



IONIAN UNIVERSITY
DEPARTMENT OF INFORMATICS
HUMANISTIC AND SOCIAL INFORMATICS LABORATORY

Higher order knowledge mining for humanistic data-generated graphs

Georgios Drakopoulos MSECE

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy (PhD) in Computer Science

Defense Committee

Phivos Mylonas, Associate Professor, University of West Attica, Hellas [Advisor]
Spyros Sioutas, Professor, University of Patras, Hellas
Katia-Lida Kermanidis, Associate Professor, Ionian University, Hellas
Lazaros Iliadis, Professor, Democritus University of Thrace, Hellas
Eleanna Kafeza, Associate Professor, Zayed University, UAE
Yannis Velegarakis, Professor, Utrecht University, The Netherlands
Aristides Gionis, WASP Professor, KTH, Sweden

©2024 Georgios Drakopoulos, all rights reserved
July 2024

This page has been intentionally left blank

This page has been intentionally left blank

Contents

List of tables	v
List of figures	vii
List of algorithms	vii
I Introduction	xi
1 Preface	xiii
1.1 General introduction	xiii
1.2 Motivation, objectives, and contribution	xx
1.3 Intended audience and notation	xxiv
1.4 This is a live document	xxv
1.5 Ευχαριστίες	xxvii
1.6 Bibliography	xxviii
2 Executive Summary	xxxi
2.1 Summary	xxxi
2.2 Keywords	xxxiv
II Contributions	1
3 Tensor stack networks	3
3.1 Chapter introduction	3
3.2 Previous work	5
3.3 Introductory material	7
3.3.1 Overview	7
3.3.2 Tensors	7
3.3.3 Least squares power law fit	11
3.3.4 Activation functions	14
3.3.5 Learning rate	15
3.4 Graph resiliency evaluation with TSNs	16
3.4.1 TSN architecture	16

3.4.2	Results	20
3.5	Compressed graph recovery with TSNs	21
3.5.1	TSN architecture	21
3.5.2	Training	23
3.5.3	Graph topology correlation	23
3.5.4	Spatial graph compression	24
3.5.5	Results	25
3.6	Bibliography	26
4	Self organizing maps	37
4.1	Chapter introduction	37
4.2	Previous work	39
4.3	Clustering cognitive tasks with SOMs	40
4.3.1	SOM architecture	40
4.3.2	Training process	43
4.3.3	Results	45
4.4	Bibliography	46
5	Graph neural networks	53
5.1	Chapter introduction	53
5.2	Previous work	57
5.3	Introductory material	59
5.4	Twitter affective coherent community discovery with GNNs	60
5.4.1	GNN architecture	60
5.4.2	Results	62
5.5	GNN for edge fuzzy Twitter graphs	65
5.5.1	Results	68
5.6	Bibliography	69
6	Conclusions	77
6.1	Chapter introduction	77
6.2	Previous work	79
6.3	Conclusions	80
6.3.1	Modeling power	80
6.3.2	Computational complexity	81
6.3.3	Space complexity	81
6.3.4	Digital awareness	81
6.4	Contributions	82
6.4.1	Overview	82
6.4.2	Distinctions	82
6.4.3	Deep learning	86
6.4.4	Graph mining	87
6.4.5	Social network analysis	88
6.4.6	Computational behavioral sciences	89
6.4.7	Computational tools	90
6.5	Prospects	90
6.6	Bibliography	93

<i>CONTENTS</i>	iii
A' World events	99
A'.1 Appendix introduction	99
A'.2 Appendix bibliography	103
Index	109

