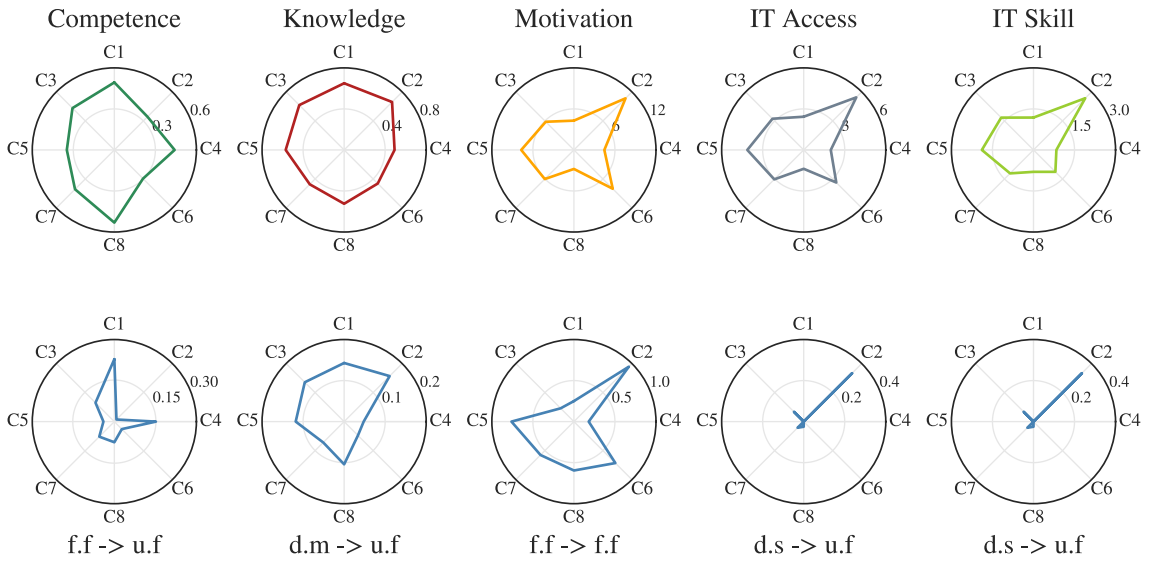



Figure 5.6: Score and probability distribution of the most correlated transition for each of the eight biggest clusters.

Table 5.1: Most strongly correlated event transitions with each score.

| Score | Max Corr. | Event | Min Corr. | Event |
|---|---|---|---|---|
| Competency | 0.697 | $f.f \to u.f$ | -0.719 | $u.m \to u.s$ |
| Knowledge | 0.962 | $d.m \to u.f$ | -0.947 | $d.s \to d.m$ |
| Motivation | 0.748 | $f.f \to f.f$ | -0.714 | $f.f \to u.f$ |
| IT Access | 0.751 | $d.s \to u.f$ | -0.735 | $f.f \to d.f$ |
| IT Skill | 0.837 | $d.s \to u.f$ | -0.743 | $d.m \to d.s$ |

Table 5.1. To not clutter the presentation, we consider only the positive correlations. The similar shape of the average knowledge score (first row second column) and of the transition probability (second row second column) in each cluster suggest that the latter could serve as an indicator. The same can be said for the pupils' motivation. Regarding competency score, a high probability of *fix.fast* $\to$ *up.fast* seemingly also implies a high score in the assessment. However, the opposite does not hold. This transition has a low probability in Pattern 8, but the average competency score of the cluster is the largest.

## 5.3 Conclusion

We presented a Bayesian non-parametric model to reach a finer resolution of the behaviors. The challenge that represents the larger number of parameters in comparison to the size of the data-set is tackled using Dirichlet processes. However, such technique hinders the efficiency of the inference of the model. Therefore, our model, the infinite mixtures of Markov chains (iMMC), relies on a degree $k$-weak limit approximation. Controlled experiments showed significant improvements over related approaches. Regarding the mBook, iMMC identified scrolling behaviors as characteristics of pupils' online navigational habits but also of their performance. Furthermore, we showed that certain transition probabilities between scrolling events correlates strongly with several psychometric scores. This suggests that the monitoring of the scrolling behavior could be used to predict pupils' performance.

# Chapter 6

# Trajectories and Online Behaviors

We have reviewed and proposed several methods to extract patterns from log files, mostly based on mixtures of Markov chains [41, 42, 103, 54]. Because of the Markov condition, these approaches can be qualified as local models. Indeed, the focus is on event or state transitions and do not consider historical and future events. Higher-order Markov chains could possibly handle longer sequences that condition these transitions, but the computations become rapidly intractable. In this chapter, we choose to literally extend the navigation metaphor and build a structure to handle sessions as is they were spatio-temporal trajectories.

Temporally structured data are ubiquitous as sensor measurements are usually attached with timestamps and naturally form a sequence or trajectory. Examples arise in a great deal of different areas including video surveillance [196], traffic monitoring [148], anomaly detection [250], analyses of GPS data [263], recommendation [113], mining usage in electronic text books [40], urban planning [262, 94] or preservation efforts [124, 230, 37].

When dealing with sequential data, a simple approach is to discretize the universe and pursue a grid-based approach. A movement in some space can then be processed as an image [210, 83] or as a sequence of states [137]. The latter representation often leads to approaches such as latent Dirichlet allocation [36], hidden Markov models [213] or neural networks [220]. However, the discretization causes a non-negligible loss of information and may introduce noise.

Instead, the similarity between two trajectories can be quantified directly via their time-series. There are many different similarity measures, and many of them are based on metric axioms [229]. These measures differ in many characteristic traits, including whether they satisfy the triangle inequality [224], whether they are bounded [246, 123], or their computational costs [92, 223]. Perhaps, most importantly, how the measures deal with time [200].

This chapter provides a theoretically grounded framework to classify and understand existing similarity measures for sequential data. Based on this framework, we use an upper-bound of the Kullback-Leibler (KL) divergence between distributions induced by two trajectories to devise a novel similarity measure for trajectories. The distributions generalize the Laplace distribution and lead to a formula that corresponds to the normalized point-wise distance with a penalty for the difference in duration. Empirical evidence shows that our measure performs on par with state-of-the-art measures in several scenarios, while being theoretically grounded and efficient. Using the data from the mBook, we show that our contribution not only distinguishes user-sessions given the topics studied, but also quantifies the differences in the pupils' navigation behaviors. Furthermore, these behavior patterns influence pupils performances and depend on the teaching style.

The remainder is organized as follows. The next section reviews related work. Section 6.2 introduces notations and the key definitions. Seven existing measures are extensively studied according to a novel classification scheme in Section 6.4. On that basis, we present our new measure in Section 6.5. We report on empirical results using real trajectory data from taxis, buoys, and finally from the mBook in Section 6.6. Section 6.7 concludes the chapter.

## 6.1 Related Work

A distance metric satisfies very restrictive properties. Many of the so-called trajectory distances proposed in the literature violate one or another property. Besides, the term *similarity* is used often, although it is not well-defined in a mathematically sense.

Lin et al. [172] define a similarity from an information theoretic point of view with the construction of ontologies in mind [28, 73, 168]. Santini et al. [224] consider mental aspects behind similarities. After reviewing the work of several psychologists [229, 93], they generalize the triangle inequality to better model inconstancies of human. There have been several attempts to establish a list of properties, in addition to the axioms, a trajectory measure should satisfy. For example, coping with asynchronous trajectories [199] or different sample rates is crucial as not all the instances in a dataset might come from the same type of sensor. Wang et al. [250] compare the robustness of several measures to noise in the temporal and spatial components. Besse et al. [32] indirectly impose constraints on the measure and provide a list of expectations on sequential clusters.

In general, two groups of similarity measures for sequences can be distinguished: the first group is based on variants of the edit-distance and the second aims to quantify the difference of shapes. Edit-distances are derived from the Levenshtein distance [166] that compares strings. The most direct descendants are the edit distance on real sequences (EDR) [62] and edit distance with real penalty (ERP)[61]. The former uses a binary cost function, while the latter relies on the distance between two locations. The longest common subsequences (LCSS) problem [30] gives rise to an eponymous measure for trajectories [246], featuring also a binary cost function. Its specificity is that it may skip some elements of the trajectory, which renders it a close neighbor of warping distances [187]. The most prominent representatives of the latter are dynamic time warping (DTW) [31] and the Fréchet distance [99]. Optimal subsequence bijection [160] is a relaxation of the DTW that excludes outliers to preserve a small total distance.

The second large group of measures includes the Hausdorff distance [123] and its modifications [14], the one-way distance [171] and its derivatives such as SSPD [32] and TIDE [199]. Note that the Euclidean distance has also been utilized together with trajectories, either normalized [165, 105, 200, 86, 40] or unnormalized [95, 100]. For continuous problems, the area spanned by two trajectories is often studied [101], e.g., STLIP [211] computes this areas but ignore temporal constraints.

While edit-distance and their variants follow a clear definition, shape-based measures appear domain specific and lack a common formal ground. The theoretical framework built in this chapter provides a more rigorous classification of trajectory measures.

Often, the goal of a task at hand is to effectively summarize similar trajectories with a clustering. Instances of a cluster are represented by their closest centroid. However, the idea does not translate one-by-one to trajectories with varying durations or sampling rates. The time-wise average trajectory [205] constitutes a suitable makeshift. Medoids also offer alternative solutions and proved useful to visualize insights [150]. A different approach is to segment a trajectory into common fragments, also called tracklets or pathlets, and then compare the segmentations. To find such pathlets, Lee et al. [164] interpolate sequences by polygonal functions and compare the line segments. Van Kreveld and Luo [243] propose to combine sub-trajectories of various durations directly. Extracting a sub-trajectory is equivalent to shifting the indexation (or the time component) for early termination. Buchin et al. [51] provides an extensive study of this approach.

## 6.2 Preliminaries

In the following, we rely on the notation and definitions (dissimilarity measures and point at infinity) given in Chapter 2. Nevertheless, recall that the set of all the possible observations is noted $\Omega$. It can be countable (e.g., nodes of a graph) or uncountable (e.g., real vector space).

### 6.2.1 Trajectories

In the following, $d$ is a metric on $\Omega$ without infinity point, while a generic trajectory dissimilarity is noted $D$. To come up with a consistent and general definition of a trajectory, we distinguish the following cases: If $\Omega$ is countable, it is the set of the nodes of an undirected graph of finite cardinality $\#(\Omega)$, and $d$ is the shortest path distance within this graph. If, on the other hand, $\Omega$ is uncountable, we suppose $\Omega = \mathbb{R}^p$ and $d$ boils down to the Euclidean distance. Extending $(\Omega, d)$ with a point at infinity, denoted by $\bar{\Omega} = \Omega \cup \{\infty\}$, leads, in the Euclidean case, to the addition of the infinite point $(+\infty, \cdots, +\infty)$. In the discrete case, it corresponds to the addition of an extra node with connected to all the others with infinite weight.

**Definition 6.1** (Trajectory)**.** *A* trajectory $X$ *on* $\overline{\Omega}$ *is a function* $X : \mathbb{R}_{\geq 0} \to \overline{\Omega}$*. If* $\Omega$ *is countable or uncountable,* $X$ *realizes a step or polygonal function, respectively. The symbol* $X$ *refers to the function or its graph* $\{(t, X(t)), t \in \mathbb{R}_{\geq 0}^{+\infty}\}$*. The elements of the domain are called* timestamps*. A trajectory is of* duration $T_X \in \mathbb{R}_{\geq 0}$ *iff*

$$\forall t > T_X,\ X(t) = \infty \quad \text{and} \quad X(T_X) \neq \infty.$$

*The* prefix *of* $X$ *of duration* $s \geq 0$*, written as* $X_{:s}$*, is given by*

$$\forall t \in \mathbb{R}_{\geq 0},\ X_{:s}(t) = \begin{cases} X(t), & \text{if } 0 \leq t \leq s \\ \infty & \text{otherwise.} \end{cases}$$

*The* suffix *of* $X$ *starting at* $s \geq 0$*, denoted by* $X_{s:}$*, is defined as*

$$\forall t \in \mathbb{R}_{\geq 0},\ X_{s:}(t) = X(t + s).$$

*A trajectory is* connected *iff it is of finite duration and*

$$\forall t \in [0\,, T_X],\ X(t) \neq \infty.$$

*In this case, the interval* $[0\,, T_X]$ *is called the* duration interval *of* $X$*. The set of the connected trajectories on* $\overline{\Omega}$ *is denoted* $\mathcal{T}(\Omega)$*.*

We can now define the time-wise average trajectory of a set of trajectories: the position at each timestamp is the average position of that of the active trajectories.

**Definition 6.2** (Average Trajectory)**.** *The* active set *of a set* $\mathcal{X} = \{X_1, \cdots, X_n\}$ *of* $n \in \mathbb{N}$ *trajectories at a time* $t \in \mathbb{R}_{\geq 0}$ *is the set*

$$A_{\mathcal{X}}(t) = \{i \in [1 \,..\, n],\ X_i(t) \neq \infty\}.$$

*The* average trajectory $\overline{\mathcal{X}}$ *of* $\mathcal{X}$ *is then given by*

$$\overline{\mathcal{X}}(t) = \begin{cases} \dfrac{1}{\#A_{\mathcal{X}}(t)} \sum_{i \in A_{\mathcal{X}}(t)} X_i(t), & \text{if } A_{\mathcal{X}}(t) \neq \emptyset, \\ \infty & \text{otherwise.} \end{cases}$$

Note that the average trajectory may be discontinuous, however, it is always represented as a continuous curve. In practice, trajectories arise from a finite set of measurements ordered by timestamps. These sequences play the role of the interface between the theoretical trajectories and the real world.

**Definition 6.3** (Temporal Sequence)**.** *A sequence* $\mathbf{x} = \langle (t_i, x_i) \rangle_N$ *of* $\mathbb{R}_{\geq 0} \times \Omega$*, is called a* temporal sequence *iff:*

$$t_1 = 0, \quad x_N \neq \infty \quad \text{and} \quad \forall (i,j) \in \mathbb{N}^2, \quad i < j \Leftrightarrow t_i < t_j.$$

*The set of the temporal sequences on* $\Omega$ *is denoted* $\mathcal{S}(\Omega)$*.*

We use the term of temporal sequence instead of time-series to emphasize the distinction between indices and timestamps. There are an infinite number of functions that interpolate chronologically the positions of a temporal sequences. However, only one is also a connected trajectory of duration $t_N$ in the sense of Definition 6.1. The *connected* property is here essential. Indeed, while trajectories are defined on $\mathbb{R}_{\geq 0}$, temporal sequences are finite. Therefore, we are interested in the trajectory that is infinite, i.e., ends, with the last event sequence.

**Proposition 6.1.** *For any temporal sequence* $\langle t_i, x_i \rangle_N$*, there exists a unique connected trajectory of duration* $t_N$ *that is affine on each interval* $[t_i \,, t_{i+1})$ *for* $1 \leq i \leq N$ *and whose graph passes chronologically through the elements of the sequence.*

*Proof.* If two trajectories satisfy these conditions, they are equal on all timestamps $t_i$. Let us show now that they are equal on $[0 \,, t_N]$ by showing that they match on every interval $[t_i \,, t_{i+1})$. We differentiate countable and uncountable cases: (i) If $\Omega$ is countable, the trajectories are step functions that are constant on the intervals

$[t_i\,, t_{i+1})$. Since both take the same value on $t_i$, they are in fact equal in each interval. (ii) If $\Omega$ is uncountable, the trajectories are continuous on their duration interval, and hence in each $[t_i\,, t_{i+1}]$. Given the proposition's statement, the trajectories are affine in the semi-open intervals and match in both extremes. Thus, in each interval $[t_i\,, t_i + 1]$, they both match the affine function interpolating $(t_i, x_i)$ and $(t_{i+1}, x_{i+1})$. Finally, being connected and of same duration, they are equal everywhere. □

Note that the relation between a sequence and its interpolated trajectory is not a one-to-one relation. If $\Omega = \mathbb{R}^p$, adding the average position between any two successive points results in an unchanged interpolation. This observation gives rise to an equivalence relation between sequences.

**Definition 6.4** (Interpolation)**.** *A trajectory $X$* interpolates *a temporal sequence $\mathbf{x}$ if the conditions of Proposition 6.1 are satisfied.*

*Two temporal sequences $\mathbf{x}$ and $\mathbf{y}$ are* equivalent*, noted $\mathbf{x} \sim \mathbf{y}$, iff they are* interpolated *by the same trajectory.*

*The set of all the sequences interpolated by the same trajectory $X$ is denoted $[X]$.*

**Proposition 6.2.** *The relation $\sim$ is an equivalence relation and $[X]$ is an equivalence class.*

*Proof.* Straightforward. □

The trajectory interpolating sequences of the same class does not belong to that class itself, but to the closure of it. That is, the trajectory can be defined as the limit of a sequence of temporal sequences of the equivalence class.

**Lemma 6.1.** *The graph of the trajectory $X$ restricted to its duration interval is the only element in the closure of $[X]$, $\mathrm{Cl}([X])$, that is not in $[X]$.*

*Proof.* Sketch of proof: First, we show that a trajectory made of a single segment is the limit of a temporal sequence; the same reasoning generalizes to trajectories with more segments. Then, we use the fact that an interval of $\mathbb{R}$ is the limit of a sequence of countable sets of points. This is can be deduced from $\mathbb{Q}$ being dense in $\mathbb{R}$ [43]. The uniqueness of $X$ is thus guaranteed by Definition 6.4. □

By equipping $[X]$ with a partial order, we can also show that each equivalence class has a *smallest* sequence with respect to the number of points. The trajectory $X$ is a good candidate to be the greatest, if the order is extended to the closure of $[X]$.

**Proposition 6.3.** *Each equivalence class* $[X]$ *is equipped with a partially order,* $\preceq$*:*

$$\langle t, x\rangle_N \preceq \langle \tau, y\rangle_M \;\Leftrightarrow\; \{(t_i, x_i), i \in [1\,..\,N]\} \subseteq \{(\tau_j, y_j), j \in [1\,..\,M]\}.$$

*i. The relation* $\preceq$ *has a minimal element on* $[X]$*: The time sequence containing only the extreme points of the segments of the graph of* $X$ *and* $(T_X, X(T_X))$*.*

*ii. The graph of* $X$ *is the maximal element of that relation on* $\mathrm{Cl}([X])$

*Proof.* (i) is a direct consequence of Proposition 6.1. The function $X$ is either a step or a polygonal function in its duration interval. Therefore, it can be fully reconstructed by interpolating linearly and chronologically the vertices of its graph. If $\Omega$ is countable, the segments of $X$ on its duration interval have only one extremity, except for the last one that might also contain $(T_X, X(T_X))$.

(ii) is a consequence of Definition 6.4 and Lemma 6.1. □

## 6.3 Trajectory Measures

The general axioms of Definition 2.1 allow for a great deal of possible trajectory measures. In the following, we introduce a new classification scheme of existing measures and discuss our approach on the example of prominent similarity measures for trajectories in Section 6.4.

### 6.3.1 Point and Path-measures

#### Point-Measures

Dubuisson and Jain [87] state a generic recipe to build distance measures for trajectories. Their approach scans the first trajectory and computes for every timestamp the distance to the second trajectory. The same operation is repeated after swapping the trajectories. Finally, a decision is taken to say which value to return. Their procedure does not to describe the way the second trajectory is browsed.

**Definition 6.5** (Point-Measure)**.** *A* point-measure $D$ *is a measure on* $\mathcal{T}(\Omega)$ *defined by a symmetric bivariate real function* $\mathcal{S}ym$*, two operators* $\mathcal{O}^1$*,* $\mathcal{O}^2$*, and a function* $\mathcal{D}$*, such that:*

$$D(X, Y) = \mathcal{S}ym\Big(\mathcal{O}^1 \circ \mathcal{O}^2 \circ \mathcal{D}(X; Y)\ ,\ \mathcal{O}^1 \circ \mathcal{O}^2 \circ \mathcal{D}(Y; X)\Big). \tag{6.1}$$

*The arguments* $X$ *and* $Y$ *correspond to the graphs of the trajectories. The operators* $\mathcal{O}^1$ *and* $\mathcal{O}^2$ *act, respectively, on the timestamps of the first and second argument.*

*Therefore, the composed function is not necessarily symmetric, which is stressed by the semi-colon.*

Usual values for $\mathcal{S}ym$ are max, min, or the average function $(a, b) \mapsto \frac{a+b}{2}$. Regarding the operators $\mathcal{O}^1$ and $\mathcal{O}^2$, they can be for instance $\sup_t$, $\inf_t$, $\int_t$, or again the average function $\frac{1}{T}\int_t$. Although the definition targets measures between trajectory functions, the operators can be adapted for the temporal sequences. As an example, we explicit the decomposition of the Hausdorff *distance*:

$$\mathcal{S}ym = \max,\ \mathcal{O}^1 = \sup_t,\ \mathcal{O}^2 = \inf_t,$$

$$\text{Hausdorff}(X, Y) = \max \begin{pmatrix} \sup_{t\in[0\,,T_X]} \inf_{\tau\in[0\,,T_Y]} d\Big(X(t), Y(\tau)\Big), \\ \sup_{\tau\in[0\,,T_Y]} \inf_{t\in[0\,,T_X]} d\Big(Y(\tau), X(t)\Big) \end{pmatrix}.$$

**Path-Measures**

The second class of measures acts on the time component. These measures transforms time to give both trajectories the same duration and minimize a certain cost function. The transformations are non-decreasing surjective functions from the same interval (usually $[0\,, L] \subset \mathbb{R}_{\geq 0}$) into the duration intervals of the trajectories. They are often called *warping* functions and compute a sequential *alignment*. Geometrically, the cartesian product of the warping functions draws a *path* of length $L$ connecting $(0, 0)$ and $(T_X, T_Y)$

**Definition 6.6** (Path-Measure)**.** *A* path-measure $D$ *is a measure on* $\mathcal{T}(\Omega)$ *defined by an operator* $\mathcal{O}$ *and a function* cost*, such that*

$$D(X, Y) = \inf_{\alpha,\beta} \Big(\mathcal{O}_{l\in[0\,,L]} \circ \text{cost}(X \circ \alpha, Y \circ \beta)\Big), \tag{6.2}$$

*where* $X$ *and* $Y$ *are graphs of trajectories, and* $\alpha$ *and* $\beta$ *are two non-decreasing surjective functions of* $[0\,, L]$ *into* $[0\,, T_X]$ *and* $[0\,, T_Y]$*, respectively.*

Path-measures are usually defined on the temporal sequences to have a finite set of warping functions. Consequently, the warping functions act on the indices: $(\alpha, \beta) : [0\,..\,L]^2 \to [0\,..\,N] \times [0\,..\,M]$. For this, $\Omega$ is extended with a special point $g$ representing a gap in time. All sequences are prepended with the special point at the index 0, and also the warping functions are extended to $\alpha(0) = \beta(0) = 0$.

Using $\mathcal{O}_l = \sum_{l=1}^{L}$ and costs for deletions $\text{cost}(g, b)$, insertions $\text{cost}(a, g)$, and substitutions $\text{cost}(a, b)$, we obtain the class of edit-distances [62], e.g.,

$$\text{EDR}(\langle t, x\rangle_N, \langle \tau, y\rangle_M) = \inf_{\alpha,\beta} \sum_{l=1}^{L} \text{cost} \begin{pmatrix} g + (x_{\alpha(l)} - g)(\alpha(l) - \alpha(l-1)), \\ g + (y_{\beta(l)} - g)(\beta(l) - \beta(l-1)) \end{pmatrix},$$

where $\text{cost}(g, \cdot) = \text{cost}(\cdot, g) = 1$ and $\text{cost}(u, v) = 1 - \delta_{uv}$ for $u, v \in \Omega \setminus \{g\}$.

### 6.3.2 Conformal Measures

Definition 2.1 is not sufficient to capture the relationship that temporal sequences have with their interpolating trajectory. In practice, we do not have access to the trajectory function, but to a time sequence from its equivalence class. Hence, it is desirable that a measure does not depend on the class's representative. For example, this ensures that points can be added to the sequence to facilitate storage or calculations.

**Definition 6.7** (Well-Defined)**.** *A measure $D$ on $\mathcal{S}(\Omega)$ is called* well-defined *if it induces a dissimilarity $D^*$ on $\mathcal{S}(\Omega)/\!\sim$ such that:*

$$\forall([\mathbf{x}], [\mathbf{y}]) \in (\mathcal{S}(\Omega)/\!\sim)^2,\ D^*([\mathbf{x}], [\mathbf{y}]) = D(\mathbf{x}, \mathbf{y})$$

Although, the definition seems to focus on $D^*$, it has several implications on $D$ as shown in the following proposition.

**Proposition 6.4.** *A* well-defined *measure $D$ on $\mathcal{S}(\Omega)$ satisfies the following properties:*

1. *It is class invariant under $\sim$, i.e., for any three time sequences $\mathbf{x}$, $\mathbf{x}'$ and $\mathbf{y}$:*

$$\mathbf{x} \sim \mathbf{x}' \Rightarrow D(\mathbf{x}, \mathbf{y}) = D(\mathbf{x}', \mathbf{y}).$$

2. *It does not satisfy the identity of the indiscernibles on $\mathcal{S}(\Omega)$.*

3. *It can be extended to $\mathcal{T}(\Omega)$, such that*

$$\forall(X, Y) \in \mathcal{T}(\Omega)^2,\ D(X, Y) = D^*([X], [Y]).$$

*Proof.* 1. By symmetry, it is sufficient to prove the property only for one argument of $D$. If $\mathbf{x}$ and $\mathbf{x}'$ are two equivalent temporal sequences, $[\mathbf{x}] = [\mathbf{x}']$ and

$$D(\mathbf{x}, \mathbf{y}) = D^*([\mathbf{x}], [\mathbf{y}]) = D^*([\mathbf{x}'], [\mathbf{y}]) = D(\mathbf{x}', \mathbf{y}).$$

2. Let $\mathbf{x}$ and $\mathbf{x}'$ be two sequences such that $\mathbf{x} \sim \mathbf{x}'$ but $\mathbf{x} \neq \mathbf{x}'$, given that $D^*$ is a dissimilarity on $\mathcal{S}(\Omega)/\sim$, $D^*([\mathbf{x}], [\mathbf{x}']) = 0$. Hence, $D(\mathbf{x}, \mathbf{x}') = 0$, which contradicts the identity of the indiscernibles as the two sequences are different.
3. The trajectory $X$ is an element of $\mathrm{Cl}([X])$. There exists a sequence of temporal sequences $(\mathbf{x}_i)_{\mathbb{N}}$ of $[X] \subset \mathcal{S}(\Omega)$ that have $X$ as a limit. $D$ is invariant within a class equivalence and for any given temporal sequence $\mathbf{y} \in \mathcal{S}(\Omega)$, $D(\mathbf{x}_i, \mathbf{y})$ is constant and equal to $D^*([X], [\mathbf{y}])$, for any $i \in \mathbb{N}$. It holds

$$\lim_{i \to +\infty} D(\mathbf{x}_i, \mathbf{y}) = \lim_{i \to +\infty} D^*([X], [\mathbf{y}]) = D^*([X], [\mathbf{y}]).$$

The same reasoning can be applied to the second argument, concluding the proof. □

Timestamps have been so far expressed without unit. In practice, measurements may come from devices with different time units such as seconds or milliseconds. In such a situation, the measure is expected to return the same value independently of the time scale.

**Definition 6.8** (Time Scale-Invariance)**.** *A measure $D$ on $\mathcal{S}(\Omega)$ is called* time scale-invariant*, if it is invariant under a positive scaling of the timestamps, i.e., for any $\rho \in \mathbb{R}_+$,*

$$\forall \Big( \langle t, x \rangle_N, \langle \tau, y \rangle_M \Big) \in \mathcal{S}(\Omega)^2,\ D\Big( \langle t, x \rangle_N, \langle \tau, y \rangle_M \Big) = D\Big( \langle \rho t, x \rangle_N, \langle \rho \tau, y \rangle_M \Big).$$

The time efficiency of a measure can be crucial, especially when dealing with many long trajectories. As we shall see (Section 6.4), many of the existing measures are restricted in their use due to their quadratic computational cost.

**Definition 6.9** (Efficient)**.** *A measure $D$ on $\mathcal{S}(\Omega)$ is said* efficient *if it can be computed in linear time with respect to the length of the sequences.*

The final definition in this section names the class of measures satisfying all three definitions.

**Definition 6.10** (Conformity)**.** *A* conformal *measure on $\mathcal{S}(\Omega)$ is* well-defined*,* time scale-invariant *and* efficient*.*

## 6.4 Classification of Existing Measures

In this section we review existing trajectory measures with respect to the classification scheme developed in the previous section. In the following definitions, $\mathbf{x} = \langle t_i, x_i \rangle_N$ and $\mathbf{y} = \langle \tau_j, y_j \rangle_M$ are two temporal sequences of length $N$ and $M$. They are interpolated, respectively, by the connected trajectories $X$ and $Y$ of duration $T_X = t_N$ and $T_Y = \tau_M$. If nothing else is stated, we suppose that $T_X < T_Y$ and $N < M$.

Table 6.1: Summary of point-measures, with respect to the formalization of Definition 6.5. Relevant parameters are shown in brackets, differentials are omitted.

| Measure | $\mathcal{S}ym$ | $\mathcal{O}^1 \circ \mathcal{O}^2 \circ \mathcal{D}(X;Y)$ | Discr. |
|---|---|---|---|
| Hausdorff | max | $\sup_{t\in[0\,,T_X]} \inf_{\tau\in[0\,,T_Y]} d\Big(X(t),Y(\tau)\Big)$ | - |
| OneWay | $a,b \mapsto \frac{a+b}{2}$ | $\frac{1}{T_X}\int_{t=0}^{T_X} \inf_{\tau\in[0\,,T_Y]} d\Big(X(t),Y(\tau)\Big)$ | SSPD |
| LCSS* | min | $\frac{1}{T_X}\int_{t=0}^{T_X} \inf_{\tau\in[0\,,T_Y]} \varphi\Big(T_X\Big)\begin{pmatrix}1-\\ \end{pmatrix}\begin{matrix}\mathbb{1}_{\omega_T}\Big(\lvert t-\tau\rvert\Big)\times\\ \mathbb{1}_{\omega_D}\Big(d\Big(x_t,y_\tau\Big)\Big)\end{matrix}\Bigg)$ | discrete LCSS |
| Euclid. | min | $\int_{t=0}^{T_X} \max_{\tau\in[0\,,T_Y]} \delta_{t\tau} d\Big(X(t),Y(\tau)\Big)$ | DISSIM |
| av.Euclid. | min | $\frac{1}{T_X}\int_{t=0}^{T_X} \max_{\tau\in[0\,,T_Y]} \delta_{t\tau} d\Big(X(t),Y(\tau)\Big)$ | $\Delta$ |

* $\varphi(t) = 1 + \frac{t-\min(T_X,T_Y)}{t-\max(T_X,T_Y)}$.

### 6.4.1 Point-Measures

There are five main approaches that fit to the definition of a point-measure. Table 6.1 summarizes them according to Definition 6.5. Equations assume continuous trajectories; the last column indicates, when necessary, the discretization that we use. We now review each measure, provide formulas for temporal sequences, and discuss their *conformity*.

**The Hausdorff distance** quantifies the difference between sets of points [123]. It is commonly used in computer vision for object detection [132]. It is also used in the analysis of video surveillance to cluster trajectories and detect anomalies [140, 195].

**Definition 6.11** ([123])**.** *The* Hausdorff *distance between two temporal sequences is given by:*

$$\text{Hausdorff}(\mathbf{x},\mathbf{y}) = \max\left(\sup_i\left(\inf_j\Big(x_i,y_j\Big)\right),\sup_j\left(\inf_i d\Big(x_i,y_j\Big)\right)\right).$$

Although, the measure is a mathematical distance [123] on the power set of $\Omega$, it does not induce a dissimilarity on $\mathcal{S}(\Omega)/\sim$. The example of Figure 6.1 shows that the measure is not constant on an equivalence class (not well-defined). Since the timestamps are not used, the measure is time scale-invariant. The computations requires the comparison of all the possible pairs, hence a time complexity of $O(NM)$ (not efficient).

**The One-Way distance** introduced in [171] sums the minimum distances between points of the first trajectory and the full graph of the second one, and vice-versa. The average of the two sums is then returned. In practice, computing the distance from a point to a trajectory is costly. The authors proposed OWDgrid: The space is divided into grid-cells, such that the cell containing the point is compared with the closest cell intersecting the second trajectory. The *Symmetrized Segment-Path Distance* (SSPD) introduced in [32] takes advantage of the polygonal approximation and computes the distance from a point to the segments of the other trajectory using orthogonal projections.

**Definition 6.12** ([32])**.** *The* SSPD *between two trajectories sequences is given by*

$$\mathrm{SSPD}(\mathbf{x},\mathbf{y}) = \frac{s(\mathbf{x};\mathbf{y}) + s(\mathbf{y};\mathbf{x})}{2},$$

*where* $s(\mathbf{x};\mathbf{y})$ *is given by*

$$s(\mathbf{x};\mathbf{y}) = \frac{1}{N+1}\sum_{i=1}^{N} \min_{j\in[1..M-1]} \Big(d_{ps}\big(x_i, [y_j\,, y_{j+1}]\big)\Big),$$

*and* $d_{ps}$ *by*

$$d_{ps}\big(x, [a\,, b]\big) = \min_{u\in[0,1]} \Big(d\big(x, au + (1-u)b\big)\Big).$$

The example of Figure 6.1 highlights a case where the measure is not constant on an equivalence class; hence, it is not well-defined. The timestamps are not used in the formula, thus, the measure is time scale-invariant. With a quadratic time complexity of $O(NM)$, SSPD is also not an efficient measure.

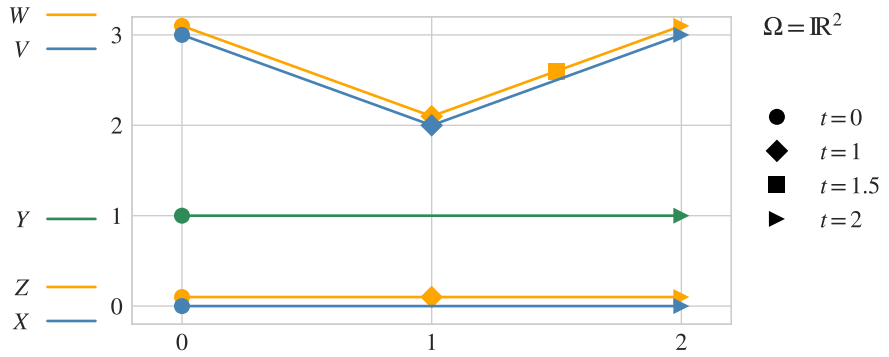
**The Longest Common SubSequence** (LCSS) is supposed to overcome limitations of the edit-distance [166, 62, 61]. The measure that we present here corresponds to $D1$ in [246]. Besides, we give the original formula for temporal sequences. For the continuous version see Table 6.1.

**Definition 6.13** ([246])**.** *Let* $\omega_T$ *and* $\omega_D$ *be a non-negative integer and a positive real number, respectively. The* LCSS *measure between two trajectories is given by*

$$\mathrm{LCSS}(\mathbf{x},\mathbf{y} : \omega_T, \omega_D) = 1 - \frac{l_{\omega_T,\omega_D}(\langle x\rangle_N, \langle y\rangle_N)}{\min(N, M)}$$

*where* $\ell \equiv l_{\omega_T,\omega_D}\Big(\langle x\rangle_N, \langle y\rangle_M\Big)$ *equals*

$$\begin{cases} 0 & \text{if } N = 0 \text{ or } M = 0, \\ 1 + \ell\Big(\langle x\rangle_{N-1}, \langle y\rangle_{M-1}\Big) & \text{if } d(x_N, y_M) \le \omega_D \\ & \quad \text{and } |N-M| \le \omega_T, \\ \max\Big(\ell\big(\langle x\rangle_N, \langle y\rangle_{M-1}\big), \ell\big(\langle x\rangle_{N-1}, \langle y\rangle_M\big)\Big) & \text{otherwise.} \end{cases}$$




| | Fréchet | DTW | Hausdorff | SSPD | $\text{LCSS}_{.5,1}$ | DISSIM | $\Delta$ |
|---|---|---|---|---|---|---|---|
| $(X, Y)$ | 1.0 | 2.0 | 1.0 | 1.0 | 0.0 | 2.0 | 1.0 |
| $(Z, Y)$ | $\sqrt{2}$ | $\approx 3.41$ | $\sqrt{2}$ | 1.0 | 0.5 | 2.0 | 1.0 |
| $(V, Y)$ | 2.0 | $\approx 5.41$ | 2.0 | $\approx 1.54$ | 1.0 | 3.0 | 1.5 |
| $(W, Y)$ | 2.0 | $\approx 6.99$ | 2.0 | $\approx 1.52$ | 1.0 | 3.0 | 1.5 |

Figure 6.1: The marks indicate the timestamps. Equivalent trajectories sequences are represented by the trajectory. To distinguish them, they are slightly translated. The table reports the dissimilarities between some trajectories and $Y$ for each baseline measure. Measures invariant under $\sim$ have their two first and two last lines of the table equal.

Despite a recursive definition, LCSS is not a path-measure: Some indices might be skipped, which prevents the warping functions to be surjective. However, it can be expressed as a point-measure (Table 6.1). The measure quantifies the ratio of points of the smallest trajectory that have a point of the other trajectory within a spatial and temporal window parameterized by $\omega_D$ and $\omega_T$, respectively. The combination of $\mathcal{S}ym = \min$ and the function $\varphi$ ensures that the returned value corresponds to a sum over the shortest trajectory. In the desired case $\varphi$ equals 1. It is infinite, otherwise.

Figure 6.1 gives an example where LCSS fails to be well-defined. The measure is time scale-invariant, since the timestamps are not used. The time complexity of its computation is $O(\omega_T(N + M))$, what is considered as efficient.

**The Euclidean distance** is a standard baseline [258, 246, 159, 62, 187, 199, 33]. Although the name is used abundantly, it is not always the same function behind it. It can, for instance, be discrete (the sum of the distances between position with the same index) or continuous (the area spanned between the two curves). We focus on the continuous case as the former omit the time component. The function measuring the area is piece-wise hyperbolic [243] which renders an exact computation

expensive [100]. Frentzos et al. [101] introduce DISSIM as an approximation using the trapezoid rule.

**Definition 6.14** ([101])**.** *Let* $\langle t_l \rangle_L$ *be the sequence of timestamps of* $\mathbf{x}$ *and* $\mathbf{y}$ *that are smaller or equal than* $\min(T_X, T_Y)$*, the* DISSIM *measure between these two trajectories sequences is given by:*

$$\text{DISSIM}(\mathbf{x}, \mathbf{y}) = \sum_{l=1}^{L-1} \frac{d\Big(x_l, y_l\Big) + d\Big(x_{l+1}, y_{l+1}\Big)}{2} \Big(t_{l+1} - t_l\Big),$$

*where* $x_l = X(t_l)$ *and* $y_l = Y(t_l)$*.*

However, the DISSIM measure between a trajectory and a prefix is null. This contradicts, the previous implication and DISSIM turns out to be not well-defined. It is also not time scale-invariant since

$$\forall \rho \neq 0,\ \text{DISSIM}(\langle \rho t, x \rangle_N, \langle \rho \tau, y \rangle_M) = \rho\, \text{DISSIM}(\langle t, x \rangle_N, \langle \tau, y \rangle_M).$$

However, this approximation of the Euclidean distance is efficient and can be computed in linear time by browsing chronologically the sequences of timestamps $\langle t_l \rangle_L$.

**The average Euclidean distance** was redefined several times, e.g., for similarity search of sequences [165], for pattern extraction [105], to cluster trajectories of hurricanes [86], and recently to group and analyze online behavior in educational contexts [40]. All but the latter assume a regular sampling of the position. Similarly to DISSIM, we will approximate the continuous formula by the sum of the areas of trapezoids, and refer to it as $\Delta$.

**Definition 6.15** ([200])**.** *Let* $\langle t_l \rangle_L$ *be the sequence of timestamps of* $\mathbf{x}$ *and* $\mathbf{y}$ *that are smaller or equal than* $\min(T_X, T_Y)$*, the* $\Delta$ *measure between these two trajectories sequences is given by:*

$$\Delta(\mathbf{x}, \mathbf{y}) = \frac{1}{\min(T_X, T_Y)} \sum_{l=1}^{L-1} \frac{d\Big(x_l, y_l\Big) + d\Big(x_{l+1}, y_{l+1}\Big)}{2} \Big(t_{l+1} - t_l\Big),$$

*where* $x_l = X(t_l)$ *and* $y_l = Y(t_l)$*.*

The definition echoes that of DISSIM, except for the time normalization. Moreover, $\Delta$ is also not well-defined, as it does not distinguish a trajectory from its prefixes. On the other hand, the time normalization ensures the time scale-invariance. The same algorithm can be used to compute simultaneously DISSIM and $\Delta$ in linear time (efficient).

Table 6.2: Summary of path-measures, with respect to the formalization of Definition 6.6. Parameters, if relevant, are in brackets, $(x_l, y_l) = (x_{\alpha(l)}, y_{\beta(l)})$, and $(\alpha_l, \beta_l) = (\alpha(l), \beta(l))$.

| | $\mathcal{O}_l \circ \text{cost}(\langle t, x\rangle_N, \langle \tau, y\rangle_M)$ |
|---|---|
| Fréchet | $\max_{1\leq l\leq L} d(x_l, y_l)$ |
| DTW | $\sum_{l=1}^{L} d(x_l, y_l)$ |
| EDR($\omega_\text{D}$) | $\sum_{l=1}^{L} \min\Big(1, (\alpha_l - \alpha_{l-1}) + (\beta_l - \beta_{l-1}) + \big(1 - \mathbb{1}_{\omega_D} d(x_l, y_l)\big)\Big)$ |
| ERP | $\sum_{l=1}^{L} d\Big(g + (x_l - g)(\alpha_l - \alpha_{l-1}), g + (y_l - g)(\beta_l - \beta_{l-1})\Big)$ |

### 6.4.2 Path-Measures

As noted in Section 6.3.1, path-measures are usually defined for temporal sequences only. Table 6.2 gives the decomposition of four measures: Fréchet [99], DTW [213], EDR [62], and ERP [61]. Given their similarity and their expensive computations, we review only the former two.

**The Fréchet distance** is a measure between continuous functions [99]. In practice, the discrete version is often preferred [7] and has been successfully used in trajectory analysis [50, 180, 251]. The difference between the discrete Fréchet and the Hausdorff distance is that the former respects the ordering of the points. The non-decreasing constraint on the alignments prevents indices from appearing in two non-contiguous sub-sequences.

**Definition 6.16** ([92])**.** *The discrete* Fréchet *measure between two temporal sequences is given by*

$$\text{Fréchet}(\mathbf{x}, \mathbf{y}) = \inf_{\alpha,\beta} \left( \max_{l\in[1..L]} \left( d\big(x_{\alpha(l)}, y_{\beta(l)}\big) \right) \right),$$

*where $\alpha$ and $\beta$ are two non-decreasing surjective functions of $[1\,..\,L]$ into $[1\,..\,N]$ and $[1\,..\,M]$, respectively, with $L \geq \max(N, M)$.*

The example of Figure 6.1 shows that Fréchet is not well-defined: its value differs for two equivalent trajectories. Since timestamps are ignored, the measure is time scale-invariant. The computations of the discrete Fréchet distances are, however, not efficient and requires $O(NM)$ [92] operations. Agarwal et al. [2] proposed a faster algorithm with a time complexity of $O\left(NM \log\log N / \log N\right)$.

Table 6.3: Summary of the properties satisfied by the baselines.

| | Well-defined | Time scale-invariant | Efficient | Complexity |
|---|---|---|---|---|
| Fréchet | - | ✓ | - | $O(NM)$ |
| DTW | - | ✓ | - | $O(NM)$ |
| Hausdorff | - | ✓ | - | $O(NM)$ |
| SSPD | - | ✓ | - | $O(NM)$ |
| LCSS | - | ✓ | $\approx$ | $O(\omega_T(N+M))$ |
| DISSIM | - | - | ✓ | $O(N+M)$ |
| $\Delta$ | - | ✓ | ✓ | $O(N+M)$ |

**Dynamic Time Warping** (DTW) was first introduced [213] for speech detection. Its successes made it a candidate for many other tasks ranging from signal processing [31] to analyzing human behavior [267]. Unlike Fréchet, the measure returns the sum of the difference instead of a maximum.

**Definition 6.17.** *The* DTW *measure for two trajectories is given by*

$$\mathrm{DTW}(\mathbf{x}, \mathbf{y}) = \inf_{\alpha,\beta} \left( \sum_{l=1}^{L} \left( d\big(x_{\alpha(l)}, y_{\beta(l)}\big) \right) \right),$$

*where $\alpha$ and $\beta$ are two non-decreasing surjective function of $[1\,..\,L]$ into $[1\,..\,N]$ and $[1\,..\,M]$, respectively, with $L \geq \max(N, M)$.*

DTW is not well-defined as subdividing a segment might increase the returned value (Figure 6.1). The measure does not use the timestamps and is thus time scale-invariant. Finally, with a time complexity of $O(NM)$, DTW is not efficient according to our definition. Approximations, such as fastDTW [223], have been developed to bring the complexity to an almost linear asymptote. However, this gain in computation time comes at the cost of a lower precision.