List of tables

1.1	Chapter research objectives.	xxi
1.2	Bibliography review topics for each chapter.	xxii
1.3	Graph terminology.	xxii
1.4	Key differentiating topics in each chapter.	xxiii
1.5	Notation synopsis.	xxv
1.6	Technical acronyms synopsis.	xxvi
3.1	Power laws.	11
3.2	Properties of the testing, training, and validation graphs.	19
3.3	Components of the two TSN architectures.	19
3.4	Contingency tables for the four pairs architecture/strategy after training.	20
3.5	Metrics for the four TSN pairs architecture/strategy after training.	21
3.6	Contingency tables for the four pairs architecture/strategy after validation.	21
3.7	Metrics for the four TSN pairs architecture/strategy after validation	21
3.8	TSN configurations.	26
3.9	Normalized results for each TSN configuration and benchmark sequence.	26
4.1	Distribution of tasks.	45
4.2	SOM configuration options.	45
4.3	SOM configurations.	45
4.4	Best (left) and average (right) confusion matrices.	46
5.1	Common GNN attribute types.	59
5.2	Vertex state attributes.	60
5.3	Benchmark graph synopsis.	63
5.4	Results for the LS solutions.	64
5.5	Fractions for sign flips of 1821 (left) and US2020 (right) graphs.	64
5.6	Social graph sentiment analysis.	65
5.7	Fuzzy function options.	68
5.8	Normalized number of iterations as a function of the hyperparameter β_0	69
5.9	Emotional distributions (pos/neu/neg) computed with the best value of β_0	69
6.1	Architecture taxonomy.	78
6.2	Research papers forming the basis for each chapter.	79
6.3	Contributions per year.	84

6.4	Number of contributions per field.	84
6.5	Scientific works per field per year.	85
6.6	Scientific works for humanistic data per year.	85
6.7	Deep learning contributions.	87
6.8	Graph mining contributions.	88
6.9	Social network contributions.	89
6.10	Behavioral science contributions.	89
A'.1	Major events in the world.	100
A'.2	Major events in the world (continued).	101

List of figures

1.1	From humanistic data to graphs.	xviii
1.2	Fractals expressed as Lindenmayer systems.	xix
1.3	Plant modelling with Lindenmayer systems.	xix
1.4	Drawing hands – M.C. Escher	xx
1.5	Relationships between chapters.	xxiv
3.1	TSN architecture.	17
3.2	TSN layer structure.	18
3.3	The first TSN architecture.	20
3.4	The second TSN architecture.	20
3.5	Hidden layer structure.	22
3.6	Two example input graphs.	24
4.1	SOM projection.	40
4.2	Neighboring neurons in a rectangular grid with a cross strategy.	42
4.3	Projection of a cylinder to a square.	43
4.4	Mean topological error vs SOM configuration.	47
5.1	Message passing in GNNs in rounds.	55
5.2	Community discovery is the holy grail of graph mining.	61
5.3	Relative steady steady difference vs iteration.	63
5.4	The proposed account behavior model.	65
5.5	Weight functions.	66
6.1	Proposed model.	80
6.2	Announcement of the 2022 Bourbakis award.	92
A'.1	Stairs – M.C. Escher.	102

List of Algorithms

1	Assigning affective polarity to hashtags	62
2	The proposed GNN	62

Part I

Introduction

Chapter 1

Preface

*The next statement is false.
The preceding statement is true.*

1.1 General introduction

WITH the advent of computers, initially analog and now digital, the incessant quest for knowledge has entered a new phase whose potential is yet to be explored. Said quest historically throughout their stages has been heavily relying on innovative and ingenious ways to perform knowledge mining, beginning from modelling natural phenomena with systems of partial differential equations and culminating to modern day data driven knowledge discovery applied on a broad spectrum of domains from long supply chain optimization and smart city planning to computational neurosciences as in the Human Brain Project¹ EU flagship research programme, advanced omics sciences including the Human Genome Project², and space exploration carried out by high end computational clusters and even quantum computers as of recently. Besides the obvious progress in computer science and computers themselves this description implies, there has been also a substantial increase in both the algorithmic and computational complexity of the problems to be addressed as well as a seemingly disproportional one in resources required including

¹<https://www.humanbrainproject.eu>

²<https://www.genome.gov>

but not limited to silicone chip wafers and electrical power, as the recent global semiconductor shortage of 2020-2023³ and the ongoing cryptocurrency mining massive trend⁴ make perfectly clear respectively.