In Table 6.3, we summarize the categorization by focusing on properties of a conformal measure. No measure satisfies all the required properties. Except for the Eculidean-based measures, timestamps are usually not used. Hence, most of the measures are time scale-invariant. The closest to a conformal measure is $\Delta$ that only lacks the well-defined property.

## 6.5 A Conformal Point-Measure

Point-measures estimate the closeness of two trajectories. To devise a conformal point-measure, we change the paradigm. We suppose that a trajectory is a real-

ization of a distribution, such that comparing trajectories boils down to comparing distributions.

### 6.5.1 A Probabilistic Approach

The approach is based on a generalization of the Laplace distribution.

**Proposition 6.5.** *Given a trajectory $X$ and a positive real number $\lambda$, there exist a real number $Z(\lambda)$ such that the following function, $p_X$, is a distribution on $\mathbb{R}_{\geq 0}\times\Omega$:*

$$\forall (t,x) \in \mathbb{R}_{\geq 0}\times\Omega,\ p_X\Big((t,x):\lambda\Big) = \frac{1}{Z(\lambda)T_X} e^{-\lambda d(X(t),x)}. \tag{6.3}$$

To ensure that $p_X$ gives rise to a distribution, the sum over its domain of definition must be 1. The parameter $\lambda$ needs to be strictly positive, otherwise the pdf is greater than 1 and the integral over the domain is no longer defined. Our proof relies on the following lemma.

**Lemma 6.2.** *Let $\lambda$ be a positive real number and $z$ be an element of $\overline{\Omega}$, then*

$$0 \leq \int_\Omega e^{-\lambda d(z,x)} \partial x < +\infty. \tag{6.4}$$

*The integral is null if, and only if, $z = \infty$.*

*Proof of Lemma 6.2.* According to Definition 2.3, if $z = \infty$, $\exp\Big(-\lambda d(z,x)\Big) = 0$ and the integral is null. Let us assume from now on that $z$ is not the point at infinity of $\overline{\Omega}$. If $\Omega$ is countable, it is assumed finite, and the integral can be bounded as follows:

$$1 \leq \int_\Omega e^{-\lambda d(z,x)} \partial x = \sum_{x\in\Omega} e^{-\lambda d(z,x)} \leq \#(\Omega) < +\infty.$$

The lower-bound implies that the integral is null iff $z = \infty$. Now, if $\Omega = \mathbb{R}^p$ and $d$ is the Euclidean distance, without loss of generality, we can assume that $z = \mathbf{0}_p$. The integral can be computed by using the $p$-spherical coordinates of $x \in \mathbb{R}^p$:

$$\begin{aligned} x_i &= r\sin(\varphi_1)\cdots\sin(\varphi_{i-1})\cos(\varphi_i), \quad \forall i \in [1\,..\,p-1], \\ x_p &= r\sin(\varphi_1)\cdots\sin(\varphi_{p-2})\sin(\varphi_{p-1}), \end{aligned}$$

where $r \in \mathbb{R}_{\geq 0}$, $\forall i \in [1\,..\,p-2]$, $\varphi_i \in [0,\pi]$, and $\varphi_{p-1} \in [0, 2\pi)$:

$$\int_{\mathbb{R}^p} e^{-\lambda d(z,x)} \partial x = \int e^{-\lambda r} \partial(r \times \varphi_1 \times \cdots \times \varphi_{p-1}) = \frac{2\pi^{p-1}}{\lambda} < +\infty,$$

where the domain of the last integral is the domain of the $p$-spherical coordinates. □

*Proof of Proposition 6.5.* Firstly, $p_X$ inherits the non-negativity of the exponential. To be a distribution, it remains to prove that there exists $Z(\lambda)$ that normalizes the sum of $p_X$:

$$\iint_{\mathbb{R}_{\geq 0}\times\Omega} p_X((t,x):\lambda)\,\partial t\partial x = \frac{1}{Z(\lambda)T_X}\int_0^{+\infty}\!\!\int_\Omega e^{-\lambda d(X(t),x)}\,\partial t\partial x$$

Given that the trajectory $X$ is supposed connected, if $t > T_X$, $X(t) = \infty$ and $e^{-\lambda d(X(t),x)} = 0$. The domain of the first integral can, hence, be reduced to $[0, T_X]$. On this interval, $X(t)$ is an element of $\Omega$. Lemma 6.2 implies that, since $\lambda > 0$, the sum over $\Omega$ is positive and finite. Setting $Z(\lambda)$ to the value of that sum renders $p_X$ a distribution. □

Remark that the relationship between a trajectory and the distribution it induces is goes ways. Indeed, a trajectory can also be defined as the most likely realization.

**Lemma 6.3.** *Let $X$ be a trajectory and $p_X$ the distribution it induces, let $x, t$ be a random variable on $\mathbb{R}_{\geq 0}\times\Omega$:*

$$(t,x) \sim p_X \Rightarrow \forall t \in \mathbb{R}_{\geq 0},\ \mathbb{E}(x|t) = X(t).$$

Thanks to Proposition 6.5, we can now compare trajectories by comparing the distributions they induce using the Kullback-Leibler (KL) divergence.

**Definition 6.18** (KL Divergence [157])**.** *The KL divergence between two trajectories $X$ and $Y$ is the KL divergence between the induced distributions. For a positive real number $\lambda$, it is given by:*

$$D_{KL}\Big(X||Y\Big) = -\iint_{\mathbb{R}_{\geq 0}\times\Omega} p_X\Big((t,x):\lambda\Big)\log\frac{p_Y\Big((t,x):\lambda\Big)}{p_X\Big((t,x):\lambda\Big)}\,\partial t\partial x. \tag{6.5}$$

At first sight, the KL divergence serves as a good candidate for comparing trajectories. It is non-negative and satisfies the axiom of identity of indiscernibles. However, it is not symmetric. Worse, depending on the order of the trajectories, it can be infinite.

**Proposition 6.6.** *Let $X$ and $Y$ be two trajectories, and $\lambda$ a positive real number, $D_{KL}$ satisfies the following properties:*

*(i) If $T_X > T_Y$, $D_{KL}(X||Y) = +\infty$,*

*(ii) If $T_X \leq T_Y$, $D_{KL}(X||Y) \leq \log\frac{T_Y}{T_X} + \frac{\lambda}{T_X}\int_0^{T_X} d\Big(X(t),Y(t)\Big)\partial t < +\infty$.*

*Proof.* The KL divergence can also be stated as the sum of an entropy and a cross-entropy. For legibility, we omit the arguments of $p_X$ and $p_Y$, the domain ($\mathbb{R}_{\geq 0}\times\Omega$) and the differential of the integrals. We obtain

$$D_{KL}\Big(X||Y\Big) = -\iint p_X \log p_Y + \iint p_X \log p_X = H(p_X, p_Y) - H(p_X)$$

Assuming that $T_X > T_Y$ and focus on $H(p_X, p_Y)$ yields

$$H(p_X, p_Y) = \frac{\lambda}{Z(\lambda)T_X} \iint d\Big(Y(t), x\Big) e^{-\lambda d(X(t),x)} + \text{constant}.$$

For $t \in (T_Y, T_X]$, $Y(t)$ is infinite but not $X(t)$. Hence, on this interval

$$d\Big(Y(t), x\Big) e^{-\lambda d(X(t),x)} = +\infty,$$

and $D_{KL}(X||Y) = +\infty$. By contrast, assuming $T_X \leq T_Y$ leads to

$$\begin{aligned}
D_{KL}\Big(X||Y\Big) &= -\iint p_X \log \frac{p_Y}{p_X} \\
&= -\iint p_X \log \frac{\frac{1}{(Z(\lambda)T_Y)}\exp(-\lambda d(Y(t), x))}{\frac{1}{(Z(\lambda)T_X)}\exp(-\lambda d(X(t), x))} \\
&= \log\frac{T_Y}{T_X}\iint p_X - \iint p_X \log\exp\Big(-\lambda\Big(d(Y(t), x) - d(X(t), x)\Big)\Big) \\
&= \log\frac{T_Y}{T_X} + \lambda\iint \Big(d(Y(t), x) - d(X(t), x)\Big)p_X \\
&\qquad\qquad\qquad\qquad\qquad\qquad (p_X \text{ is a distribution}) \\
&\leq \log\frac{T_Y}{T_X} + \lambda\iint \Big(d\Big(Y(t), X(t)\Big) + d(X(t), x) - d(X(t), x)\Big)p_X \\
&\qquad\qquad\qquad\qquad\qquad\qquad (\text{triangle inequality}) \\
&= \log\frac{T_Y}{T_X} + \frac{\lambda}{Z(\lambda)T_X}\iint d\Big(Y(t), X(t)\Big)e^{-\lambda d(X(t),x)}
\end{aligned}$$

The domain of the integration of $t$ can be split as follows:

- If $t > T_Y$, then $X(t) = Y(t) = \infty$ and $d\Big(Y(t), X(t)\Big)e^{-\lambda d(X(t),x)} = 0$.
- If $T_X < t \leq T_Y$, then $X(t) = \infty$ but $Y(t) \neq \infty$. However, given that the limit of the function $z \mapsto z\exp(-z)$ is 0, as $z$ approaches infinity, we still have $d\Big(Y(t), X(t)\Big)e^{-\lambda d(X(t),x)} = 0$.
- If $t < T_X$, none of the trajectories is infinite. The value of the double integral is deduced from Proposition 6.5:

$$\begin{aligned}
\iint d\Big(Y(t), X(t)\Big)e^{-\lambda d\Big(X(t),x\Big)} &= \int_0^{T_X} d\Big(Y(t), X(t)\Big)\int_\Omega e^{-\lambda d(X(t),x)}\partial x\,\partial t \\
&= Z(\lambda)\int_0^{T_X} d\Big(Y(t), X(t)\Big)dt < +\infty.
\end{aligned}$$

Hence, the KL divergence is finite. □

Although the KL-divergence is not symmetric, Proposition 6.6 suggests that we can use the upper-bound of the second statement as the basis for a new trajectory measure.

**Definition 6.19.** *The $\Delta_{\mathrm{KL}}$ measure between two trajectories, such that $T_Y > T_X$ and $\lambda \in [0\,,1]$, is given by:*

$$\Delta_{\mathrm{KL}}(X, Y : \lambda) = (1-\lambda)\log\frac{T_Y}{T_X} + \frac{\lambda}{T_X}\int_0^{T_X} d\Big(X(t), Y(t)\Big)\partial t. \tag{6.6}$$

*The integral is approximated using the trapezoid rule (see Definition 6.14).*

Unlike in Proposition 6.6, Equation 6.6 is a convex combination of the two terms. Such that both extreme cases are reachable : If $\lambda$ is null, the shape of the trajectories is ignored, and $\Delta_{\mathrm{KL}}$ only compares the durations. On the other hand, the measure equals the average Euclidean distance ($\Delta$) for $\lambda = 1$.

**Proposition 6.7.** *The measure $\Delta_{\mathrm{KL}}$ is a point-measure on $\mathcal{T}(\Omega)$ such that $\mathcal{S}ym =$ min and:*

$$\mathcal{O}^1 \circ \mathcal{O}^2 \circ \mathcal{D}(X;Y) = \frac{1}{T_X}\int_{t=0}^{T_X} \max_{\tau\in[0\,,T_Y]} \delta_{t\tau}\left((1-\lambda)\log\left(\frac{T_Y}{T_X}\right) + \lambda d\Big(X(t), Y(\tau)\Big)\right)\partial t. \tag{6.7}$$

*Proof.* The combination of $\mathcal{O}^2 = \max_{\tau\in[0\,,T_Y]}$ and $\delta_{t\tau}$ is a complicated way to impose $t = \tau$ in the sum. That way, Equations 6.6 and 6.7 are equivalent. Proposition 6.6 states that the integral is finite only when its domain is the shortest duration interval. With $\mathcal{S}ym = \min$, only the finite value is returned. The function $\Delta_{\mathrm{KL}}$ is hence symmetric. In addition, the KL divergence is non-negative and null if, and only if, the two compared distributions are equal. Thus, $\Delta_{\mathrm{KL}}$ is a dissimilarity on $\mathcal{T}(\Omega)$. □

**Proposition 6.8.** *The measure $\Delta_{\mathrm{KL}}$ induces a conformal trajectory measure on $\mathcal{S}(\Omega)$ if, and only if, $\lambda > 0$.*

*Proof.* The measure $\Delta_{\mathrm{KL}}$ for two trajectories sequences can be rewritten using $\Delta$ as

$$\Delta_{\mathrm{KL}}(\mathbf{x}, \mathbf{y} : \lambda) = (1-\lambda)\log\frac{\max(t_N, \tau_M)}{\min(t_N, \tau_M)} + \lambda\,\Delta(\mathbf{x}, \mathbf{y}).$$

If $\lambda = 0$, only the log of the ratios remains. In this case, $\Delta_{\mathrm{KL}}$ is symmetric and non-negative, since the ratio is always greater than 1 and null only if the trajectories have same duration.

For $\lambda > 0$, $\Delta_{\mathrm{KL}}$ acquires its *efficiency* from $\Delta$. It is *timescale-invariant* since $\Delta$ and the function $(\mathtt{x}, \mathtt{y}) \mapsto \log\frac{\max(t_N,\tau_M)}{\min(t_N,\tau_M)}$ share both the property. To show that $\Delta_{\mathrm{KL}}$

is *well-defined*, we need to show that it is invariant under $\sim$, and that $\Delta^*_{\mathrm{KL}}$ satisfies the identity of indiscernibles. The invariance on an equivalence class is satisfied by $\Delta$. In addition, two temporal sequences on the same class have necessarily the same duration, hence the ratio of duration is constant and $\Delta_{\mathrm{KL}}$ is invariant under $\sim$. $\Delta^*_{\mathrm{KL}}$ satisfies the identity of indiscernibles iff:

$$\Delta^*_{\mathrm{KL}}([\mathbf{x}],[\mathbf{y}]) = 0 \Leftrightarrow [\mathbf{x}] = [\mathbf{y}].$$

The invariance under $\sim$ of $\Delta_{\mathrm{KL}}$ gives the implication from right to left. To prove the other direction, let us consider two temporal sequences $\mathbf{x}$ and $\mathbf{y}$, such that

$$\Delta^*_{\mathrm{KL}}([\mathbf{x}],[\mathbf{y}]) = \Delta_{\mathrm{KL}}(\mathbf{x},\mathbf{y}) = 0.$$

This implies that the log of the ratios equals zero, i.e., the trajectories have same the duration. Besides, $\Delta(\mathbf{x},\mathbf{y}) = 0$ implies that the area spanned between the two trajectories interpolating each sequences is null. Therefore, the two temporal sequences are interpolated by the same trajectory: they are equivalent and $[\mathbf{x}] = [\mathbf{y}]$. □

### 6.5.2 Implementation

The measure $\Delta_{\mathrm{KL}}$ can be computed with $O(N+M)$ operations by simultaneously browsing the two trajectories, as described in Algorithm 10

The algorithm takes as input two temporal sequences $\langle t_i, x_i\rangle_N$ and $\langle \tau_j, y_j\rangle_M$. The timestamps of both sequences are processed chronologically until the smallest final timestamp, $\mathtt{mT} = \min(t_N, \tau_M)$. For each timestamp, the area between the current and the previous visited position is computed and added to `D`.

In the second and third cases of the **if** statement (lines 10 and 15), the algorithm stops at a timestamp that is not in $\langle t_i, x_i\rangle_N$ or $\langle \tau_j, y_j\rangle_M$, respectively. The missing position is assessed using an interpolation (lines 11 and 16, respectively). This new point is, then, used to compute the area spanned between the two trajectories since the previous timestamp.

The algorithm relies on the auxiliary function `Volume` to compute the areas. The true area between two segments is the sum of the distances between two moving particles along the segments. The computation of this integral is expensive [100]. It is instead approximated by the area of the trapezoids bounded by the four extremes of the two segments:

- If $X$ and $Y$ are step functions :

$$\mathtt{Volume}\Big((\mathtt{x_0},\mathtt{y_0},\mathtt{t_0}),(\mathtt{x_1},\mathtt{y_1},\mathtt{t_1})\Big) = d(\mathtt{x_0},\mathtt{y_0})|\mathtt{t_1}-\mathtt{t_0}|.$$

- If $X$ and $Y$ are polygonal functions :

$$\texttt{Volume}\Big((\texttt{x}_0, \texttt{y}_0, \texttt{t}_0), (\texttt{x}_1, \texttt{y}_1, \texttt{t}_1)\Big) = \frac{1}{2}\Big(d(\texttt{x}_0, \texttt{y}_0) + d(\texttt{x}_1, \texttt{y}_1)\Big)|\texttt{t}_1 - \texttt{t}_0|.$$

Note that the final `D` and `D/mT` correspond to DISSIM and $\Delta$, respectively.

**Algorithm 10** Computation of $\Delta_{\text{KL}}$

**Require:** $\langle t_i, x_i \rangle_N,\ \langle \tau_j, y_j \rangle_M,\ \lambda \in [0, 1]$.

$\texttt{xT}, \texttt{mT} = \max(t_N, \tau_M), \min(t_N, \tau_M)$
$\texttt{D}, \texttt{i}, \texttt{j} = 0, 1, 1$
$\texttt{x}, \texttt{y}, \texttt{t} = x_1, y_1, 0$
**while** $\texttt{t} < \texttt{mT}$ **do**
  **if** $t_{\texttt{i}+1} == \tau_{\texttt{j}+1}$ **then**
    $\texttt{D} \mathrel{+}= \texttt{Volume}\Big((\texttt{x}, \texttt{y}, \texttt{t}), (x_{\texttt{i}+1}, y_{\texttt{j}+1}, t_{\texttt{i}+1})\Big)$
    $\texttt{i} \mathrel{+}= 1$
    $\texttt{j} \mathrel{+}= 1$
    $\texttt{x}, \texttt{y}, \texttt{t} = x_{\texttt{i}}, y_{\texttt{j}}, t_{\texttt{i}}$
  **else if** $t_{\texttt{i}+1} > \tau_{\texttt{j}+1}$ **then**
    $\texttt{x}' = (x_{\texttt{i}+1} - x_{\texttt{i}})\frac{(\tau_{\texttt{j}+1} - t_{\texttt{i}})}{(t_{\texttt{i}+1} - t_{\texttt{i}})} + x_{\texttt{i}}$
    $\texttt{D} \mathrel{+}= \texttt{Volume}\Big((\texttt{x}, \texttt{y}, \texttt{t}), (\texttt{x}', y_{\texttt{j}+1}, \tau_{\texttt{j}+1})\Big)$
    $\texttt{j} \mathrel{+}= 1$
    $\texttt{y}, \texttt{t} = y_{\texttt{j}}, \tau_{\texttt{j}}$
  **else**
    $\texttt{y}' = (y_{\texttt{j}+1} - y_{\texttt{j}})\frac{(t_{\texttt{i}+1} - \tau_{\texttt{j}})}{(\tau_{\texttt{j}+1} - \tau_{\texttt{j}})} + y_{\texttt{j}}$
    $\texttt{D} \mathrel{+}= \texttt{Volume}\Big((\texttt{x}, \texttt{y}, \texttt{t}), (x_{\texttt{i}+1}, \texttt{y}', t_{\texttt{i}+1})\Big)$
    $\texttt{i} \mathrel{+}= 1$
    $\texttt{x}, \texttt{t} = x_{\texttt{i}}, t_{\texttt{i}}$
  **end if**
**end while**
**return** $(1 - \lambda)\log(\texttt{xT}/\texttt{mT}) + \lambda\texttt{D}/\texttt{mT}$

## 6.6 Empirical Evaluation

In this section, we conduct an empirical evaluation of our measure. Every experiment follows the same setting. To avoid heuristics on the number of clusters, we cluster using DP-means [156]. The computations are lighten by truncating the Dirichlet processes accordingly to [134], with up-to $K$ clusters and a concentration parameter equal to 0. Updating the centroids is costly, hence the assignments are decided with respect to the average dissimilarity. In each experiment, the retained clustering is the best out of 30 in terms of inertia or entropy, depending on the application. The

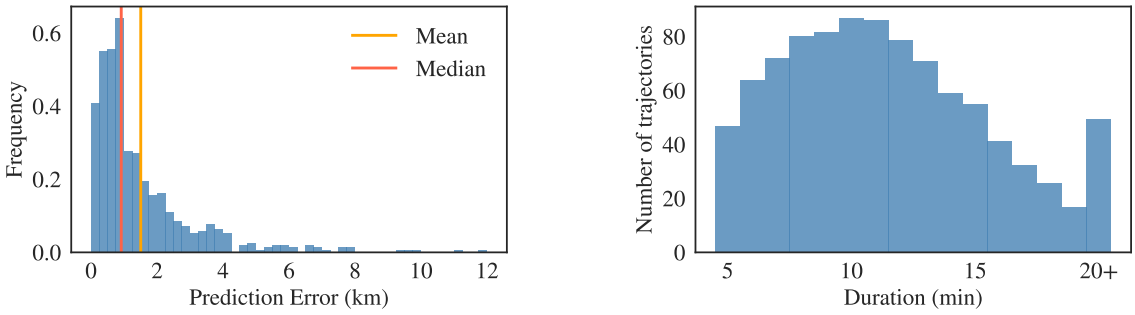



Figure 6.2: Left: Distribution of the prediction error. Right: Distribution of the duration of the journeys.

temporal window of LCSS is fixed to $\omega_T = 1$, while $\omega_D$ corresponds to the average square-root distance between two consecutive positions in the data-set. The hyper-parameter $\lambda$ of $\Delta_{\text{KL}}$ is learned using a grid search on $[0\,,1]$ with a step of 0.05.

### 6.6.1 Prediction of Taxi Journeys

Several approaches have been proposed to predict the final destination of a taxi trip based on partial trajectories, either by precomputing a clustering of the trajectories, or by taking grid-based approaches. Prediction models have been developed using mixture models [32], Bayesian inference [155], Markov chains [255, 169], trees [193, 60] and neural networks [77, 176, 220]. In this section we propose to predict the final destination of a taxi given a clustering of the set of trajectories.