A critical observation is that *humanistic data* play a crucial role in said quest for reasons which will be shortly explained later on. Such data will be extensively studied in this dissertation and in fact they will be one of its focal points, as its title made clear. Humanistic data are generated and are instrumental across a broad spectrum of diverse social, financial [SOOO24], and technological sectors [IKR24] such as digital innovation and entrepreneurship [LWGR24], education [SD24], smart cities [Ces23; HAA⁺24; YJU24], political campaigns [DKK24], Industry 4.0 [MMLF24; JAM⁺23; GTBM23], social media [MT24], protein to protein interaction [BHP⁺23], brain network connectivity [XYH⁺24; VAVA⁺23], and large omics analysis [TLG23]. Central to this dissertation is definition 1.1 outlining the nature of these data, which up to an extent also defines their scope and the scope of this work.

Definition 1.1 (Humanistic data) *Humanistic data are defined as the data collected, processed, and utilized with the explicit purpose of describing, understanding, or improving part of the human condition. Such data can be generated by humans or by the social, financial, political, or technological environment directly influenced by human activity and can come in various forms allowing different processing methodologies.*

From the above description it follows that the contemporary world abounds with humanistic data which are ripe for processing with a wide array of algorithms and techniques and for gaining valuable insight on all sorts of purposes. As a concrete example, the Greater London Authority (GLA) under the Mayor of London has released and maintains since 2017 online multiple datasets relating to the urban life there including traffic, COVID-19 reports, air and water quality, and employment⁵ governed by a strict code of data quality and privacy –not to be confused with the city of London in Canada which coincidentally also maintains a similar digital archive of open data⁶. GLA reports that its data repository is over a hundred terabytes already. Furthermore, on a global scale it is currently estimated that more than 400 million terabytes are being generated with each passing day⁷. The potential of efficiently harnessing the knowledge found inside these enormous datasets is tremendous and it emphasizes the necessity for algorithmic schemes which are inherently designed to overcome the following challenges of humanistic data [Wu24]:

- ♣ **Volume:** Humanistic data are generated *en masse* from a wide array of heterogeneous sources with different properties. With the advent of high speed electronics, the development of various digital ecosystems, the Internet of Things (IoT) and personal area networks, and the digital culture of the general population, the amount of humanistic data generated daily is enormous and continues to rise at an unprecedented scale posing a considerable challenge to existing computational infrastructure. Therefore, more sophisticated processing methodologies should be developed in order to counter this challenge, while simultaneously harvesting knowledge spread in the vast ocean of humanistic data.
- ♣ **Higher order dependencies:** Often humanistic data are described by higher order relationships, such as second and third order correlations, because of memory and feedback effects. These relationships cannot in the general case be reduced down to a collection of linear ones or involving less factors. Therefore, although humanistic data can and frequently are heavily preprocessed, their inherent

³<https://spectrum.ieee.org/chip-shortage>

⁴<https://www.britannica.com/money/what-is-crypto-mining>

⁵<https://data.london.gov.uk/dataset>

⁶<https://opendata.london.ca>

⁷<https://explodingtopics.com/blog/data-generated-per-day>

complexity remains high. However, this is also a major part of their intrinsic value, since the amount of knowledge which can be extracted by them is related not only to their size, but also to their inherent complexity. So, data processing schemes should account for this level of complexity.

- ♣ **Non linearity:** Non linearities frequently arise in humanistic data partly as a direct result of the higher order interplay between the various factors involved. Additionally, the nature of humanistic data, which can often be described by systems of differential equations, leads to non linear solutions, even if said equations are linear, which is not always the case, or can be linearized, which should be done carefully. In order not to lose accuracy in the final solution in the latter case, suitable linearization or iterative schemes can be employed. Alternatively, any neural network architecture should be equipped with an equivalent level of non linearity, typically as activation functions.
- ♣ **Time dependency:** Since human activity progressively changes with time, often in unpredictable ways after disruptive events, it follows immediately that humanistic data, which by definition describe this activity, reflect these changes. Of course the mechanisms through which the original changes are propagating in the humanistic data sources are not the same for every such source and may well be rather indirect. Consequently, changes in the human condition may result in subtle variations to humanistic data. Nevertheless, time dependency exists within the data collected and should be factored in if meaningful knowledge is to be obtained.
- ♣ **Generation velocity:** The velocity at which humanistic data is increasing as technology and the rate of human activity increase. This creates the need for equally quick information processing systems which may occasionally yield inaccurate results up to a point or provide only a sketch of the required information which then is gradually enriched within acceptable time limits. In any case and irrespective of the reasons leading to quicker humanistic data generation, the speed factor should be taken into account during the design face of current information processing systems. To this end, suitable aggregation or dimensionality reduction mechanisms can be integrated into them.
- ♣ **Missing values:** Partly as a result of the volume and speed humanistic data are generated, missing values may be present, whether systematically. Other factors accounting for such missing values range over a wide array of reasons including schema incompatibility, communication errors, incomplete protocols, and faulty equipment. Therefore, missing values are bound to appear in humanistic data and it is to the best interest of the stakeholders and end users alike that sophisticated algorithms designed to either enrich or discard data entries with missing values be designed. Neural network architectures exist or can be training for that task, especially if they perform interpolation.

From the above description it follows that humanistic data are among the most prominent data sources and, perhaps more importantly, most of these sources are open and public, enabling thus the creation of large data sets and digital repositories waiting to be mined and their potential harnessed. Because humanistic data abound with relationships about human activity, it would not be an exaggeration to say that the entire planet is building its own *ad hoc* neural system through myriads of official and unofficial channels alike, underpinning the universal human need for communication and knowledge. This is reflected in the definition of digital awareness given later in definition 1.3 in this section.

The current state of information processing, especially the high level functions allowing the transformation of information to knowledge, frequently entails some form of graph mining or, equivalently in more recent scientific terminology, linked oriented data processing. This is not an accident, since human activity heavily focuses on forming, maintaining, and occasionally dissolving relationships of various types,

such as personal or professional, each with its own semantics and dynamics, paving the way for numerous interesting scenarios when more than one relationships interact, most often when they involve the same group of individuals. This is a key aspect in formulating the principles underlying the design and operations of graph databases like Neo4j⁸, which since the beginning of the last decade have been gradually been integrated into the backbone of modern day digital infrastructure. A major part of the latter has been and is still being optimized for mining linked data defined below in definition 1.2.

Definition 1.2 (Linked data) *Linked data are data point collections structured and interlinked in such a way that the most efficient knowledge mining exploits the nature and properties of the links between the data compared to those in the data points themselves.*

Definition 1.2 introduces a duality in linked data in the sense that the associated knowledge mining relies on two closely related yet distinct factors shown right below:

- ♣ **Vertices (nodes):** In most scenarios vertices represent the data points and contain attributes and other mineable information. Usually they are the starting point in most graph mining algorithms.
- ♣ **Edges (links):** They model the relationships between the data points and are minable themselves. Typically they are used to connect a data point to relevant neighboring ones.

Observe additionally that definition 1.2 besides the aforementioned duality introduces the relative importance between them, akin to the one frequently encountered in regularization problems in linear algebra. Moreover, this duality and the fact that both vertices and edges can be mined is reminiscent of the duality in linear operators defined over Hilbert spaces. Within this link oriented and heavily interconnected context it makes perfect sense to wonder whether existing data processing and knowledge mining algorithms, originally designed for tabular data, suffice for graphs and, more general, for two-dimensional data where location information, which will be examined in chapter 3, topology, namely the main topic of chapter 4, and geometry, which will be the focus of chapter 5, are frequently of paramount importance. For instance, in graphs the set of vertices of a given vertex may contain important information about said vertex. Additionally, in digital image processing the color coherence or the lack thereof between a pixel and its neighboring ones can assist in the identification of object boundaries. In addition to the challenges found in mining humanistic data in general, mining linked data has the additional challenges:

- ♣ **Distributed knowledge:** In linked data a certain piece of information typically is not found in a single vertex and, instead, it has to be constructed from data found in a neighborhood. This frequently entails mining information stored in edges which belong to this neighborhood.
- ♣ **Location sensitivity:** In a certain sense it is the opposite of recursion and it contributes to self similarity through small local perturbations. Moreover, it adds another layer of complexity in graph mining besides time dependency. In linked data the nature of information stored at a vertex or at an edge often depends on the segment of the graph they belong to.
- ♣ **Connection dominance:** In real world graphs edges outnumber the vertices by far. Therefore, the information found in the edges is more than that stored in vertices. So from a graph mining perspective edges frequently contribute more, whether directly or indirectly.