The deterministic model we present here is similar to the one proposed in [32]. Suppose a given clustering of trajectories into $K$ groups computed with a distance $D$. For a cluster $k$, we denote the cluster's average trajectory by $\overline{C^{(k)}}$ and $E^{(k)}$ the average final destination within the cluster. The destination of an ongoing trajectory $Y$ at time $T$ is predicted by

$$\Psi(Y : b, \mathcal{C}) = \sum_{k=0}^{K-1} \frac{\exp\left(-bD(Y, \overline{C^{(k)}}_{:T})\right)}{\sum_{l=0}^{K-1} \exp\left(-bD(Y, \overline{C^{(l)}}_{:T})\right)} E^{(k)},$$

where $b > 0$ is a parameter called the *base* of $\Psi$. Note that it is important to distinguish the average final destination $E^{(k)}$ and the final destination of $\overline{C^{(k)}}$. Replacing former with the latter gives greater weight to longer trajectories and worsens the predictions.

We use data from a prediction challenge[1] [194]. We focus on the 5,000 first trajectories that start at "Campanhã" train station, contain no missings position, and are

[1] "Taxi Service Trajectory - Prediction Challenge, ECML PKDD 2015"

made of to 10 to 100 GPS coordinates, captured every 15 seconds, which corresponds to 2 min 30 sec to 25 min journeys. The quality of the predictions is evaluated using a 5-folds cross-validation. In each setting, the same measure is used for the clustering and the predictions. The clusterings are computed on training sets and correspond to the solution with the highest entropy. For each journey in the respective test set, a prediction is computed every minute between the 5th and the 20th minute of travel. The best performing base for $\Psi$ is chosen within the set $b \in \{1, 5, 10, \cdots, 100\}$. We measure the error, that is, the distance between the prediction returned by $\Psi$ and the true final destination using the haversine distance [241]. Note that the distribution of the errors is skewed, as shown in Figure 6.2. Thus, using the sample mean as descriptive statistics is inappropriate as a single outlier might pull it to the right. We resort, therefore, to the median for robustness. The results are shown in Figure 6.3.

Trips that are shorter than 10 minutes are well predicted by DISSIM, $\Delta$, and also $\Delta_{\text{KL}}$. While the former two degrade for longer durations, $\Delta_{\text{KL}}$ remains accurate until the end. Since $\Delta$ is a special case of $\Delta_{\text{KL}}$, varying $\lambda$ given the duration may improve performances of $\Delta_{\text{KL}}$ for short trajectories. After about 10 minutes, Hausdorff outperforms its peers, followed by SSPD, and DTW. However, their excellent performance comes at high computational costs.

Table 6.4 reports the average duration in seconds of prediction for a trajectory with 40 coordinates on an Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz with 252GB of RAM. While the computations of $\Psi$ using Hausdorff or SSPD require several seconds per run, predictions based $\Delta_{\text{KL}}$ are produced in about 60 milliseconds. Despite their better performance, the computational costs of Hausdorff, SSPD, and DTW renders

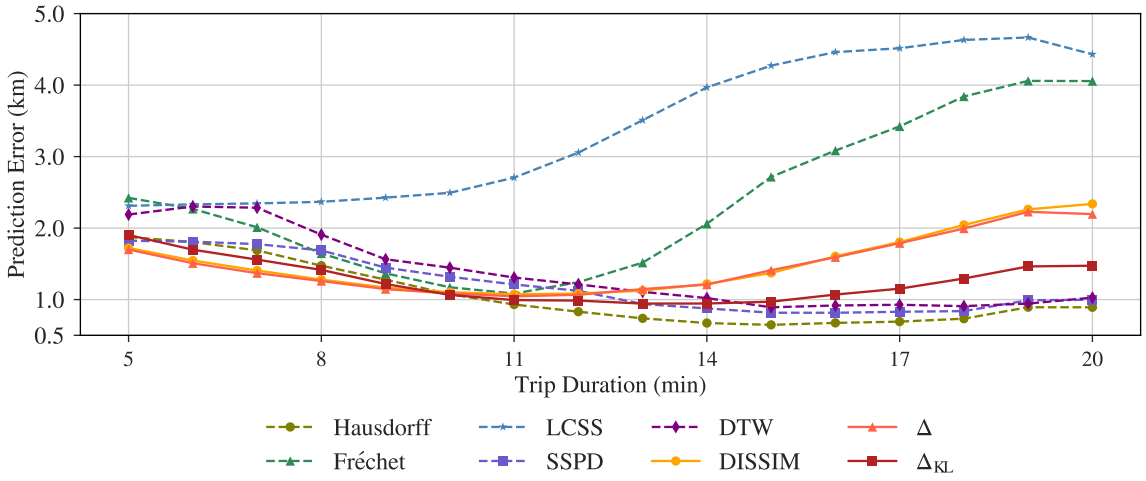



Figure 6.3: Prediction error over travel time.

Table 6.4: Computational costs in seconds.

| Hausdorff | Fréchet | LCSS | DTW | SSPD | DISSIM | $\Delta$ | $\Delta_{\mathrm{KL}}$ |
|---|---|---|---|---|---|---|---|
| 4.681 | 2.847 | 0.185 | 1.208 | 10.866 | 0.061 | 0.060 | 0.061 |

them inappropriate for real applications.

### 6.6.2 Discovering Flows

In this section, we study the extraction of flows. To appropriately address flow problems, we need to take into account that movements in one trajectory are delayed by a certain offset. We thus compute *shifted distances* [51] between a trajectory $X$ and the suffixes of $Y$ and vice versa and keep the minimum.

**Definition 6.20.** *Let $D$ be a measure between trajectories, the associated* shifted measure*, $D^S$, between two trajectories $X$ and $Y$ for a given minimum offset duration $\tau$, is given by:*

$$D^S(X,Y) = \min_{s\in[\tau-T_Y,T_X-\tau]} \Big(D\Big(X_{\max(0,s):}, Y_{\max(0,-s):}\Big). \tag{6.8}$$

*The argument of the minimum,* $\operatorname{argmin} D^S(X,Y)$*, is called the* time-shift associated to $D^S(X,Y)$.

We represent a set of trajectories $\mathcal{X}$ by a weighted directed graph such that trajectories are identified with nodes and edge weights are given by the shifted distances. A *flow*within $\mathcal{X}$ is a minimum directed spanning tree (MDST) of a connected component in that graph.

**Definition 6.21** (Flow)**.** *Let $\mathcal{X} = \{X_i\}_N$ be a set of $N$ trajectories, we consider the graph $G^+ = (V,E)$ weighted by the function $\omega$ such that:*

$$V = [0\,..\,n],\ E = \{(i,j) : \operatorname{argmin} D^S(X_i,X_j) \geq 0\},\ \omega(i,j) = D^S(X_i,X_j).$$

*A* flow *$F$ within $\mathcal{X}$ is a minimum directed spanning tree of a connected component of $G$, whose weights are the time-shifts associated to the edges.*

Note that this construction is not unequivocal. For instance, the components are not necessarily acyclic, hence the spanning trees are not unique [237]. We use [67, 91] to extract the flows.

### Discovering Attractors

In mathematics, a dynamical system is a system that evolves with time. It can be seen as a set of moving particles with certain constrains. Their study includes the discovery of these constrains or the study of their trajectories. We show that the latter can reveal the existence of *attractors*. Consider the system with a structure of two gyres or vortices [207] governed by the following equations:

$$\begin{array}{c} \forall\, x \in [0,2] \\ \forall\, y \in [0,1] \\ \forall\, t \in \mathbb{R}^+ \end{array} \quad \begin{cases} \dfrac{dx}{dt}(x,y,t) & = -\frac{\pi}{10}\, \sin(\pi f(x,y,t)) \cos(\pi y) \\ \dfrac{dy}{dt}(x,y,t) & = -\frac{\pi}{10}\, \cos(\pi f(x,y,t)) \sin(\pi y) \frac{df}{dt}(x,y,t) \\ f(x,y,t) & = \frac{1}{10} \sin(\frac{\pi t}{5})(\frac{x^2}{2} - x) + x \end{cases}$$

We use the Runge-Kutta fourth-order method [236] with 0.5 time unit increments between each point to generate 500 trajectories with 50 to 100 points, shown in Figure 6.4. Offsets of the shifted distance measures are at least 5 points. We use $K = 20$ and the entropy to select the clusterings. Figure 6.5 (left and center column) shows the three largest clusters that focus on trajectories around the right gyre. The flows extracted by $\Delta_{\mathrm{KL}}^{\mathrm{S}}$ are displayed in the third column, the corresponding clusters are shown in gray in the background. Dots and triangles indicate initial and final locations, respectively.

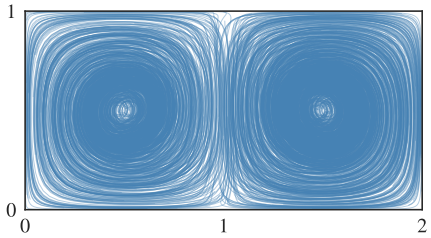



Figure 6.4: Two gyres dynamical system.

The first observation is that $\Delta_{\mathrm{KL}}$ failed to distinguish the two gyres. Its second largest cluster in Figure 6.5 (b) contains trajectories from both gyres. By contrast, its shifted version $\Delta_{\mathrm{KL}}^{\mathrm{S}}$ successfully separates the two parts of the system: Two groups stay in orbit (d and f) while the other one converges toward the gyre's center (e). Note how crisp are the shapes formed by the clusters. The analysis of the flows in the right column shows that the largest cluster of $\Delta_{\mathrm{KL}}^{\mathrm{S}}$ presents two flows that emerge from different orbits (g). Surprisingly, the flows first converge toward the center and then jump to a "higher" orbit. The flows in Figure 6.5 (h) plunges into the center of the gyre, while the one in Figure 6.5 (i) seems to stay in orbit, or it converges but at a much lower rate.

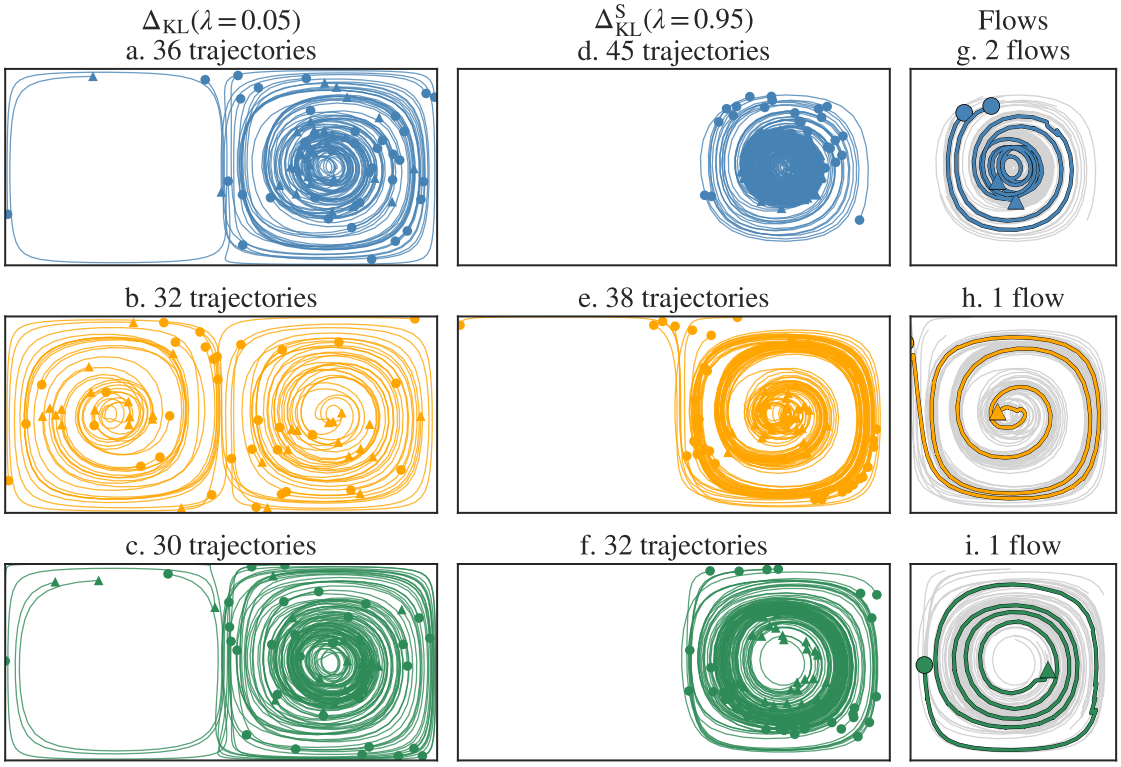



Figure 6.5: Clusterings and flows discovered by $\Delta_{KL}$ within the two gyres dynamical system.

### Clustering Drifters

Oceans can be modeled as a dynamical system as well. They do not have theoretical attractors, as that would mean that water accumulates on some location. However, they do present similar phenomena, especially gyres. To study the oceanic currents, GPS-tracked buoys (also called drifters) are released on all oceans. In this section, we show that a map of the oceanic currents can be learned by clustering the trajectories of these drifters.

We use data maintained by the US National Oceanic and Atmospheric Administration [44]. The information provided by the buoys is interpolated to 6-hour intervals [121]. We use 2,168 linearly interpolated trajectories collected between 2005 and 2009. The position of the buoys are given in terms of latitude and longitudes.

We focus the analysis on the north Atlantic ocean. In particular, we want to extract the Gulf Stream (GF) and its descendant, the North Atlantic Drift (NAD) [234]. The Gulf Stream takes its source in the hot waters of the Gulf of Mexico. It then follows the East coast of North America before flowing into the Atlantic Ocean. The

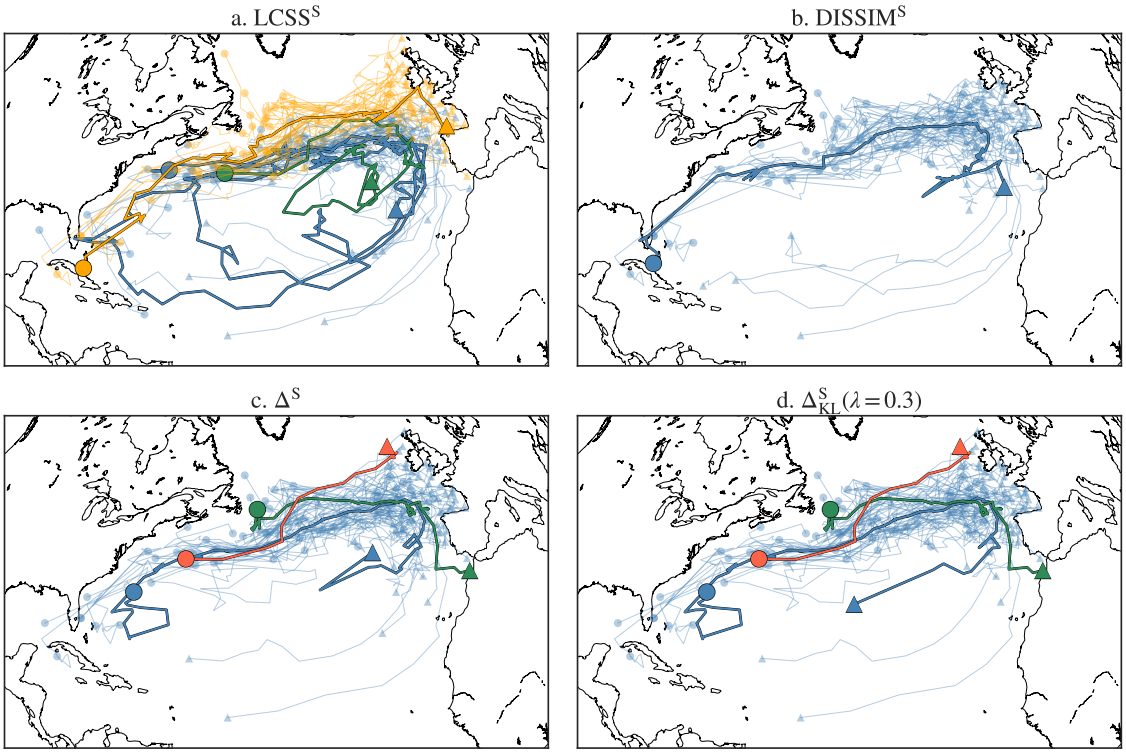



Figure 6.6: Clusters covering the North Atlantic ocean with their flows. The orange cluster of LCSS$^S$ has a single flow drawn in the same color.

North Atlantic Drift takes over and brings the warm water to Europe's West coast. It branches out toward the North as the Norwegian current, and to the South as the Canary Current (CAN).

Given the computational cost, we only compare the distances with linear time complexities, namely: LCSS, DISSIM, $\Delta$, and $\Delta_{KL}$. We use again offsets of at least 5 points and the entropy to select the best clusterings for $K = 50$. In Figure 6.6, we show only the clusters containing the currents of interests. The dots and triangles indicate, respectively, the initial and last position of each buoy trajectory. Thick curves represent the flows.

The simplest clustering is returned by DISSIM$^S$ and features one group with a single flow. Interestingly, only LCSS$^S$ separated the buoys drifting in the region into two clusters along a north-south axis. By contrast, $\Delta^S$ and $\Delta^S_{KL}$ embed several flows in their cluster. The red one roughly corresponds to the northern cluster of LCSS$^S$.

In this section, we have seen the representation power of the flows. However, not all trajectory measures make the best out of it. For example, oceanic flow of DISSIM$^S$ is *correct* but as informative as that of $\Delta^S_{KL}$.

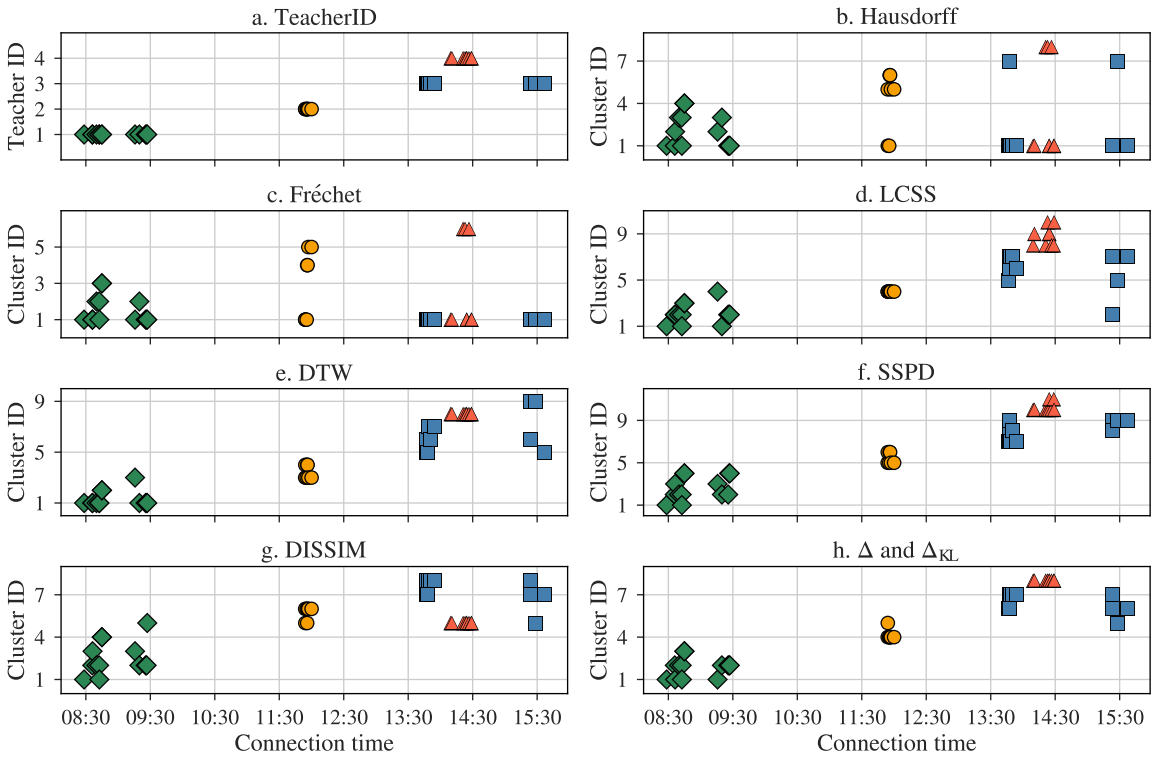



Figure 6.7: Teacher and cluster assignments of each sessions for $K = 20$ .

### 6.6.3 User-sessions as Trajectories

The page graph of the mBook resembles a directed tree with some extra edges. Trajectories are sequences of page ids and timestamps indicating the time when the page was loaded. Formally, $\Omega$ is made of the pages plus an extra node indicating the loss of focus, different from the point at infinity which represents the end of a session. The focus node is connected to all the other pages with an arbitrary large weight in both directions to penalize the loss of focus. This node favor the grouping of unfocused users independently of the content. The end event $\infty$ is added few seconds after the last recorded event. Accordingly to Definition 2.1, the trajectories are modeled as step-functions to implement the assumption that the user stays on the same page between two events. The shortest path [84] suggests itself as a natural metric on the tree although is not symmetric as the page graph is directed. Since there are very few asymmetries, we pursue the analysis with it.

#### Clustering

We show that the proposed similarity measure $\Delta_{\mathrm{KL}}$ assigns similar user sessions to the same cluster and the computed groups exhibit similar behavior. We thus focus

Table 6.5: Number of clusters and homogeneity scores.

| | | Hausdorff | Fréchet | LCSS | DTW |
|---|---|---|---|---|---|
| $K = 4$ | # Clusters | 4 | 4 | 4 | 4 |
| | Homog. | 0.195 | 0.216 | 0.846 | 0.945 |
| $K = 20$ | # Clusters | 8 | 6 | 5 | 9 |
| | Homog. | 0.459 | 0.334 | 0.877 | 0.954 |
| | | SSPD | DISSIM | $\Delta$ | $\Delta_{\text{KL}}$ |
| $K = 4$ | # Clusters | 4 | 4 | 4 | 4 |
| | Homog. | 1.0 | 0.676 | 0.868 | 0.868 |
| $K = 20$ | # Clusters | 11 | 8 | 8 | 8 |
| | Homog. | 1.0 | 0.776 | 0.975 | 0.975 |

on sessions from different classes and aim to re-identify the classes in the clustering. The data consists of 41 sessions initiated by 37 pupils during a single day. The ground-truth clustering groups pupils with respect to their teacher, see Figure 6.7 (a). Each dot represents a user session. The x-axis shows connection times of the sessions and the $y$-axis the cluster/teacher id. The six classes of this very day, given by four different teachers, are easily identifiable by the timestamps of the sessions and topic of the class: teachers 1 and 3 study *Classical Antiquity*, teacher 2 *World War 2*, and teacher 4 devotes that day to the *Reformation*. Table 6.5 shows the results for the re-identification of the four classes, where we use $K = 4$ (top row) which is also the true number of classes, and $K = 20$, to allow for capturing diverse behavior. The parameter of $\Delta_{\text{KL}}$ is set to $\lambda = 0.14$ (highest entropy). The quality of the clusterings is the homogeneity score [219] of the run with the lowest inertia. The resulting distributions for $K = 20$ are displayed in Figure 6.7.

Clustering based on Hausdorff and Fréchet distances fail to recover the teachers and have only small homogeneity scores. SSPD perfectly groups the sessions with respect to the teachers. DISSIM, $\Delta$ and $\Delta_{\text{KL}}$ perform better when $K = 20$. Figure 6.7 (h) shows that $\Delta$ and $\Delta_{\text{KL}}$ successfully re-identify the ground-truth clustering with a minor flaw: two sessions in cluster 5 are wrongly grouped together. The homogeneity score indicate that the measures successfully detect topics. However, it is yet unclear whether the groupings make sense in terms of behaviors: e.g., DISSIM has a homogeneity greater than 0.7 but its fifth cluster is a mix of sessions from different classes (Figure 6.7 (g)). It is, therefore, unlikely that this cluster presents any consistency in terms of behavior.

Figure 6.8 shows the trajectories from classes from teacher 3 of two clusters for each measure. Each curve corresponds to a session evolving over time (x-axis). The end of the sessions is denoted by a square. The y-axis shows the type of the page, where *other* indicates that the viewed page is not part of the lecture's main chapter "*Classical Antiquity*". DTW shows a balanced grouping of sessions that can be warped onto each other (c/d). The other measures identify a large cluster and a few small ones. $\Delta$ and $\Delta_{\mathrm{KL}}$ seem to differentiate the clusters according to the loss of focus event after about 20 minutes.

To wrap up, an appropriate clustering of user sessions respects topic and behavior. Clusterings based on Hausdorff and Fréchet distances fell short in both senses. DISSIM ignored the topic and children visiting different chapters of the textbook are grouped together (cluster 5 in Figure 6.7.g). Although LCSS, DTW, $\Delta$ and $\Delta_{\mathrm{KL}}$ return different clusterings, they all satisfy both objectives.

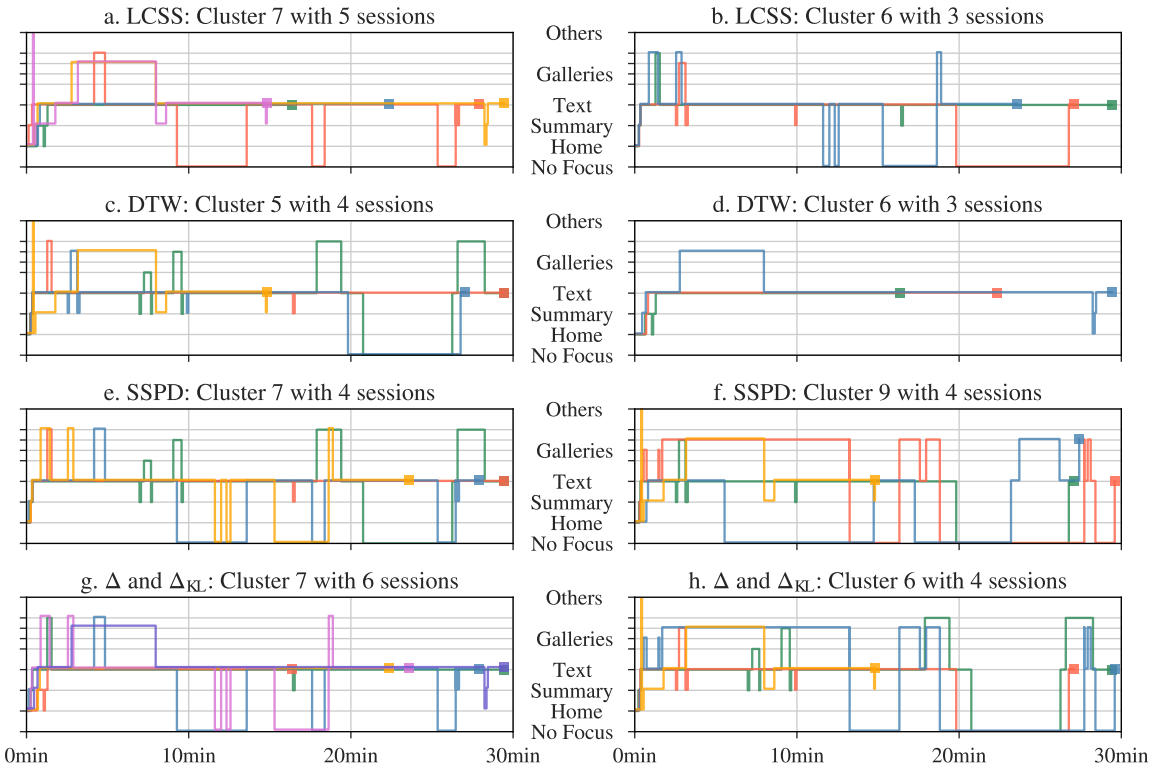



Figure 6.8: Trajectories of two clusters associated to the class of Teacher 3 for different measure.

Table 6.6: Summary of the analyzed classes.

| | #Class | #Sessions | #Pupils | avg.$\psi$ | avg.PPM | avg.EPM |
|---|---|---|---|---|---|---|
| Teacher A | 27 | 276 | 65 | 3.33 ( 0.83 ) | 0.97 (0.57) | 0.98 (0.47) |
| Teacher B | 11 | 80 | 22 | 2.7 ( 0.96 ) | 1.0 (0.65) | 1.01 (1.02) |

**Class Behavior**

In this section, we study the relationship between the pupil's psychometric scores and the expressed behaviors, especially in terms of activity. The latter can be estimated by the *number of pages seen per minute* (PPM) or the *number of events per minute* (EPM). However, these statistics can only be compared relatively to the average value of each class. Indeed, in a class with an average number of pages seen per minute of 1, a user viewing 1 page per minute is considered as regular. On the other hand, if the average of the class is 3, the same user appears rather inactive.

We also propose an indicator based on a dissimilarity measure, here $\Delta_{\text{KL}}$. We define $\psi(\boldsymbol{s})$ as the average dissimilarity ($\Delta_{\text{KL}}(\lambda = .55)$) between session $\boldsymbol{s}$ and the other sessions occurring during the same class. A high $\psi(\boldsymbol{s})$ means that the user behind session $\boldsymbol{s}$ acts in a singular manner in comparison to the average behavior of the class. A small average $\psi$ in a group indicates that the pupils use the mBook the same way, i.e., they move from one page to another almost synchronously. On the other hand, a large average $\psi$ signals that pupils have more freedom in terms of navigation and usage of the resources.

For this analysis, we extract 359 classes between February and July 2017 supervised by two teachers (A and B) in two different schools. A *class* is defined as a cluster of at least five sessions initialized within a 10 minutes interval, by pupils tagged with the same teacher, and occurring between 08:00 and 16:00. Table 6.6 reports the number of classes, sessions, and pupils per teacher. The average intra-class dissimilarity of the teachers' classes ($\Psi$), as well as the average PPM and EPM are given in the last three columns with standard deviations. Correlations between the average $\psi$ for each pupils and psychometric scores are reported in Table 6.7. Pearson's correlations with a $p$-value smaller than 5% are marked in bold face.

Table 6.6 reveals that the classes of teacher A presents a higher entropy, reflected by a high average $\psi$, in comparison to teacher B. Remark that at the same time the average PPM and EPM of both teachers are almost equal. This difference of average $\psi$ suggests that the teachers apply different teaching styles. A Mann-Whitney U test [185, 97] between the average $\psi$ of the two teachers' classes returns a U-value

Table 6.7: Pearson's correlations and associated p-values for each combination of pupils' activity indicators and score.

| | | Teacher A | | | Teacher B | | |
|---|---|---|---|---|---|---|---|
| | | $\psi$ | PPM | EPM | $\psi$ | PPM | EPM |
| Competency | $r$ | **0.189** | **0.145** | **0.185** | **-0.245** | **-0.232** | **-0.232** |
| | $p$-value | 0.008 | 0.044 | 0.009 | 0.025 | 0.039 | 0.04 |
| Knowledge | $r$ | 0.099 | 0.133 | **0.156** | -0.17 | 0.049 | -0.141 |
| | $p$-value | 0.168 | 0.064 | 0.03 | 0.135 | 0.671 | 0.216 |
| Motivation | $r$ | **-0.155** | 0.039 | -0.065 | 0.072 | 0.111 | -0.142 |
| | $p$-value | 0.03 | 0.587 | 0.37 | 0.51 | 0.331 | 0.212 |
| IT Access | $r$ | 0.011 | -0.002 | -0.022 | -0.071 | 0.188 | 0.081 |
| | $p$-value | 0.877 | 0.979 | 0.761 | 0.534 | 0.097 | 0.481 |
| IT Skill | $r$ | 0.107 | 0.019 | 0.063 | **-0.379** | -0.156 | 0.059 |
| | $p$-value | 0.135 | 0.789 | 0.381 | 0.001 | 0.171 | 0.604 |

of 95 (96 critical) and a one-sided $p$-value of 0.044. In other words, the difference is significant. Remark this that the average PPM and EPM per teacher do not suggests this

The fact that the three indicators compared in Table 6.7 correlate with competency could mistakenly be interpreted as they are redundant. However, we observe cases where only $\psi$ correlates significant: e.g., a small $\psi$ correlates with high motivation in group A.

Note the correlations between the three activity indicators and competency have different signs for teacher A and B. These differences should be interpreted in the light the class average $\psi$, given in Table 6.6. We can thus surely state that pupils in teacher B's classes more or less all do the same at the same time and pupils who diverge from the predominant path perform worse. In contrast, the worst performing pupils of teacher A, whose classes present in average a larger average $\psi$, are those that under-use the textbook.

In addition to the classical PPM and EPM, $\psi$ also captures the pupils' activity that correlates with competency in a direction depending on the teaching style. However, this difference of pedagogy choices is only captured by the average $\psi$ and not by the average PPM and EPM.

## 6.7 Conclusion

We proposed a theoretical framework that allows for a rigorous definition of spatio-temporal similarity measures. It also yields a classification scheme that captures the distinction between edit and shape-based approaches. The desired class of conformal measures naturally arise from this discussion. Such measures induce a dissimilarity on the quotient set, are time scale-invariant, and have linear time complexity. Reviewing the prominent existing measures showed that none of them fulfills all conformal conditions. Hence, we proposed the first conformal dissimilarity measure $\Delta_{\mathrm{KL}}$ for sequential data. $\Delta_{\mathrm{KL}}$ derives from the KL divergence between the generalized Laplace distributions induced by the trajectories. Our measure corresponds to the normalized point-wise distance with a penalty for differences in duration. This penalty term notably distinguishes a trajectory from its prefixes. Empirical evaluations on clusterings and predictions tasks using $\Delta_{\mathrm{KL}}$ showed that it performs always on par or better compared to existing measures.

We showed that trajectory dissimilarities can extract pupils' very different types of behaviors during classes. We also showed that the average dissimilarity between sessions during a class can thus be turned into an effective indicator of pupil performance and teaching technique. Finally, this study attests that modeling sessions as trajectories is a relevant method for the analysis of learning and teaching behaviors.

# Chapter 7

# Deep Clustering as a Unifying Method

This chapter is an opening toward future works. We have seen that mixture models merge modeling and clustering into a single step. The work that we present here targets the same simplification using neural networks. That way, modeling hypotheses could be reduced to a choice of architecture.

Clustering is one of the oldest and most difficult problems in machine learning, especially in high dimensions [55] and for complex data. In recent years, there has been increased interest in designing deep learning-based clustering approaches [254, 115, 142, 139]. One dominant line of research, focusing on designing centroid-based clustering algorithms and combining dimensionality reduction with clustering, has in particular delivered promising results [254, 115, 268].

These approaches tend to utilize autoencoder architectures to perform dimensionality reduction and perform clustering in feature space. However, these two steps are often separated [254, 256]. Moreover, the design of the proposed approaches is mainly ad-hoc and empirically-driven, and the proposed algorithms rely heavily on good initialization by autoencoder pre-training. However, the latter has been shown to often have a negative influence on clustering performance due to inconsistent optimization goals [240, 256].

Here, we take a step back and provide a novel theoretical analysis where we show that isotropic Gaussian mixture models (GMMs) can be formulated in the form of an objective function for a neural autoencoder. We denote the resulting neural network a *clustering module* (CM). The CM can easily be inserted into deeper architectures to give rise to our proposed C-Net, which jointly learns a lower-dimension embedding and a clustering. Empirically, we demonstrate that CM performs on par with $k$-means and GMMs. While these comparisons serve more as a sanity check as CM is derived

from GMMs, we also show that C-Net outperforms other deep clustering approaches in most of the scenarios.

The chapter is organized as follows. Section 7.1 reviews related work. We derive our main contribution in Section 7.2 and discuss implementation issues in Section 7.3. Section 7.4 reports empirical results, and finally Section 7.5 concludes.

## 7.1 Related Work

Several approaches based on deep learning have been proposed to group data in a supervised or semi-supervised manner [145, 177]. We address here the unsupervised task that has recently become an active research area. In addition to specialized clustering algorithms for image-specific tasks [257, 129, 59, 130, 56, 116], several generalist approaches exist [256, 115, 254, 139]. We differentiate between centroid-based approaches and those that do not rely on centroids in the remainder.

**Non-centroid methods** These include generative models such as generative adversary networks (GAN) [109] and Variational Autoencoders (VAE) [146]. For example, CatGAN [231] predicts a categorical distribution while maximizing robustness against an adversarial generative model. The adversarial autoencoder [181] is close to VAE since it uses two adversarial networks to impose a Gaussian distribution and a categorical distribution in latent space. The Deep Embedding Network (DEN) aims at learning an embedding that facilitates $k$-means clusterings [179], through locality preserving and group sparsity constraints. Kampffmeyer et al. [142] use divergence measures drawn from Information Theory to obtain separability and compactness of the clusters. Recently, Invariant Information Clustering (IIC) [139] uses mutual information to train a network to embed an image and its distortions close together in the embedding space. This model belongs to the families of agglomerative [110] and co-clustering [82] approaches that have recently met with some success [22, 257]

**Centroid-based Methods** Deep Embedded Clustering (DEC) [254] is one of the first generally applicable contributions to centroid-based deep clustering. First, an autoencoder is pre-trained as a stacked denoising autoencoder (SDA) [245]. The decoder part is then discarded and replaced by a fully connected layer whose weights represent the centroids. The loss function derives from that of t-SNE [178]: the optimization learns an embedding and centroids such that points aggregate around their nearest centroid with as little ambiguity as possible. Guo et al. [115] improve

the robustness of the model by keeping the autoencoder complete in both phases and add its reconstruction loss to serve as a regularizer. The Deep Clustering Network (DCN) [256] is based on a loss and architecture comparable to IDEC but includes hard clustering. Consequently, the optimization scheme has three steps: update of the network weights with stochastic gradient descent (SGD) [235]; assignments to clusters; update of the centroids using the assignments in gradient-descent fashion.

Deep Adaptive Image Clustering [116] combines an agglomerative measure and centroids to jointly learn an embedding and clustering. The Deep Autoencoder Gaussian Mixture Model [268] aims at anomaly detection, however, it does represent progress toward an end-to-end centroid-based deep clustering. The structure is similar to IDEC: a deep autoencoder and an auxiliary network that learns centroids. As its name suggests, the loss function includes the log-likelihood of a GMM.In practice, the model has shown to be computationally unstable since it includes the invertion of the covariance matrix.

## 7.2 Towards a Theoretically-Grounded Clustering Network

To obtain a centroid-based clustering loss function, it is reasonable to start from the objective function of a GMM. In the following, we show that, under some rather general assumptions, the $\mathcal{Q}$-function maximized by the Expectation-Maximization algorithm (EM) [80, 35] for a GMM is a regularized loss function for an autoencoder. We then discuss appropriate (deep) neural architectures for that loss function.

### 7.2.1 From GMMs to Autoencoders

Throughout the chapter, $\mathcal{X} = \{\boldsymbol{x}_i\}_N \subset \mathbb{R}^d$ refers to an *i.i.d.* sample of $N \in \mathbb{N}$ points and $\mathcal{Z} = \{z_i\}_N$ aggregates the corresponding assignments to $1 \leq i \leq K$ clusters. When the context allows, the range of the indices are abbreviated using the upper-bound, e.g., a vector $\boldsymbol{x} \in \mathbb{R}^d$ is written as $\boldsymbol{x} = \langle x_i \rangle_{1 \leq i \leq d} = \langle x_i \rangle_d$. The zero vector and the vector containing only ones in $\mathbb{R}^d$ are denoted as $\mathbf{0}_d$ and $\mathbf{1}_d$, respectively. The set of the stochastic vectors is given by

$$\mathbb{S}^d = \{\boldsymbol{x} \in \mathbb{R}^d_{\geq 0} \ : \ \sum_{i=1}^{d} x_i = 1\}.$$

Consider a Gaussian mixture model with $K$ components. The centroids are summarized in the matrix $\boldsymbol{\mu} = \langle \boldsymbol{\mu}_k \rangle_K \in \mathbb{R}^{K \times d}$. The mixture weights form a stochastic

vector

$$\Phi = \langle \phi_k := p(z = k) \rangle_K \in \mathbb{S}^K$$

and the responsibility of cluster $k$ on a data-point $x_i$ is $\gamma_{ik} := p(z_i = k|x_i)$. The vector made of the cluster responsibilities on $x_i$ is the stochastic vector $\boldsymbol{\gamma}_i \in \mathbb{S}^K$. The mean responsibility of cluster $k$ is $\tilde{\boldsymbol{\gamma}}_k = 1/N \sum_{i=1}^N \gamma_{ik}$. These vectors form the rows of matrix $\Gamma = \langle \gamma_{ik} \rangle_{N \times K} \in \mathbb{R}^{N \times K}$.

Similarly to an EM-based optimization, we maximize the $\mathcal{Q}$-function associated to the mixture instead of the intractable log-likelihood [80]. Since the $\mathcal{Q}$-function is a lower-bound of the log-likelihood, minimizing $-\mathcal{Q}$ also maximizes the likelihood of the underlying GMM.

To obtain a tractable derivation, it is helpful to avoid the inversion of covariance matrices and to assume an isotropic mixture. For simplicity, we focus on equal covariance matrices of the form $\frac{1}{2}\mathbf{I}_d$. Secondly, in order to control the distribution of assignments across the clusters, we assume a Dirichlet prior distribution on the mixture weights with parameter $\boldsymbol{\alpha} \in \mathbb{R}^K_{\geq 0}$, given by

$$p(\Phi) \propto \prod_{k=1}^K \phi_k^{\alpha_k - 1}.$$

Both $\Phi$ and $\gamma_i$ influence the assignment of $x_i$ to a cluster but they may present diverging behaviors. The former governs the average distribution of the data-points across all clusters, while the latter decides for a single data-point. In order to avoid redundancy and related issues, without loss of generality, we can suppose that we start our derivations after a Maximization step (M-step) of the EM, when the mixture weights are updated by

$$\phi_k = \frac{1}{N} \sum_{i=1} \gamma_{ik} = \tilde{\boldsymbol{\gamma}}_k.$$

Under theses assumptions, after an M-step, the negative of the $\mathcal{Q}$-function associated to an isotropic Gaussian mixture model [35] with a Dirichlet prior for the mixture weights is, up to a constant, equal to:

$$-\mathcal{Q}(\boldsymbol{\Gamma}, \boldsymbol{\mu}) = \sum_{i=1}^N \sum_{k=1}^K \gamma_{ik} \log \tilde{\boldsymbol{\gamma}}_k + \gamma_{ik} ||\boldsymbol{x}_i - \boldsymbol{\mu}_k||^2 + \sum_{k=1}^K (1 - \alpha_k) \log(\tilde{\boldsymbol{\gamma}}_k), \tag{7.1}$$

Note that we focus on minimizing $-\mathcal{Q}$ to exploit relations to empirical risk minimization and the use of loss functions. The first term simplifies as the entropy of $\tilde{\boldsymbol{\gamma}} = \langle \tilde{\boldsymbol{\gamma}}_k \rangle_K$ which is a function of $\boldsymbol{\alpha}$ and constant with respect to the model's parameters,

$$\sum_{i=1}^N \sum_{k=1}^K \gamma_{ik} \log \tilde{\boldsymbol{\gamma}}_k = N \sum_{k=1}^K \tilde{\boldsymbol{\gamma}}_k \log \tilde{\boldsymbol{\gamma}}_k = -N H(\tilde{\boldsymbol{\gamma}}) = F(\boldsymbol{\alpha}).$$

The second term of Equation 7.1 can be transformed to allow for an interpretation as a reconstruction term. For any given $i \in [1 \,..\, N]$, we obtain

$$\begin{aligned}\sum_{k=1}^{K}\gamma_{ik}||\boldsymbol{x}_i-\boldsymbol{\mu}_k||^2 &= \sum_{k=1}^{K}\gamma_{ik}||\boldsymbol{x}_i||^2 - 2\boldsymbol{x}_i^\top\Big(\sum_{k=1}^{K}\gamma_{ik}\boldsymbol{\mu}_k\Big) + \sum_{k=1}^{K}\gamma_{ik}||\boldsymbol{\mu}_k||^2 + ||\tilde{\boldsymbol{x}}_i||^2 - ||\tilde{\boldsymbol{x}}_i||^2 \\ &= ||\boldsymbol{x}_i||^2 - 2\boldsymbol{x}_i^\top\tilde{\boldsymbol{x}}_i + ||\tilde{\boldsymbol{x}}_i||^2 + \sum_{k=1}^{K}\gamma_{ik}||\boldsymbol{\mu}_k||^2 - \sum_{k=1}^{K}\gamma_{ik}^2||\boldsymbol{\mu}_k||^2 \\ &= ||\boldsymbol{x}_i-\tilde{\boldsymbol{x}}_i||^2 + \sum_{k=1}^{K}(\gamma_{ik}-\gamma_{ik}^2)||\boldsymbol{\mu}_k||^2.\end{aligned}$$

where $\tilde{\boldsymbol{x}}_i = \sum_{k=1}^{K}\gamma_{ik}\boldsymbol{\mu}_k$. The $\mathcal{Q}$-function becomes the sum of three terms, which we discuss in the following,

$$-\mathcal{Q}(\Gamma,\boldsymbol{\mu}) = \underbrace{\sum_{i=1}^{N}||\boldsymbol{x}_i-\tilde{\boldsymbol{x}}_i||^2}_{=:E_1} + \underbrace{\sum_{i=1}^{N}\sum_{k=1}^{K}(\gamma_{ik}-\gamma_{ik}^2)||\boldsymbol{\mu}_k||^2}_{=:E_2} + \underbrace{\sum_{k=1}^{K}(1-\alpha_k)\log(\tilde{\boldsymbol{\gamma}}_k)}_{=:E_3}. \tag{7.2}$$

Note that the EM algorithm is guaranteed to lead the likelihood toward a local maximum as $\mathcal{Q}$ is equal to the log-likelihood after each E-step. For our proposed optimization procedure, however, we do not perform the E-step, and therefore the negative of Equation 7.2 remains a lower-bound of the log-likelihood [35].