⁸<https://neo4j.com>

- ♣ **Local vs global scale:** In graphs scale is instrumental for knowledge mining. Certain properties and patterns may well hold parts of the graph, but they may vanish when larger parts of the graph or even the entire graph are examined. This can be partly attributed to the inherently distributed nature of the information in graphs. In other words, scale plays a crucial role in graph mining. At this point it should be highlighted that as whether a local or a global property is to be mined is determined by the graph knowledge mining algorithm employed and the task at hand.
- ♣ **Emergence phenomena:** Emergence is said to happen in an evolving graph when a property, usually a global one, does not exist until the graph size reaches a certain level. Alternatively, a graph property may disappear once the number of vertices reaches a threshold. Emergence has been observed in many large scale evolving graphs and it indicates a fundamental graph transformation.
- ♣ **Self similarity:** Self similarity across different scales or even in the same scale but at different parts thereof has been one of the reasons that scale free graphs, an important class of real world graphs. Here it should be mentioned that self similarity is not absolute. Rather, as its name suggests, parts of the graph are quite similar and often identical according to a structural metric. This property gives rise to fractal like^{9 10 11} phenomena similar to these encountered in Lindenmayer systems like the ones shown in figures¹² 1.2 and 1.3. This allows large graphs of high structural complexity to be constructed with a few set of evolution rules and a relatively small start graph.
- ♣ **Recursion:** Similar to the property of self similarity, recursion allows the quick construction of large graphs. However, contrary to self similarity, recursion is stricter as there are no variations between the various parts of the graph. Examples in art are figures 1.4 in section 1.1 and figure A'.1 in section A'.1 both by M.C. Escher (1898-1972)¹³.

Mining humanistic data as shown in figure 1.1 largely consists of three distinct yet related steps which yield a humanistic graph complete with attributes. Each such graph models an underlying domain, possibly in more than one ways depending on the semantics, and typically it is further processed in order to extract knowledge as it will be seen in chapters 3, 4, and 5 of this work. Moreover, chapter 6 will consolidate the technical findings of these chapters through the unifying prism of definition 1.3.

- ♣ **Entity mining:** In this stage the important entities of the problem at hand are identified. Typically, each such entity is a vertex with its own properties as determined either in this step or in the next ones. Typically this step relies heavily on the problem description as well as on any available context. Entity discovery can be probabilistic, in which case each vertex has a probability of existence.
- ♣ **Relationship mining:** Once vertices are determined, the relationships between them are established. These allow the formulation of equivalence classes between vertices and also they give rise to structural patterns. In the simplest case there is only one class of relationships, but this needs not be the case. In fact, there can very well be multiple classes of relationships.
- ♣ **Attribute mining:** Depending on the context, this may be optional or a critical step. Features can be found in both vertices and edges and, furthermore, they can be interrelated. In the latter case this

⁹Fractal systems such as the *Sierpinski gasket* are mentioned in the lyrics of Jonathan Coulton songs

¹⁰<https://www.youtube.com/watch?v=6tsutU92rrE>

¹¹Observe that the outline of France is often referred to as *l'Hexagone*, which is clear in the one and two euro coins –Many thanks to Caprice for this observation!

¹²Recovered from <https://allenpike.com/moeling-plants-with-l-systems>

¹³https://en.wikipedia.org/wiki/M._C._Escher

creates an additional dependence between the attributes of neighboring vertices. This is one natural way to create distributed knowledge in a given graph by exploiting its interconnected nature.

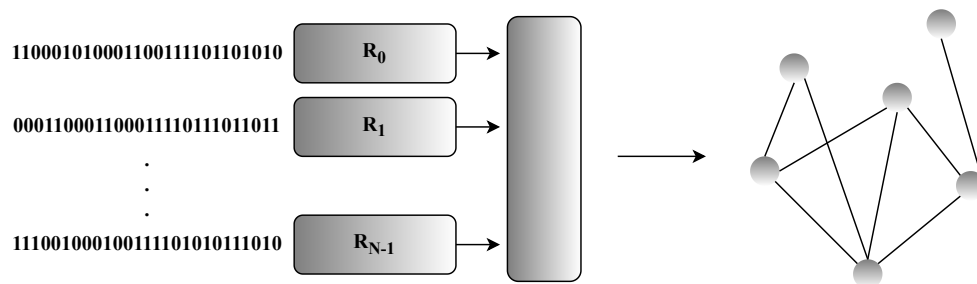


Figure 1.1: From humanistic data to graphs.