**$E_1$: Reconstruction** The first term $E_1$ can be interpreted as a reconstruction loss for mapping $\boldsymbol{x}_i$ to $\tilde{\boldsymbol{x}}_i$. Indeed, the cluster responsibilities $\gamma_{ik}$ models the posterior probability of $z_i$ given $\boldsymbol{x}_i$. Therefore, the vector $\boldsymbol{\gamma}_i$ can also be seen as the projection of $\boldsymbol{x}_i$ into $\mathbb{S}^K$ by a function $\mathcal{E}nc : \mathbb{R}^d \to \mathbb{S}^K$, parameterized with $\boldsymbol{\eta}$. On the other hand, $\tilde{\boldsymbol{x}}_i$ is the image of $\boldsymbol{\gamma}_i$ by the linear function $\mathcal{D}ec : \mathbb{S}^K \to \mathbb{R}^d$ with parameter $\boldsymbol{\mu}$. We thus write

$$\tilde{\boldsymbol{x}}_i = \mathcal{D}ec(\mathcal{E}nc(\boldsymbol{x}_i;\boldsymbol{\eta});\boldsymbol{\mu}) = \mathcal{D}ec(\boldsymbol{\gamma}_i;\boldsymbol{\mu}) = \sum_{k=1}^{K}\gamma_{ik}\boldsymbol{\mu}_k. \tag{7.3}$$

The term $E_1$, therefore, suggests an autoencoder structure, where $\mathcal{E}nc$ and $\mathcal{D}ec$ correspond to the encoding and decoding function, respectively.

**$E_2$: Sparsity and Regularization** The second term $E_2$ is simply the Gini impurity index [46] applied to $\boldsymbol{\gamma}_i$. This can be shown by

$$\sum_{k=1}^{K}(\gamma_{ik}-\gamma_{ik}^2)||\boldsymbol{\mu}_k||^2 \leq \sum_{k=1}^{K}\Big(\gamma_{ik}-\gamma_{ik}^2\Big)||\boldsymbol{\mu}||_F^2 = \mathbf{Gini}(\boldsymbol{\gamma}_i)||\boldsymbol{\mu}||_F^2.$$

This measure occurs in decision tree theory to select features for branching and equivalent to the entropy. The Gini index is non-negative and vanishes when $\boldsymbol{\gamma}_i$ is a one-hot vector, thereby cancelling out $(\gamma_{ik} - \gamma_{ik}^2)$.

The terms $||\boldsymbol{\mu}_k||^2$ play a role similar to an $\ell_2$-regularization: they prevent the centroids from diverging away from the data-points. However, they may also favor the trivial situation where all the centroids are concentrated in zero.

$E_3$**: Balancing** The Dirichlet prior is introduced to steer the distribution of the cluster assignments. It may also compensate for the penchant of $E_2$ for the trivial clustering. Note that $E_3$ can be re-written in terms of a Kullback-Leibler (KL) divergence. In particular, for $\boldsymbol{\alpha} = \left(1 + \frac{1}{K}\right)\mathbf{1}_K$, the negative prior is, up to a constant, the KL divergence for a multinomial distribution with parameter $\tilde{\boldsymbol{\gamma}}$ and the uniform multinomial distribution:

$$D_{KL}\left(\frac{1}{K}\mathbf{1}_K \middle|\middle| \tilde{\boldsymbol{\gamma}}\right) = \sum_{k=1}^{K}(1 - (1 + \frac{1}{K}))\log(\tilde{\boldsymbol{\gamma}}_k) + \mathrm{C}.$$

### 7.2.2 Clustering Module

We define the *Clustering Module* (CM) as the autoencoder where $\mathcal{E}nc$ is the combination of an affine transformation and a softmax. Without loss of generality, we also assume $\mathcal{D}ec$ affine instead of linear (Equation 7.3). The operations involved in CM are formalized as follows:

$$\mathcal{E}nc(\boldsymbol{X}) = \mathrm{softmax}(\boldsymbol{X}\boldsymbol{W}_{\mathrm{enc}} + \boldsymbol{B}_{\mathrm{enc}}) = \boldsymbol{\Gamma}$$

$$\mathcal{D}ec(\boldsymbol{\Gamma}) = \boldsymbol{\Gamma}\boldsymbol{W}_{\mathrm{dec}} + \boldsymbol{B}_{\mathrm{dec}} = \tilde{\boldsymbol{X}},$$

for input $\boldsymbol{X} \in \mathbb{R}^{N\times d} \sim \mathcal{X}$, code representation also representing the cluster responsibilities $\boldsymbol{\Gamma} = \langle\gamma_{ik}\rangle_{N\times K} \in \mathbb{R}^{N\times K}$ s.t. $\boldsymbol{\gamma}_i \in \mathbb{S}^K$, and reconstruction $\tilde{\boldsymbol{X}} \in \mathbb{R}^{N\times d}$, respectively. The weight and bias parameters for the encoder are $\boldsymbol{W}_{\mathrm{enc}} \in \mathbb{R}^{d\times K}$ and $\boldsymbol{B}_{\mathrm{enc}} \in \mathbb{R}^K$, respectively, and analogously for the decoder $\boldsymbol{W}_{\mathrm{dec}} \in \mathbb{R}^{K\times d}$ and $\boldsymbol{B}_{\mathrm{dec}} \in \mathbb{R}^d$. The softmax is used as the activation function after the first layer to enforce the stochasticity of the code. The centroids $\boldsymbol{\mu}$ of the underlying GMM correspond to the column-vectors of matrix $\boldsymbol{W}_{\mathrm{dec}} + \boldsymbol{B}_{\mathrm{dec}}$. For consistency, we repeat the full formula of the loss function:

$$\mathcal{L}_{\mathrm{CM}}(\mathcal{X};\Theta) = \sum_{i=1}^{N}||\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i||^2 + \sum_{i=1}^{N}\sum_{k=1}^{K}(\gamma_{ik} - \gamma_{ik}^2)||\boldsymbol{\mu}_k||^2 + \sum_{k=1}^{K}(1 - \alpha_k)\log(\tilde{\boldsymbol{\gamma}}_k), \quad (7.4)$$

where $\Theta$ represents the parameters of the network. In the case of batch-based optimization, $\tilde{\gamma}$ is computed over the batch, hence the $N$ is replaced by the size of the batch.

### 7.2.3 Clustering Network

The proposed CM enables a neural implementation of a GMM. The CM may thus be seamlessly integrated as a component in larger neural networks. It is well known in clustering that discriminative dimensionality reduction may aid the clustering process. We therefore nest the clustering module in a deep autoencoder (DAE). This enables a discriminative (non-linear) embedding of input data into a lower-dimensional space to obtain clusters, by performing the dimensionality reduction and the CM optimization jointly and end-to-end. At the same time, we are leveraging the regularization properties of the DAE reconstruction loss. We refer to this deep architecture as *Clustering Network* (C-Net).

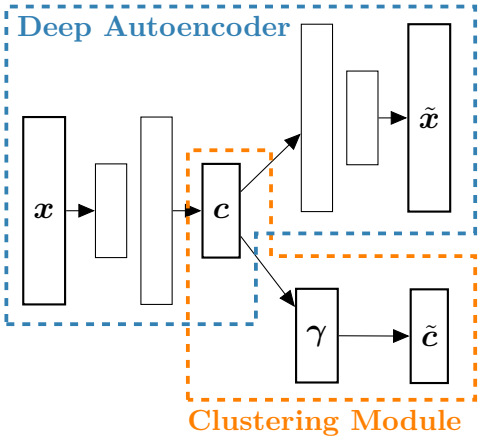



Figure 7.1: The proposed Clustering Network (C-Net) combines a deep autoencoder with a clustering module (CM).

The first part of the network encodes an input $\boldsymbol{x}$ into a vector $\boldsymbol{c}$. The latter is then fed to a CM and to the decoder of the DAE, yielding two outputs: $\tilde{\boldsymbol{c}}$, the CM reconstruction of the vector $\boldsymbol{c}$, and $\tilde{\boldsymbol{x}}$, the reconstruction of $\boldsymbol{x}$. The architecture is illustrated in Figure 7.1.

With this architecture, the additional regularization term $||\boldsymbol{\mu}_k||$ is no longer needed, and we hence remove it from $E_2$. We discuss the benefit of this choice in later, resulting in the following loss function:

$$\mathcal{L}_{\text{C-Net}}(\mathcal{X};\Theta)=\sum_{i=1}^{N}||\boldsymbol{x}_i-\tilde{\boldsymbol{x}}_i||^2+||\boldsymbol{c}_i-\tilde{\boldsymbol{c}}_i||^2+\sum_{i=1}^{N}\sum_{k=1}^{K}(\gamma_{ik}-\gamma_{ik}^2)+\sum_{k=1}^{K}(1-\alpha_k)\log(\tilde{\gamma}_k). \tag{7.5}$$

Note, that there are parallels that can be drawn between the proposed architecture and IDEC [115], as the latter also makes use of an autoencoder framework with an additional clustering loss. Besides having a different loss, IDEC nests a single dense layer to the code of the DAE instead of an autoencoder.

## 7.3 Implementation

### 7.3.1 Averaging Epoch

The optimizer updates the positions of the centroids given the current batch. The small size of the latter causes some dispersion of the intermediate centroids. Hence, choosing the final centroids only based on the last iteration is sub-optimal.

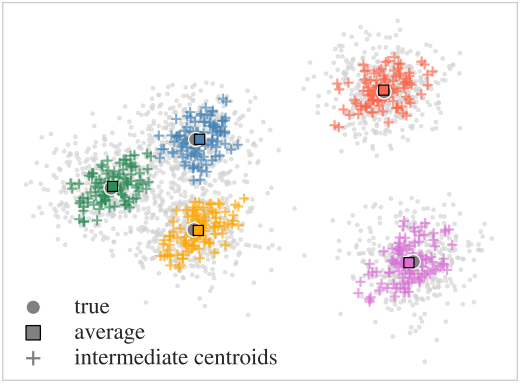



Figure 7.2: The intermediate centroids of the last epoch are spread, whereas their averages almost match the true centroids.

The phenomenon is illustrated for CM in Figure 7.2. The data consists of $N = 2{,}000$ points in $\mathbb{R}^2$ drawn from a mixture of five bi-variate Gaussians ($K = 5$). After standardizing the data, a CM is trained in mini-batches of size 20 over 100 epochs using stochastic gradient descent with a learning rate of 0.01, a momentum of 0.95, and a concentration equal to $\mathbf{5}_K$.

The dispersion of the centroids after each iteration of the last epoch (crosses) is important. On the other hand, the average positions over the last epoch (squares) provide a good approximation of the true centers (circles). The same phenomenon appears for C-NET. Therefore, implementations of both networks contain one extra epoch to compute the average position of the individual centroids over the last iterations. It is included in any subsequent computation.

### 7.3.2 Initialization and Pre-Training

Pre-training a deep autoencoder has advantages [26] and disadvantages [181]. In the case of clustering, it has been shown that it can harm the results [240]. However, several prior approaches exist, such as the two baselines DEC and IDEC, that report promising results when pre-training their deep autoencoder as stacked denoising autoencoder (SDA) and denoising autoencoder, respectively [245]. To take this discussion further, we propose a pre-training scheme for each model and evaluate their benefits in Section 7.4.

**Clustering Module** The clustering module can be pre-trained using initialization algorithms for $k$-means, such as *k-means++* [13] (k++). The centroids found by the initializer are used as column-vector of the decoder's weights, $\boldsymbol{W}_{\text{dec}}$. The pseudo-inverse of this matrix serves as the encoder's weights, $\boldsymbol{W}_{\text{enc}}$. The respective biases vectors are initialized using Glorot normal initializer [108].

**Clustering Network** In the case of C-Net, we prefer a more straightforward approach that has proven faster and more stable. There are two networks to pre-train: the deep autoencoder and the clustering module. The former is first pre-trained in an end-to-end fashion, i.e. without stacking or denoising, using the reconstruction loss only. The data-set is then encoded using the DAE and used to pre-train the CM. The module is initialized using $k$-means++ and optimized over as many epochs as the DAE using the genuine loss function (Equation 7.4).

## 7.4 Experiments

In this section, we compare the CM (as a sanity check) and the proposed C-Net on three real-world data-sets to $k$-means, GMM, and four current state-of-the-art baselines.

### 7.4.1 Experimental Setup

**Data** We focus on some representative and much used benchmark data-sets in the deep clustering context:
**MNIST** contains 70,000 handwritten images of the digits 0 to 9 [162]. The images are grayscale with the digits centered in the $28 \times 28$ images. The data-set is normalized before processing.
**USPS** consists of 9,298 gray-scale images of handwritten digits with size of 16x16 pixels. Similar to MNIST it contains 10 (slightly) unbalanced classes, digits 0 to 9.
**Reuters10k**, here abbreviated **R10K**, consists of 800,000 news stories that have been manually categorized into a category tree [167]. The data-set was pre-processed as in [115] to return a subset of 10,000 random samples embedded into a 2,000-dimensional space and distributed over 4 (highly) unbalanced categories.

**Baselines** We select as baselines recent deep clustering models that are not tailored to a specific domain. Since our method is centroid-based, we select DEC [254] and its extension IDEC [115]. We also include DCN [256] as its performance appears

competitive with the two first baselines. For completeness, we also include IIC [139] as a recent non-centroid-based model[1]. The authors explicitly claim that their model is not specialized to computer vision and that their loss can simply be *plugged* into any model [2].

**Optimization** Following the settings used in [254, 115], the encoder of the DAE is made of four layers $d - 500 - 500 - 2000 - 10$ [127], where $d$ represents the input dimension. The decoder mirrors the encoder. All activation functions are relu. For all models and data-sets, the batch-size $B$ is fixed to 256, the pre-training and training lasts respectively 200 and 2,000 epochs. In the case of pre-training, the DAE is optimized in an end-to-end fashion using SGD with a momentum of 0.9 (except for R10K where it is 0). The learning rate starts at 0.1 , and is divided by 10 every 20,000 iterations, For MNIST, the optimizations of CM and C-Net are performed using the Adam [144] optimizer (learning rate $=$ 0.001, $\beta_1 =$ 0.99, $\beta_2 = 0.999$, $\epsilon = 0.1$). Following the example of DEC and IDEC, an SGD with Nesterov acceleration is chosen for USPS and R10K, with momentum of 0.9 and 0 respectively.

DEC and IDEC rely on the same configuration for MNIST, and on an SGD without Nesterov acceleration, a learning rate of 0.01 and a momentum of 0.9 for USPS and R10K. These settings returned similar performance to those reported in [254, 115] and [256]. The target distribution is updated every 140, 30, and 20 iterations for MNIST, USPS, and R10K, respectively.

IIC is trained without auxiliary over-clustering head using only the architecture of the encoder of C-Net to which we add a softmax-activated layer with as many units as the number of clusters. Accordingly to the paper [139], the model is tested only with random initialization and optimized using Adam with a learning rate of $10^{-4}$. At each iteration, each instance is paired with five copies distorted with Gaussian noise with a standard deviation of 0.15.

The implementation of DCN[3] is configured as in the original paper. Regarding $k$-means and GMMs, we use the scikit-learn implementations [209] with up to 200 iterations and $k$-means++ initialization.

[1] `github.com/xu-ji/IIC`

[2] We also experimented with DA-GMM [268] but obtained erratic and unstable behavior. We finally refrained from including the results.

[3] `github.com/boyangumn/DCN-New`

Table 7.1: Comparison of clustering performance in terms of mean ARI (×100) with standard deviation and of the best run out of ten.

| Method | MNIST | | USPS | | R10K | |
|---|---|---|---|---|---|---|
| | avg. ± sd. | best | avg. ± sd. | best | avg. ± sd. | best |
| $k$-means | $36.4 \pm 1.8$ | 39.5 | $53.2 \pm 2.5$ | 57.2 | $27.0 \pm 11.6$ | 61.1 |
| GMM | $22.4 \pm 1.7$ | 25.5 | $36.6 \pm 3.5$ | 39.9 | $27.1 \pm 11.6$ | 61.2 |
| DEC+rand | $18.3 \pm 9.9$ | 30.4 | $28.0 \pm 3.7$ | 34.0 | $11.9 \pm 5.4$ | 19.5 |
| IDEC+rand | $22.7 \pm 4.2$ | 29.3 | $32.6 \pm 15.8$ | 56.9 | $15.1 \pm 4.0$ | 23.6 |
| DCN+rand | $32.0 \pm 2.1$ | 36.5 | $38.0 \pm 15.0$ | 51.0 | $8.3 \pm 3.3$ | 14.1 |
| DEC+pre | $75.6 \pm 2.3$ | 80.4 | $60.8 \pm 5.2$ | 66.9 | $\mathbf{58.7 \pm 4.0}$ | **65.5** |
| IDEC+pre | $77.3 \pm 1.5$ | 81.1 | $60.8 \pm 4.8$ | 67.3 | $55.0 \pm 3.5$ | 62.1 |
| DCN+pre | $78.5 \pm 0.2$ | 78.8 | $61.8 \pm 1.7$ | 66.8 | $32.6 \pm 5.2$ | 35.1 |
| IIC+rand | $40.5 \pm 6.4$ | 54.3 | $48.6 \pm 5.3$ | 56.3 | $17.6 \pm 5.0$ | 25.6 |
| CM+rand | $34.8 \pm 4.5$ | 40.3 | $49.3 \pm 6.2$ | 58.4 | $1.9 \pm 1.2$ | 3.6 |
| CM+pre | $40.7 \pm 0.3$ | 41.2 | $58.5 \pm 0.7$ | 59.9 | $23.2 \pm 1.9$ | 25.4 |
| C-Net+rand | $78.0 \pm 8.1$ | **92.1** | $56.3 \pm 2.8$ | 63.4 | $31.5 \pm 8.9$ | 45.5 |
| C-Net+pre | $\mathbf{83.5 \pm 1.6}$ | 88.1 | $\mathbf{69.2 \pm 2.1}$ | **72.4** | $51.3 \pm 6.6$ | 62.6 |

### 7.4.2 Results

Clustering performance in terms of mean ARI (×100) are reported on Table 7.1. Models are tested with and without pre-training, indicated by *+rand* and *+pre*, respectively. The highest, statistically significant score for each data-set is marked in bold face.

The scores of the three centroid-based baselines are equivalent to those reported in their respective papers. The results highlight that these models rely heavily on the pre-training of the autoencoder. Indeed, when randomly initialized, they are all outperformed by $k$-means on each data-set.

IIC returns deceiving scores, which are certainly caused by the choice of the network and noise. Indeed, a ResNet-based IIC with five auxiliary networks can cluster MNIST with an accuracy (true positives divided by sample size) of up to 99.2%. Such a score does not translate into the ARIs indicated in Table 7.1, but they highlight that the loss function of IIC requires cumbersome treatment of the features to give good results.

The vanilla clustering module performs similarly to $k$-means or GMM depending on the two first data-sets, as we have conjectured. For R10K, CM+rand failed and

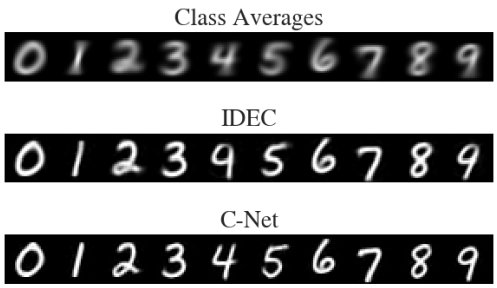



Figure 7.3: Centroids mapped back to image space for IDEC and C-Net. The first row displays the means of each class.

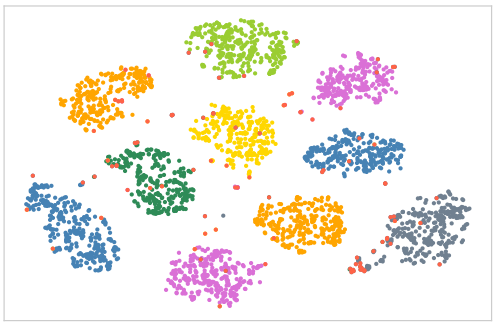


Figure 7.4: t-SNE representation of the embedding learned by C-Net on MNIST. The misclassified data-points are indicated in red.

CM+pre performs worse than the baselines. This can partly be contributed to the symmetric Dirichlet prior and the fact that the dataset has the most imbalanced cluster distribution.Overall, the model benefits from the k++ initialization.

The pre-trained C-Net out-performs all the baselines on MNIST and USPS. Without pre-training, it performs on par with the pre-trained baselines. C-Net+pre does not top the comparison on the R10K dataset, but remains among the best three. The imbalanced classes in the dataset is here again certainly a cause.

Pre-training of CM and C-Net does improve the performance but also reduces model volatility for better or worse. Indeed, C-Net+rand achieved an ARI of 92.1% on MNIST, which is 15 points above its average score only thanks to a high standard deviation. On the downside, this volatility is a challenge to reproducibility.

### 7.4.3 Discussion

**Analysis of the Results** C-Net with random initialization and pre-tuned IDEC returned two of the best ARIs for the MNIST. Figure 7.3 shows the centroids of their best run. We use the decoder to map the centroids to the input space. We can observe that the centroids of C-Net produce clear images for each class, which align reasonably well with the washed-out average image of the respective classes (first row). On the other hand, IDEC's centroids for the 4 and 9 look both like 9's. A closer investigation of the errors showed that only approximately 50% of the 4's and 9's where clustered correctly.

To analyze the learned representations of C-Net and study the misclassifications, we perform a t-SNE dimensionality reduction [178] of the embedded MNIST, i.e., from 10 to 2 dimensions. The result of this analysis is illustrated in Figure 7.4 for a subset of

3,500 randomly chosen data-points. It can be observed that C-Net achieves a distinct grouping of the individual clusters. The points that were clustered incorrectly by our method are highlighted in red. They mostly lay on the space between the individual clusters. The cluster for 9's (bottom right) contains the most errors: $511 \approx 7.1\%$ of the 9s were assigned to a wrong cluster and the cluster for 9 contains 507 images of other numbers.

**Analysis of the Optimization** The evolution of the loss function and of the ARI during the optimization of CM and C-Net, with and without pre-training is shown in Figure 7.5. The benefit of pre-training the CM and C-Net appears clearly in the plots of the last column. The respective curves in terms of loss or ARI start with better values. The impact on the stability is better observed in the plots in terms of ARI (bottom). Except for C-Net+rand, convergence is reached before 200 epochs, and even earlier for CM (about 20 seems enough). Regarding C-Net+rand, the runs converging the fastest in terms of ARI do so in about 200 epochs, which is less than the $200 + 200$ epochs of the pre-training, on the other hand, it returns a worse clustering.

The two first columns focus on the two pre-training phases of C-Net: the pre-training of the autoencoder and of the clustering module, respectively. Recall that the CM is initialized with $k$-means++ before being pre-trained, hence the gap between plots of the first and second columns. The tuning of the AE greatly reduces the loss but has limited impact on the clustering. The inverse holds for the tuning of the

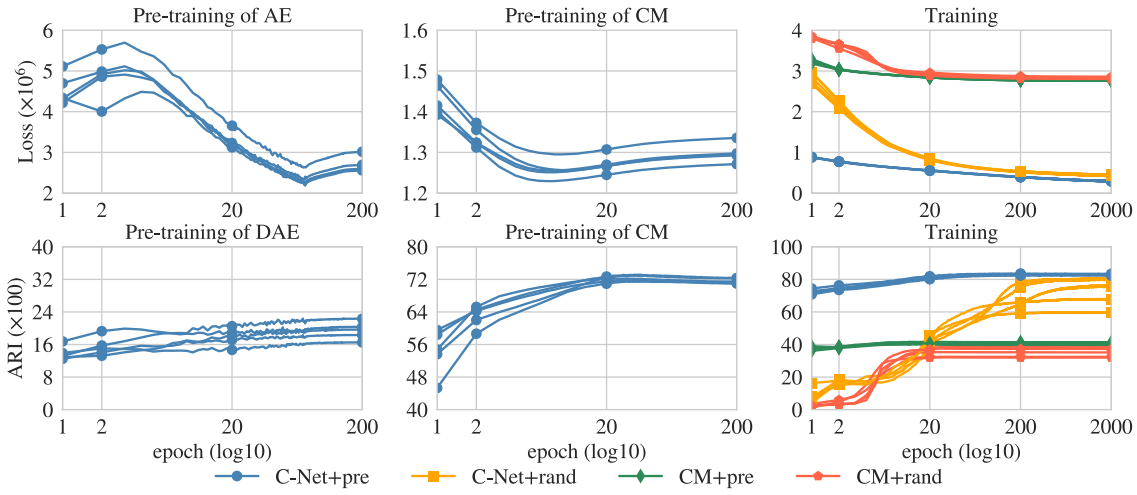



Figure 7.5: Evolution of the loss and ARI during optimization of CM and C-Net with and without pre-training. Five runs of each model are plotted. The epochs (x-axis) are represented in log-scale.

CM. Note that during both phases, the loss function reaches a minimum and then increases again. This behavior does not correlate with that of the ARI.

As a summary, CM needs less than 50 iterations to converge, be it for training or pre-training. As other studies [254] also pointed out, pre-training the AE does yield more consistent results but not necessarily a faster convergence, if the tuning epochs are included. Further studies are necessary to assess the impact of a shorter tuning of the AE, e.g., stop before convergence.

**Analysis of the Convergence** Figure 7.6 distributes the ten clusterings of MNIST reported in Table 7.1 with respect to their final value of the loss and the ARI. The dashed lines represent the best fitting regression line for each case. The corresponding $R^2$ scores and Pearsons' correlations are given under each plot.

The ARI and the value of the loss function correlate negatively and significant only for CM with random initialization. This result is a confirmation of the theoretical soundness of our model. If CM is pre-trained, the correlation is not significant anymore, but a tendency in the correct direction is present. For C-Net with and without pre-training, the slope of the regression line is positive which goes against the expected relationship. This suggests that the reconstruction of the DAE leads the value of the loss and steers the model toward a local optima. This claim is further encouraged by the fact that the best performing run of C-Net+pre returns the highest loss. Potential approach to reduce this effect caused by the inconsistent optimization goals is the introduction of additional regularization techniques, i.e. drop-out, or weighting of the terms in the loss function. However, both come at the cost of extra hyper-parameters or additional architecture choices.

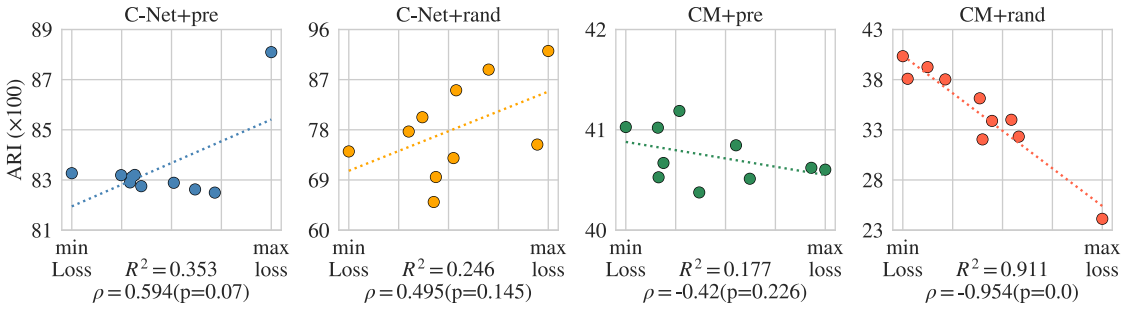



Figure 7.6: Scatter plot distributing each clusterings of MNIST reported in Table 7.1 with respect to the value of the loss function (x-axis) and ARI (y-axis). The regression lines are dashed, and $R^2$ scores and Pearson's coefficients are given below each plot.

**Analysis of the Concentration** To analyze the impact of the (symmetric) Dirichlet prior on the clustering performance of C-Net, we let $\alpha$ vary from 2 to 192. We experiment on the three previous data-sets without pre-training. The details of the optimization are the same as in Section 7.4.1. Figure 7.7 displays the variations of the ARI. The shades represent the standard error over ten runs, and the dashed curves the highest ARIs.

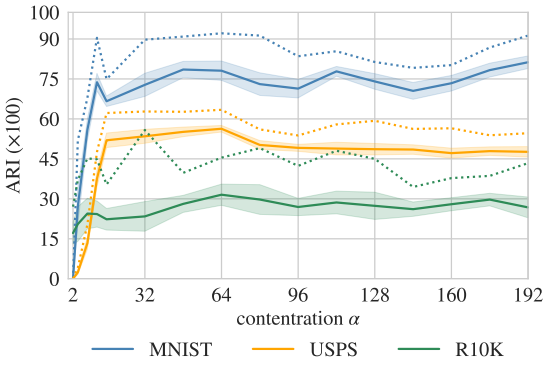



Figure 7.7: Robustness of C-Net with respect to the concentration hyper-parameter. The dashed curves represent the maximal ARI.

The three curves present a similar shape. They increase abruptly and only then become more stable. For R10K, the initial jump is smaller and earlier. We can remark, though, that the maximal ARIs for MNIST tends closer to the average curve as the concentration increases.

**Analysis of the Relaxation** The loss function of the clustering module nested in the C-Net is relaxed: the regularization term $||\boldsymbol{\mu}_k||$ is discarded. To empirically justify this modification, the same experiment is run for C-Net with the regularization term in the loss function (with and without pre-training). The results in terms of ARI are reported in Table 7.2.

The results confirm the negative impact of the regularization on the clustering performance. The regularization yields lower average ARIs in every settings, except for C-Net+rand on R10K. For MNIST and USPS, a randomly initialized C-Net returns ARIs more than 50% smaller than without the regularization. For these two data-set, the standard deviations of C-Net are larger than those reported in Table 7.1. In case of R10K, the regularized model performs on par with the non-regularized one.

Table 7.2: Clustering performance in terms of mean ARI (×100) of a regularized C-Net with and without pre-training.

| | reg. C-Net+rand | | reg. C-Net+pre | |
|---|---|---|---|---|
| | avg. ± sd. | best | avg. ± sd. | best |
| MNIST | $35.1 \pm 2.9$ | 39.2 | $62.4 \pm 2.2$ | 65.1 |
| USPS | $26.4 \pm 4.4$ | 32.5 | $59.9 \pm 4.7$ | 66.6 |
| R10K | $37.0 \pm 4.7$ | 45.2 | $30.0 \pm 5.4$ | 38.8 |

## 7.5 Conclusion

We presented an end-to-end approach to address centroid-based clustering tasks with deep learning based on a theoretically proven loss function. We first transformed the objective function of an isotropic GMM into a loss function for an autoencoder. Second, we proposed two networks. The Clustering Module (CM) consisted of a two-layer autoencoder and was optimized with the transformed loss function of a GMM, followed by an averaging epoch to obtain accurate centroids. The Clustering Network (C-Net) extended CM to deep architectures and grounded on nesting CM within any autoencoder (AE). The associated loss function balanced the reconstruction losses of the AE and a relaxation of the CM loss.

Empirical evaluations on real-world data-sets confirmed that CM performs similar to $k$-means and GMM. By contrast, C-Net outperformed existing deep-learning models on almost all problems. The analysis revealed that our models benefit from pre-training, but also perform well without. This is in contrast to existing centroid-based clustering approaches that all highly depended on pre-training.

# Chapter 8

# Conclusion

In this thesis, we advanced the fields of machine learning and data-mining, with a focus on online behaviors analysis (OBA). We presented models for different types of representations of user sessions, with an emphasis on interpretability. Each model is used to analyze the *mBook* data which yielded to the discovery of several new insights into the relationship between pupils' use of the medium and their performance and motivation in History.

We have shown that the computation of the frame of a data-set boils down to a non-negative least-squares problem that can be solved efficiently using existing techniques. This new view of the problem has led to increased efficiency and has made archetypal analyses in large dimensions possible. Our approximation, Frame-AA, combined with the divide-and-conquer strategy, brings additional speed-up to the computations with a minimal loss of precision. We went further and gave an example for OBA. While a naive approach only links the time spent on galleries to motivation, an archetypal analysis uncovered more correlations. Namely, high motivation comes with more time spent on rich content, and less time spent on less informative content. However, the relatively coarse representation of the behaviors lead to a paradoxical correlation between high use of information boxes and low motivation. We consider, however, this anomaly as a positive result since it argues in favor of modeling sessions as sequences.

Consequently, we have studied the sequences of chapters and categories of the pages visited. Our mixture of Markov chains nested with a temporal model detected weekly and daily connection patterns and related them to navigation profiles. This approach gave us a first indication of the teacher's influence on student behavior. Unfortunately, the complexity of the model was already too high for information-based model selection. In order to investigate the events themselves, we proposed a model combining Dirichlet processes and a mixture of Markov chains: the infinite

mixtures of Markov chains. A key element of the model is the use of a degree $k$-weak limit approximation to lighten the calculations. Our approach successfully managed the large gap between the number of events tracked and the relatively small size of the data-set. An analysis of the mBook data revealed that scrolling behaviors, as well as the probability of specific transitions between scroll events, can be used as indicators of several psychometric indicators.

The Markov condition places much emphasis on the transition between events, which can hinder longer-term dependencies. To address this, we took a radically new perspective on behavioral analysis. We modeled sessions as spatio-temporal trajectories within the page graph. To pursue this idea, we developed a theoretical framework for spatio-temporal similarity measures that covers most of the existing measures and formalizes the desirable properties. We also constructed the first similarity measure, $\Delta_{\mathrm{KL}}$, which satisfies them all. A thorough empirical evaluation of our measure in terms of clustering and prediction puts it on par or better with existing measures. We used $\Delta_{\mathrm{KL}}$ to reveal that the dispersion of trajectories within a class-group is an indicator of pupils' activity. Our analysis revealed the moderating role of the teacher in the correlation between online behavior and performance.

Our last results also act as an opening and show future directions for our research. We developed a theoretically well-founded deep clustering model that can mimic $k$-means while exploiting the representation capabilities offered by deep neural networks. In this work, we relied only on fully connected layers. However, other architectures are in the scope of further developments, such as convolutional layers, recurrent networks, and long short-term memory cells. This would allow to do just like mixture models but with neural networks: jointly modeling and clustering online behaviors.

# Bibliography


[1] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation, 1965.

[2] P. K. Agarwal, R. B. Avraham, H. Kaplan, and M. Sharir. Computing the discrete fréchet distance in subquadratic time. *SIAM Journal on Computing*, 43(2):429–449, 2014.

[3] M. Agosti, F. Crivellari, and G. Di Nunzio. Web log analysis: a review of a decade of studies about information acquisition, inspection and interpretation of user interaction. *Data Mining and Knowledge Discovery*, pages 1–34, 2011.

[4] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, 1993.

[5] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the eleventh international conference on data engineering*, pages 3–14. IEEE, 1995.

[6] H. Akaike. A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer, 1974.

[7] H. Alt and M. Godau. Measuring the resemblance of polygonal curves. In *Proceedings of the eighth annual symposium on Computational geometry*, pages 102–109. ACM, 1992.

[8] C. R. Anderson, P. Domingos, and D. S. Weld. Relational markov models and their application to adaptive web navigation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[9] Aristotle. *Nicomachean Ethics*. Cambridge Texts in the History of Philosophy. Cambridge University Press, 2000.

[10] M. Armentano and A. Amandi. Modeling sequences of user actions for statistical goal recognition. *User Modeling and User-Adapted Interaction*, 22(3):281–311, 2012.

[11] R. Arora, M. Gupta, A. Kapila, and M. Fazel. Clustering by left-stochastic matrix factorization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 761–768, 2011.

[12] R. Arora, M. R. Gupta, A. Kapila, and M. Fazel. Similarity-based clustering by left-stochastic matrix factorization. *Journal of Machine Learning Research*, 14(1):1715–1746, 2013.

[13] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[14] S. Atev, O. Masoud, and N. Papanikolopoulos. Learning traffic patterns at intersections by spectral clustering of motion trajectories. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4851–4856. IEEE, 2006.

[15] R. Baker and G. Siemens. *Educational data mining and learning analytics*. Cambridge Handbook of the Learning Sciences, 2014.

[16] R. Baker and K. Yacef. The state of educational data mining in 2009: A review and future visions. 1(1):3–17, 2009.

[17] S. Balakrishnan, M. J. Wainwright, B. Yu, et al. Statistical guarantees for the em algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120, 2017.

[18] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[19] C. Bauckhage, H. Kersting, C. Thurau, et al. Archetypal analysis as an autoencoder. In *Workshop New Challenges in Neural Computation 2015*, page 8. Citeseer, 2015.

[20] L. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a markov process. *Inequalities*, 3:1–8, 1972.

[21] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[22] M. A. Bautista, A. Sanakoyeu, E. Tikhoncheva, and B. Ommer. Cliquecnn: Deep unsupervised exemplar learning. In *Advances in Neural Information Processing Systems*, pages 3846–3854, 2016.

[23] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden markov model. In *Advances in neural information processing systems*, pages 577–584, 2002.

[24] R. Begleiter, R. El-Yaniv, and G. Yona. On prediction using variable order markov models. *Journal of Artificial Intelligence Research*, 22:385–421, 2004.

[25] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing user behavior in online social networks. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*, pages 49–62, 2009.

[26] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.

[27] D. Bengs, U. Brefeld, and U. Kröhne. Adaptive item selection under matroid constraints. *Journal of Computerized Adaptive Testing*, 6(2):15–36, 2018.

[28] C. H. Bennett, P. Gács, M. Li, P. M. Vitányi, and W. H. Zurek. Information distance. *IEEE Transactions on information theory*, 44(4):1407–1423, 1998.

[29] M. Berger. *Géométrie*, volume 4. Cedic, 1977.

[30] L. Bergroth, H. Hakonen, and T. Raita. A survey of longest common subsequence algorithms. In *Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000*, pages 39–48. IEEE, 2000.

[31] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, pages 359–370. AAAI Press, 1994.

[32] P. C. Besse, B. Guillouet, J.-M. Loubes, and F. Royer. Destination prediction by trajectory distribution-based model. *IEEE Transactions on Intelligent Transportation Systems*, (99):1–12, 2017.

[33] J. Bian, D. Tian, Y. Tang, and D. Tao. A survey on trajectory clustering analysis. *arXiv preprint arXiv:1802.06971*, 2018.

[34] D. Billsus and M. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2):147–180, 2000.

[35] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[36] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[37] J. M. Blumenthal, J. L. Solomon, C. D. Bell, T. J. Austin, G. Ebanks-Petrie, M. S. Coyne, A. C. Broderick, and B. J. Godley. Satellite tracking highlights the need for international cooperation in marine turtle management. *Endangered Species Research*, 2:51–61, 2006.

[38] J. Borges and M. Levene. Evaluating variable-length markov chain models for analysis of user web navigation sessions. *IEEE Transactions on Knowledge and Data Engineering*, pages 441–452, 2007.

[39] M. S. Boroujeni and P. Dillenbourg. Discovery and temporal analysis of mooc study patterns. *Journal of Learning Analytics*, 6(1):16–33, 2019.

[40] A. Boubekki, S. Jain, and U. Brefeld. Mining user trajectories in electronic text books. *International Educational Data Mining Society*, 2018.

[41] A. Boubekki, U. Kröhne, F. Goldhammer, W. Schreiber, and U. Brefeld. Toward data-driven analyses of electronic text books. In *EDM*, pages 592–593, 2015.

[42] A. Boubekki, U. Kröhne, F. Goldhammer, W. Schreiber, and U. Brefeld. Data-driven analyses of electronic text books. In *Solving Large Scale Learning Tasks. Challenges and Algorithms*, pages 362–376. Springer, 2016.

[43] N. Bourbaki. *Eléments de mathématiques: topologie générale*. Hermann, 1958.

[44] T. Boyer, J. Antonov, O. Baranova, C. Coleman, H. Garcia, A. Grodsky, D. Johnson, R. Locarnini, A. Mishonov, T. O'Brien, C. Paver, J. Reagan, D. Seidov, I. Smolyar, and Z. M.M. *World Ocean Database 2013.* NOAA Printing Office, 2013.

[45] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[46] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[47] T. Brijs, G. Swinnen, K. Vanhoof, and G. Wets. Using association rules for product assortment decisions: A case study. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 254–260, 1999.

[48] A. Brondsted. *An introduction to convex polytopes*, volume 90. Springer Science & Business Media, 2012.

[49] M. Broniatowski, G. Celeux, and J. Diebolt. Reconnaissance de mélanges de densités par un algorithme d'apprentissage probabiliste. *Data analysis and informatics*, 3:359–373, 1983.

[50] K. Buchin, M. Buchin, and J. Gudmundsson. Constrained free space diagrams: a tool for trajectory analysis. *International Journal of Geographical Information Science*, 24(7):1101–1125, 2010.

[51] K. Buchin, M. Buchin, M. Van Kreveld, and J. Luo. Finding long and similar parts of trajectories. *Computational Geometry*, 44(9):465–476, 2011.

[52] P. Bühlmann, A. J. Wyner, et al. Variable length markov chains. *The Annals of Statistics*, 27(2):480–513, 1999.

[53] K. P. Burnham and D. R. Anderson. A practical information-theoretic approach. *Model selection and multimodel inference, 2nd ed. Springer, New York*, 2002.

[54] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Visualization of navigation patterns on a web site using model-based clustering. In *KDD*, pages 280–284. Citeseer, 2000.

[55] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

[56] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[57] G. Casella and E. I. George. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[58] G. Celeux and G. Govaert. A classification em algorithm for clustering and two stochastic versions. *Computational statistics & Data analysis*, 14(3):315–332, 1992.

[59] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan. Deep adaptive image clustering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5880–5888. IEEE, 2017.

[60] L. Chen, M. Lv, and G. Chen. A system for destination and future route prediction based on trajectory mining. *Pervasive and Mobile Computing*, 6(6):657–676, 2010.

[61] L. Chen and R. Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 792–803. VLDB Endowment, 2004.

[62] L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502. ACM, 2005.

[63] M.-Y. Chen, A. Kundu, and J. Zhou. Off-line handwritten word recognition using a hidden markov model type stochastic network. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (5):481–496, 1994.

[64] Y. Chen, M. Kapralov, J. Canny, and D. Y. Pavlov. Factor modeling for advertisement targeting. In *Advances in Neural Information Processing Systems*, pages 324–332, 2009.

[65] Y. Chen, J. Mairal, and Z. Harchaoui. Fast and robust archetypal analysis for representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1478–1485, 2014.

[66] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[67] Y.-J. Chu. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400, 1965.

[68] M. Cocea and S. Weibelzahl. Cross-system validation of engagement prediction from log files. In *European Conference on Technology Enhanced Learning*, pages 14–25. Springer, 2007.

[69] R. Cooley, B. Mobasher, and J. Srivastava. Web mining: Information and pattern discovery on the world wide web. In *Proceedings ninth IEEE international conference on tools with artificial intelligence*, pages 558–567. IEEE, 1997.

[70] R. E. Crossler, A. C. Johnston, P. B. Lowry, Q. Hu, M. Warkentin, and R. Baskerville. Future directions for behavioral information security research. *computers & security*, 32:90–101, 2013.

[71] A. Cutler and L. Breiman. Archetypal analysis. *Technometrics*, 36(4):338–347, 1994.

[72] M. F. Dacrema, P. Cremonesi, and D. Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 101–109, 2019.

[73] C. d'Amato, N. Fanizzi, and F. Esposito. A semantic similarity measure for expressive description logics. *arXiv preprint arXiv:0911.5043*, 2009.

[74] R. Das and I. Turkoglu. Creating meaningful data from web logs for improving the impressiveness of a website by using path analysis method. *Expert Systems with Applications*, 36(3):6635–6644, 2009.

[75] R. Das and I. Türkoglu. Extraction of interesting patterns through association rule mining for improvement of website usability. *Istanbul University-Journal of Electrical & Electronics Engineering*, 9:18, 2010.

[76] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer. Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 273–274. International World Wide Web Conferences Steering Committee, 2016.

[77] A. De Brébisson, É. Simon, A. Auvolat, P. Vincent, and Y. Bengio. Artificial neural networks applied to taxi destination prediction. *arXiv preprint arXiv:1508.00021*, 2015.

[78] T. Delacroix, A. Boubekki, P. Lenca, and S. Lallich. Constrained independence for detecting interesting patterns. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE, 2015.

[79] C. DeMars. *Item response theory*. Oxford University Press, 2010.

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

[81] M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *ACM transactions on internet technology (TOIT)*, 4(2):163–184, 2004.

[82] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98. ACM, 2003.

[83] U. Dick and U. Brefeld. Learning to rate player positioning in soccer. *Big Data*, 7(1):71–82, 2019.

[84] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[85] C. H. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):45–55, 2010.

[86] S. Dodge, R. Weibel, and P. Laube. Trajectory similarity analysis in movement parameter space. *Plymouth, UK: Proceedings of GISRUK*, pages 27–29, 2011.