Large scale free graphs with strong self similarity tend to exhibit a Zipf law behavior [TMP⁺24] when it comes to properties like degree distribution and eigenvalue scree plot. The principle of least effort [SMS09], inherently connected to Zipf law, may be an explanation for the structural strength of such graphs. Tangible examples of scale free graphs around us are the Semantic Web [Zha24], as the relationships between variables and axioms facilitate a graph reasoners can operate on [XYP⁺24], and the process graphs associated with the enterprise digital transformation caused by the advent Industry 4.0 [YYJ24]. Another example are social media such as Twitter (**not** X) or Reddit, which offer a wide array of services which not only arguably facilitate the digital and real world lives of their users, but also unify Web portals to a single and powerful experience. Such services are social login and single sign on services (SSOS) and geolocation services. To harness their power, elaborate social network analysis (SNA) methodologies based on a multitude of factors have been developed. Notice that these examples fall under definition 1.2.

Numerous sectors of human activity converge through a multitude of heterogenous social, political, technological, and economic paths to create what in this dissertation as well as in the associated research papers is called *digital awareness*, which is defined below in definition 1.3. Intuitively the latter is the technological infrastructure allowing the extension of the human awareness spectrum in terms of event understanding and reaction time. The preceding definition has been inspired by control theory where the events play the role of input, the human reaction corresponds to state, and the actions caused by reaching a particular state to output. Therefore, in this context humans act as a closed loop system with respect to their immediate environment and digital awareness allows them to navigate in this environment with the help of technology, which provides the necessary feedback as well as the necessary control mechanisms.

Definition 1.3 (Digital awareness) *Digital awareness is the spectrum of activities and events humans can recognize at a conscious level with the help of technology or with their own senses, especially while attempting to accomplish a task, along with the cognitive and emotional reactions to these events as well as the actions taken as a result of these reactions.*

As a concluding remark, inspiration for developing innovative problem solving approaches can come from myriad directions, especially since computer science is still (thankfully I say!) more an art than

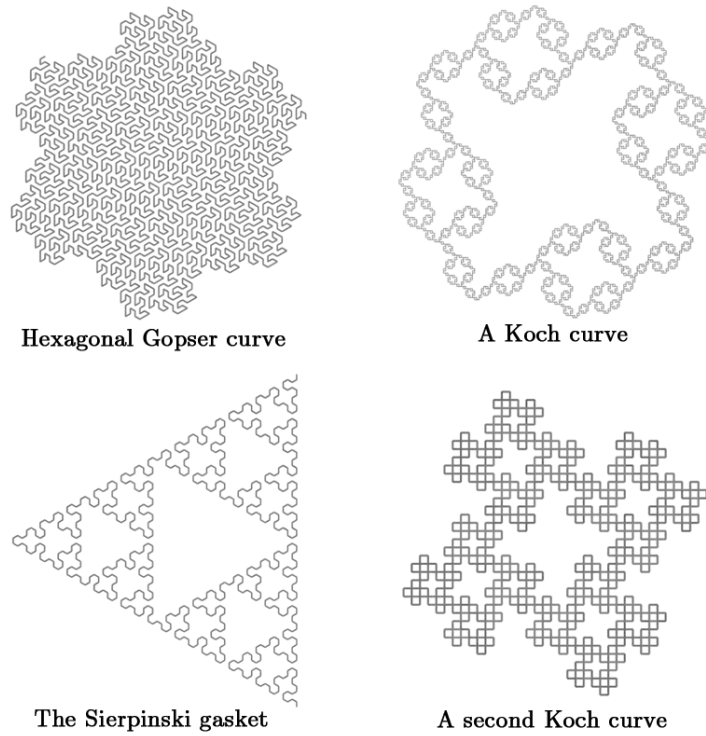


Figure 1.2: Fractals expressed as Lindenmayer systems.



Some 2D plant-like structures from bracketed L-Systems.