[87] M.-P. Dubuisson and A. K. Jain. A modified hausdorff distance for object matching. In *Proceedings of 12th international conference on pattern recognition*, pages 566–568. IEEE, 1994.

[88] J. H. Dulá and R. V. Helgason. A new procedure for identifying the frame of the convex hull of a finite collection of points in multidimensional space. *European Journal of Operational Research*, 92(2):352–367, 1996.

[89] J. H. Dulá and F. J. López. Competing output-sensitive frame algorithms. *Computational Geometry*, 45(4):186–197, 2012.

[90] T. Ebesu, B. Shen, and Y. Fang. Collaborative memory network for recommendation systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 515–524, 2018.

[91] J. Edmonds. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240, 1967.

[92] T. Eiter and H. Mannila. Computing discrete fréchet distance. Technical report, Citeseer, 1994.

[93] D. M. Ennis, J. J. Palen, and K. Mullen. A multidimensional stochastic theory of similarity. *Journal of Mathematical Psychology*, 32(4):449–465, 1988.

[94] L. Espín Noboa, F. Lemmerich, P. Singer, and M. Strohmaier. Discovering and characterizing mobility patterns in urban spaces: a study of manhattan taxi data. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 537–542. International World Wide Web Conferences Steering Committee, 2016.

[95] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. *Fast subsequence matching in time-series databases*, volume 23. ACM, 1994.

[96] L. Faucon, L. Kidzinski, and P. Dillenbourg. Semi-markov model for simulating mooc students. *International Educational Data Mining Society*, 2016.

[97] M. P. Fay and M. A. Proschan. Wilcoxon-mann-whitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics surveys*, 4:1, 2010.

[98] T. S. Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

[99] M. M. Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 22(1):1–72, 1906.

[100] E. Frentzos, K. Gratsias, N. Pelekis, and Y. Theodoridis. Algorithms for nearest neighbor search on moving object trajectories. *Geoinformatica*, 11(2):159–193, 2007.

[101] E. Frentzos, K. Gratsias, and Y. Theodoridis. Index-based most similar trajectory search. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 816–825. IEEE, 2007.

[102] E. Fuchs, I. Niehaus, A. Stoletzki, et al. *Das Schulbuch in der Forschung. Analysen und Empfehlungen für die Bildungspraxis*. Göttingen: V&R unipress, 2014.

[103] C. Geigle and C. Zhai. Modeling student behavior with two-layer hidden markov models. *JEDM— Journal of Educational Data Mining*, 9(1):1–24, 2017.

[104] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, 6:721–741, 1984.

[105] P. Geurts. Pattern extraction for time series classification. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 115–127. Springer, 2001.

[106] Z. Ghahramani and T. L. Griffiths. Infinite latent feature models and the indian buffet process. In *Advances in neural information processing systems*, pages 475–482, 2006.

[107] C. Giraud. *Introduction to high-dimensional statistics*. Chapman and Hall/CRC, 2014.

[108] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[109] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[110] K. C. Gowda and G. Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern recognition*, 10(2):105–112, 1978.

[111] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250, 2010.

[112] T. L. Griffiths, M. I. Jordan, J. B. Tenenbaum, and D. M. Blei. Hierarchical topic models and the nested chinese restaurant process. In *Advances in neural information processing systems*, pages 17–24, 2004.

[113] E. Guàrdia-Sebaoun, V. Guigue, and P. Gallinari. Latent trajectory modeling: A light and efficient way to introduce time in recommender systems. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 281–284. ACM, 2015.

[114] A. Gunawardana and W. Byrne. Convergence theorems for generalized alternating minimization procedures. *Journal of machine learning research*, 6(Dec):2049–2073, 2005.

[115] X. Guo, L. Gao, X. Liu, and J. Yin. Improved deep embedded clustering with local structure preservation. In *International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 1753–1759, 2017.

[116] P. Haeusser, J. Plapp, V. Golkov, E. Aljalbout, and D. Cremers. Associative deep clustering: Training a classification network with no labels. In *German Conference on Pattern Recognition*, pages 18–32. Springer, 2018.

[117] P. Haider, L. Chiarandini, and U. Brefeld. Discriminative clustering for market segmentation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 417–425. ACM, 2012.

[118] P. Haider, L. Chiarandini, U. Brefeld, and A. Jaimes. Contextual models for user interaction on the web. In *ECML/PKDD Workshop on Mining and Exploiting Interpretable Local Patterns (I-PAT)*, 2012.

[119] A. Hannak, P. Sapiezynski, A. Molavi Kakhki, B. Krishnamurthy, D. Lazer, A. Mislove, and C. Wilson. Measuring personalization of web search. In *Proceedings of the 22nd international conference on World Wide Web*, pages 527–538, 2013.

[120] C. Hansen, C. Hansen, N. Hjuler, S. Alstrup, and C. Lioma. Sequence modelling for analysing student interaction with educational systems. *International Educational Data Mining Society*, 2017.

[121] D. V. Hansen and P.-M. Poulain. Quality control and interpolations of wocetoga drifter data. *Journal of Atmospheric and Oceanic Technology*, 13(4):900–909, 1996.

[122] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[123] F. Hausdorff. *Mengenlehre*. Walter de Gruyter Berlin, 1927.

[124] G. C. Hays, J. A. Mortimer, D. Ierodiaconou, and N. Esteban. Use of long-distance migration patterns of an endangered species to inform conservation planning for the world's largest marine protected area. *Conservation Biology*, 28(6):1636–1644, 2014.

[125] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[126] A. Hershkovitz and R. Nachmias. Developing a log-based motivation measuring tool. In *Proceedings of the International Conference on Educational Data Mining*, 2008.

[127] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[128] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[129] C.-C. Hsu and C.-W. Lin. Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data. *IEEE Transactions on Multimedia*, 20(2):421–429, 2018.

[130] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *International Conference on Machine Learning*, pages 1558–1567, 2017.

[131] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[132] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, 15(9):850–863, 1993.

[133] H. Ishwaran and M. Zarepour. Markov chain monte carlo in approximate dirichlet and beta two-parameter process hierarchical models. *Biometrika*, 87(2):371–390, 2000.

[134] H. Ishwaran and M. Zarepour. Exact and approximate sum representations for the dirichlet process. *Canadian Journal of Statistics*, 30(2):269–283, 2002.

[135] B. J. Jansen. Understanding user-web interactions via web analytics. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1):1–102, 2009.

[136] H. Jeong and G. Biswas. Mining student behavior models in learning-by-teaching environments. In *Educational data mining 2008*, 2008.

[137] H. Jeong, H. J. Chang, and J. Y. Choi. Modeling of moving object trajectory by spatio-temporal learning for abnormal behavior detection. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 119–123. IEEE, 2011.

[138] H. Jeong, A. Gupta, R. Roscoe, Wagster, G. J. Biswas, and D. Schwartz. Using hidden markov models to characterize student behaviors in learning-by-teaching environments. In *Intelligent Tutoring Systems*, 2008.

[139] X. Ji, J. F. Henriques, and A. Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9865–9874, 2019.

[140] I. N. Junejo, O. Javed, and M. Shah. Multi feature path modeling for video surveillance. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 716–719. IEEE, 2004.

[141] M. Kalli, J. E. Griffin, and S. G. Walker. Slice sampling mixture models. *Statistics and computing*, 21(1):93–105, 2011.

[142] M. Kampffmeyer, S. Løkse, F. M. Bianchi, L. Livi, A.-B. Salberg, and R. Jenssen. Deep divergence-based approach to clustering. *Neural Networks*, 113:91–101, 2019.

[143] J. Kay, N. Maisonneuve, K. Yacef, and O. Zaïane. Mining patterns of events in students' teamwork data. In *Proceedings of the ITS Workshop on Educational Data Mining*, 2006.

[144] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[145] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[146] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[147] J. S. Kinnebrew, K. M. Loretz, and G. Biswas. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *JEDM—Journal of Educational Data Mining*, 5(1):190–219, 2013.

[148] A. Kinoshita, A. Takasu, and J. Adachi. Real-time traffic incident detection using a probabilistic topic model. *Information Systems*, 54:169–188, 2015.

[149] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[150] K. Knauf, D. Memmert, and U. Brefeld. Spatio-temporal convolution kernels. *Machine learning*, 102(2):247–273, 2016.

[151] M. Köck and A. Paramythis. Activity sequence modelling and dynamic clustering for personalized e-learning. *User Modeling and User-Adapted Interaction*, 21(1-2):51–97, 2011.

[152] A. Körber, W. Schreiber, and A. Schöner. *Kompetenzen historischen Denkens: ein Strukturmodell als Beitrag zur Kompetenzorientierung in der Geschichtsdidaktik*, volume 2. ars una, 2007.

[153] J. A. Kraemer and D. Malver. Behavioral learning for interactive user security, Oct. 9 2012. US Patent 8,286,254.

[154] A. Krogh, B. Larsson, G. Von Heijne, and E. L. Sonnhammer. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes. *Journal of molecular biology*, 305(3):567–580, 2001.

[155] J. Krumm and E. Horvitz. Predestination: Inferring destinations from partial trajectories. In *International Conference on Ubiquitous Computing*, pages 243–260. Springer, 2006.

[156] B. Kulis and M. I. Jordan. Revisiting k-means: new algorithms via bayesian nonparametrics. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1131–1138. Omnipress, 2012.

[157] S. Kullback. *Information theory and statistics*. Courier Corporation, 1997.

[158] T. Lang and M. Rettenmeier. Understanding consumer behavior with recurrent neural networks. In *Workshop on Machine Learning Methods for Recommender Systems*, 2017.

[159] J. S. Larson, E. T. Bradlow, and P. S. Fader. An exploratory look at supermarket shopping paths. *International Journal of research in Marketing*, 22(4):395–414, 2005.

[160] L. J. Latecki, Q. Wang, S. Koknar-Tezel, and V. Megalooikonomou. Optimal subsequence bijection. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 565–570. IEEE, 2007.

[161] C. L. Lawson and R. J. Hanson. *Solving least squares problems*, volume 15. SIAM, 1995.

[162] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[163] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[164] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604. ACM, 2007.

[165] S.-L. Lee, S.-J. Chun, D.-H. Kim, J.-H. Lee, and C.-W. Chung. Similarity search for multidimensional data sequences. In *Proceedings of 16th International Conference on Data Engineering (Cat. No. 00CB37073)*, pages 599–608. IEEE, 2000.

[166] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[167] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

[168] M. Li, X. Chen, X. Li, B. Ma, and P. Vitányi. The similarity metric. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 863–872. Society for Industrial and Applied Mathematics, 2003.

[169] X. Li, M. Li, Y.-J. Gong, X.-L. Zhang, and J. Yin. T-desp: Destination prediction based on big trajectory data. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2344–2354, 2016.

[170] P. Liang and D. Klein. Online em for unsupervised models. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 611–619, 2009.

[171] B. Lin and J. Su. Shapes based trajectory queries for moving objects. In *Proceedings of the 13th annual ACM international workshop on Geographic information systems*, pages 21–30. ACM, 2005.

[172] D. Lin et al. An information-theoretic definition of similarity. In *Icml*, volume 98, pages 296–304. Citeseer, 1998.

[173] W. J. Linden, W. J. van der Linden, and C. A. Glas. *Computerized adaptive testing: Theory and practice*. Springer, 2000.

[174] S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[175] F.-J. Lopez. Generating random points (or vectors) controlling the percentage of them that are extreme in their convex (or positive) hull. *Journal of Mathematical Modelling and Algorithms*, 4(2):219–234, 2005.

[176] J. Lv, Q. Li, Q. Sun, and X. Wang. T-conv: A convolutional neural network for multi-scale taxi trajectory prediction. In *Big Data and Smart Computing (BigComp), 2018 IEEE International Conference on*, pages 82–89. IEEE, 2018.

[177] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther. Auxiliary deep generative models. In *33rd International Conference on Machine Learning (ICML 2016)*, 2016.

[178] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[179] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, number 14, pages 281–297. Oakland, CA, USA, 1967.

[180] A. Maheshwari, J.-R. Sack, K. Shahbaz, and H. Zarrabi-Zadeh. Fréchet distance with speed limits. *Computational Geometry*, 44(2):110–120, 2011.

[181] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[182] R. S. Mamon and R. J. Elliott. *Hidden Markov models in finance*, volume 4. Springer, 2007.

[183] E. Manavoglu, D. Pavlov, and C. L. Giles. Probabilistic user behavior models. In *Third IEEE International Conference on Data Mining*, pages 203–210. IEEE, 2003.

[184] U. Manber, A. Patel, and J. Robison. Experience with personalization of yahoo! *Communications of the ACM*, 43(8):35–39, 2000.

[185] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[186] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data mining and knowledge discovery*, 1(3):259–289, 1997.

[187] P.-F. Marteau. Time warp edit distance with stiffness adjustment for time series matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):306–318, 2009.

[188] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[189] D. R. Miller, T. Leek, and R. M. Schwartz. A hidden markov model information retrieval system. In *SIGIR*, volume 99, pages 214–221, 1999.

[190] B. Mobasher. Data mining for web personalization. In *The adaptive web*, pages 90–135. Springer, 2007.

[191] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.

[192] D. Mochihashi and E. Sumita. The infinite markov model. In *Advances in neural information processing systems*, pages 1017–1024, 2008.

[193] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 637–646. ACM, 2009.

[194] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas. Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1393–1402, 2013.

[195] B. Morris and M. Trivedi. Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 312–319. IEEE, 2009.

[196] B. T. Morris and M. M. Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE transactions on circuits and systems for video technology*, 18(8):1114–1127, 2008.

[197] M. Mørup and L. K. Hansen. Archetypal analysis for machine learning. In *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, pages 172–177. IEEE, 2010.

[198] K. P. Murphy. Hidden semi-markov models (hsmms). 2002.

[199] S. Naderivesal, L. Kulik, and J. Bailey. An effective and versatile distance measure for spatiotemporal trajectories. *Data Mining and Knowledge Discovery*, 33(3):577–606, 2019.

[200] M. Nanni and D. Pedreschi. Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, 27(3):267–289, 2006.

[201] A. Narayanan and V. Shmatikov. Robust de-anonymization of large datasets (how to break anonymity of the netflix prize dataset). *University of Texas at Austin*, 2008.

[202] R. M. Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[203] I. Olkin, J. W. Pratt, et al. Unbiased estimation of certain correlation coefficients. *The Annals of Mathematical Statistics*, 29(1):201–211, 1958.

[204] T. Ottmann, S. Schuierer, and S. Soundaralakshmi. Enumerating extreme points in higher dimensions. *Nordic Journal of Computing*, 8(2):179–192, 2001.

[205] S. Owega, G. J. Evans, R. E. Jervis, M. Fila, et al. Identification of long-range aerosol transport patterns to toronto via classification of back trajectories by cluster analysis and neural network techniques. *Chemometrics and Intelligent Laboratory Systems*, 83(1):26–33, 2006.

[206] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[207] K. Padberg-Gehle and C. Schneide. Trajectory-based computational study of coherent behavior in flows. *PAMM*, 17(1):11–14, 2017.

[208] G. Pagès and M. Briane. *Théorie de l'intégration. Cours et exercices. Licence et master de mathématiques*. Vuibert, 2004.

[209] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[210] N. Pelekis, I. Kopanakis, E. Kotsifakos, E. Frentzos, and Y. Theodoridis. Clustering trajectories of moving objects in an uncertain world. In *2009 Ninth IEEE international conference on data mining*, pages 417–427. IEEE, 2009.

[211] N. Pelekis, I. Kopanakis, G. Marketos, I. Ntoutsi, G. Andrienko, and Y. Theodoridis. Similarity search in trajectory databases. In *14th International Symposium on Temporal Representation and Reasoning (TIME'07)*, pages 129–140. IEEE, 2007.

[212] J. Qiqi, T. Chuan-Hoo, P. Chee Wei, and K. K. Wei. Using sequence analysis to classify web usage patterns across websites. In *Proceedings of the 45th Hawaii International Conference on System Science (HICSS)*, pages 3600–3609, 2012.

[213] L. R. Rabiner, B.-H. Juang, and J. C. Rutledge. *Fundamentals of speech recognition*, volume 14. PTR Prentice Hall Englewood Cliffs, 1993.

[214] H. Raiffa and R. Schlaifer. *Applied statistical decision theory.* Division of Research, Graduate School of Business Adminitration, Harvard ..., 1961.

[215] C. E. Rasmussen. The infinite gaussian mixture model. In *Advances in neural information processing systems*, pages 554–560, 2000.

[216] G. D. P. Regulation. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46. *Official Journal of the European Union (OJ)*, 59(1-88):294, 2016.

[217] C. Romero and S. Ventura. Educational data mining: A survey from 1995 to 2005. *Expert systems with applications*, 33(1):135–146, 2007.

[218] M. Rosenbaum, A. B. Tsybakov, et al. Sparse recovery under matrix uncertainty. *The Annals of Statistics*, 38(5):2620–2651, 2010.

[219] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007.

[220] A. Rossi, G. Barlacchi, M. Bianchini, and B. Lepri. Modelling taxi drivers' behaviour for the next destination prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[221] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[222] S. Salmeron-Majadas, O. C. Santos, and J. G. Boticario. Exploring indicators from keyboard and mouse interactions to predict the user affective state. In *Educational Data Mining 2014*, 2014.

[223] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[224] S. Santini and R. Jain. Similarity measures. *IEEE Transactions on pattern analysis and machine Intelligence*, 21(9):871–883, 1999.

[225] W. Schreiber, A. Schöner, and F. Sochatzy. *Analyse von Schulbüchern als Grundlage empirischer Geschichtsdidaktik*. Kohlhammer Verlag, 2013.

[226] G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[227] O. Semerci, A. Gruson, C. Edwards, B. Lacker, C. Gibson, and V. Radosavljevic. Homepage personalization at spotify. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 527–527. ACM, 2019.

[228] S. Seth and M. J. Eugster. Probabilistic archetypal analysis. *Machine learning*, 102(1):85–113, 2016.

[229] R. N. Shepard. Toward a universal law of generalization for psychological science. *Science*, 237(4820):1317–1323, 1987.

[230] M. A. Silva, R. Prieto, I. Jonsen, M. F. Baumgartner, and R. S. Santos. North atlantic blue and fin whales suspend their spring migration to forage in middle latitudes: building up energy reserves for the journey? *PLoS One*, 8(10):e76507, 2013.

[231] J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *International Conference on Learning Representations*, 2016.

[232] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations Newsletter*, 1(2):12–23, 2000.

[233] G. W. Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566, 1993.

[234] R. H. Stewart. *Introduction to physical oceanography*. Robert H. Stewart, 2008.

[235] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[236] D. Tan and Z. Chen. On a general formula of fourth order runge-kutta method. *J. Math. Sci. Math. Educ*, 7(2):1–10, 2012.

[237] R. E. Tarjan. Finding optimum branchings. *Networks*, 7(1):25–35, 1977.

[238] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[239] C. Thurau, K. Kersting, M. Wahabzada, and C. Bauckhage. Convex non-negative matrix factorization for massive datasets. *Knowledge and information systems*, 29(2):457–478, 2011.

[240] D. J. Trosten, M. C. Kampffmeyer, and R. Jenssen. Deep image clustering with tensor kernels and unsupervised companion objectives. *arXiv preprint arXiv:2001.07026*, 2020.

[241] G. Van Brummelen. *Heavenly mathematics: The forgotten art of spherical trigonometry*. Princeton University Press, 2012.

[242] J. Van Gael, Y. Saatci, Y. W. Teh, and Z. Ghahramani. Beam sampling for the infinite hidden markov model. In *Proceedings of the 25th international conference on Machine learning*, pages 1088–1095. ACM, 2008.

[243] M. van Kreveld and J. Luo. The definition and computation of trajectory and subtrajectory similarity. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, page 44. ACM, 2007.

[244] O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini. Online human-bot interactions: Detection, estimation, and characterization. In *Eleventh international AAAI conference on web and social media*, 2017.

[245] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

[246] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 673–684. IEEE, 2002.

[247] H. M. Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM, 2006.

[248] C. Wang, J. Paisley, and D. Blei. Online variational inference for the hierarchical dirichlet process. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 752–760, 2011.

[249] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1235–1244, 2015.

[250] X. Wang, K. T. Ma, G.-W. Ng, and W. E. L. Grimson. Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. *International journal of computer vision*, 95(3):287–312, 2011.

[251] Z. Wang, C. Long, G. Cong, and C. Ju. Effective and efficient sports play retrieval with deep representation learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 499–509. ACM, 2019.

[252] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM European conference on Computer systems*, pages 205–218, 2009.

[253] C. J. Wu et al. On the convergence properties of the em algorithm. *The Annals of statistics*, 11(1):95–103, 1983.

[254] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016.

[255] A. Y. Xue, R. Zhang, Y. Zheng, X. Xie, J. Huang, and Z. Xu. Destination prediction by sub-trajectory synthesis and privacy protection against such prediction. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 254–265. IEEE, 2013.

[256] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3861–3870. JMLR. org, 2017.

[257] J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2016.

[258] B.-K. Yi and C. Faloutsos. Fast time sequence indexing for arbitrary lp norms. In *VLDB*, volume 385, page 99. Citeseer, 2000.

[259] C. Yin, N. Uosaki, H.-C. Chu, G.-J. Hwang, J.-J. Hwang, I. Hatono, E. Kumamoto, and Y. Tabata. Learning behavioral pattern analysis based on students' logs in reading digital books. 12 2017.

[260] A. Ypma and T. Heskes. Automatic categorization of web pages and user clustering with mixtures of hidden markov models. In *International Workshop on Mining Web Data for Discovering Usage Patterns and Profiles*, pages 35–49. Springer, 2002.

[261] S.-Z. Yu. Hidden semi-markov models. *Artificial intelligence*, 174(2):215–243, 2010.

[262] J. Yuan, Y. Zheng, and X. Xie. Discovering regions of different functions in a city using human mobility and pois. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 186–194. ACM, 2012.

[263] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*, pages 99–108. ACM, 2010.

[264] A. Zhang, S. Gultekin, and J. Paisley. Stochastic variational inference for the hdp-hmm. In *Artificial Intelligence and Statistics*, pages 800–808, 2016.

[265] Q. Zhao, M. C. Willemsen, G. Adomavicius, F. M. Harper, and J. A. Konstan. From preference into decision making: modeling user interactions in recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 29–33, 2019.

[266] Z. Zheng, R. Kohavi, and L. Mason. Real world performance of association rule algorithms. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 401–406, 2001.

[267] F. Zhou and F. Torre. Canonical time warping for alignment of human behavior. In *Advances in neural information processing systems*, pages 2286–2294, 2009.

[268] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. 2018.