Figure 1.3: Plant modelling with Lindenmayer systems.

a science. Major inspirations at a high conceptual level for this dissertation include among others the pioneering ideas regarding communication and information theory, human perception, and how these can be integrated [Wie19]. Additionally, the evolutionary ideas, including how species can actually simulate games in the strict game theoretic sense, as explained in [Daw16]. Other sources of inspiration were the framework for intuitively handling self reference and infinite loop variations [Hof79; Hof07], randomness [CC07], simulation [Bau94], which in fact was one of the mandatory reading material for actors in [Tm99], and discrete memes, like hashtags, and their importance to building thoughts [Bla00]. Special mention goes to Sid Meier's¹⁴ legendary games which were among the first to include an inclination for strategic thinking, but also empathy and a spirit of cooperation. After all, it is well known the unbreakable rule that nobody dies in a Sid Meier game [MN20]. So let us remain civilized.

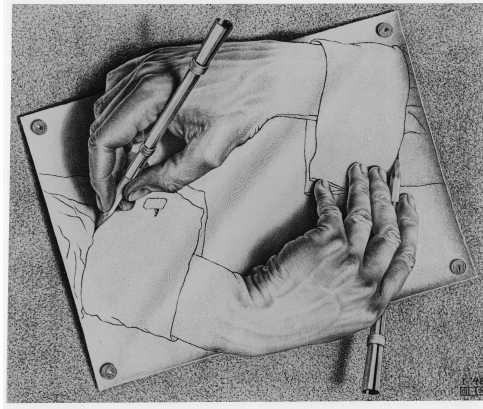


Figure 1.4: Drawing hands – M.C. Escher

1.2 Motivation, objectives, and contribution

The primary research objectives of this dissertation are succinctly given in objective statement 1.1. The research objectives of this dissertation along with the conclusions will be given in detail in chapter 6.

Objective 1.1 (Thesis research objectives and contributions) *The primary research objective is the study of graphs generated by humanistic data and the development of knowledge mining algorithms based on higher order principles. The latter are natural for graphs because of the inherently distributed nature of information stored therein. As an additional objective the effect of these algorithms have on digital awareness as explained in definition 1.3 is explored. The contributions are in the following areas:*

- ♣ **Graph resiliency evaluation:** *In numerous technological and financial sectors such as long supply chain monitoring and logistics, large omics analysis, and transportation networks graph resiliency is an important operational parameter. Graphs with high resiliency can offer alternatives, often in the form of richer structural patterns, against structural failures. A tensor stack network trained on Kronecker synthetic graphs of varying complexity can evaluate said resiliency with results having a high correlation with Estrada and odd Estrada indices. These rely on graph spectrum which is costly to compute.*

¹⁴https://en.wikipedia.org/wiki/Sid_Meier

- ♣ **Graph information assisted decompression:** Since real graphs are frequently large, they come in compressed forms. One way to compress them is the discrete cosine transform, which heavily relies on higher order spatial patterns. Keeping only a fraction of the transform coefficients results in information loss, which can be negated by exploiting graph similarity as measured by graph topological correlation. In this case, compressed graphs are expanded by a tensor stack network which is given knowledge of similar social graphs. As a concrete example graphs from three different fields are used.
- ♣ **Clustering of brain images for cognitive task recognition:** The properties of brain connectivity networks and its evolution over time are paramount for interpreting neuroimaging results. Although brain atlases do exist, the need for clustering brain images remains. A self organizing map with various distance metrics, proximity metrics, reward functions, and learning rates clusters fMRI images representing three distinct cognitive tasks taken from an open dataset. Combining a tensor distance metric with Gaussian proximity metric and reward function achieves the best results.
- ♣ **Affective coherency assessment of Twitter graphs:** Emotions are prime action motives for humans and this is taken well into consideration in the design of social media. A graph neural network relying on local ground truth state vectors containing primarily linguistic attributes discovers communities in two large Twitter (**not** X) graphs with diverse features with considerably more consistency compared to a first order approximation based on emotional word count. The same holds true in fuzzy Twitter graphs where the probability of a message passing across an edge is also factored in.

Additionally, research objective statements 3.1, 4.1, and 5.1 summarize the work presented in chapters 3, 4, and 5 respectively as shown in table 1.1. This table can serve as a quick index for the reader.

Table 1.1: Chapter research objectives.

Statement	Section	Objective
Statement 3.1	Section 3.1	Tensor stack network for decompressing social graphs
Statement 4.1	Section 4.1	Self organizing maps for clustering cognitive tasks
Statement 5.1	Section 5.1	Graph neural networks for community discover in social graphs
Statement 6.1	Section 6.1	Conclusions and research overview

As the name of this dissertation suggests, graphs are the mainstay since not only they are the main data type used, but also they are the basic building block of the knowledge mining algorithms developed here. Another major driver, closely related to graphs, is tensor algebra as it allows for compact, elegant, and more powerful notation than that of linear algebra as well as it naturally captures adjacency tensors. Moreover, moving to higher dimensions is a common technique in the fields of machine learning (ML) and pattern recognition (PR), often coupled with a suitable non linear kernel.

Each chapter has its own dedicated literature review section which reflects the topics each such chapter relies on and it is complete with recent as well as with classical references. In table 1.2 these topics are shown. Observe the symmetry in that the topics progressively move from the more specific to the more general, always beginning with the neural network architecture described and used in the specific chapter.

Table 1.2: Bibliography review topics for each chapter.

Section	Topics
Section 3.2	Tensor stack networks, tensor signal processing, neural networks, graph compression
Section 4.2	Self organizing maps, fMRI analysis, clustering
Section 5.2	Graph neural networks, graph signal processing, graph mining, SNA
Section 6.2	Research papers by the author.

At this point a note regarding terminology is necessary. Specifically, there are two sets of basic graph terms, one associated with graphs as algorithmic objects and their structural properties and one denoting functionality. In particular, the terms *vertex*¹⁵, *edge*, and *graph* belong to the first set, whereas the terms *node*, *link*, and *network* to the second one. In this dissertation the first set will be heavily used. Also, the terms *attribute* and *feature* are considered completely equivalent. The above are summarized in table 1.3, wherein examples are taken from computer science, logistics, and large omics processing.

Table 1.3: Graph terminology.

Term	Context	Examples	Term	Context	Examples
Vertex	Structural	Entities Boolean variables River banks States Symbol states	Node	Functional	Computers Social media accounts Proteins Brain parts Access points
Edge	Structural	Functional dependency Same clause Bridges Transitions Symbols	Link	Functional	Network connections Friendship Protein interactions Concurrent activation Wireless connections
Graph	Structural	Entity collection Proposition Königsberg Finite state automaton Trellis diagram	Network	Functional	Workgroup, domain, or realm Social graph Protein interaction network Brain connectivity network Wireless networks

The key differentiating points of this dissertation from the majority of preceding research regarding knowledge mining from connection oriented humanistic data are listed below. They capture a major part of the innovation spirit of this work, although the interested reader can possibly find other appealing research topics or bibliographic citations as well. Brief descriptions of these topics follow.

- ♣ Tensor algebra is the generalization of linear algebra in at least three dimensions. In this context vectors and matrices are tensors of dimension one and two respectively. The central idea of connecting linear spaces remain, but this is extended to simultaneously coupling more such spaces, not necessarily distinct between them, through a linear mapping. Despite tensors being linear, they fre-

¹⁵Literally peak in Latin.

quently appear in the linear approximation of non-linear functions, especially in expansions such as Taylor or Volterra series, often being satisfactory first order approximations.

- ♣ Higher order interaction is prevalent in graph mining and in general in knowledge extraction from link oriented data (LOD) since graphs inherently contain distributed graph which cannot by definition be harnessed by first or second order methods, namely by accessing a vertex or an edge and their adjacent vertices or edges respectively. Thus extending the graph information search and processing in neighborhoods of bigger sizes yields more accurate knowledge, especially if information locality and spatial coherence is strategically exploited, at the expense of complexity.
- ♣ Geometry has recently re-emerged as a popular topic in knowledge mining and engineering for two primary reasons. First, it places heavy emphasis on loci which have clear conditions. Second, geometry leads to natural constraints based on easily understood by humans concepts such as distances and boundaries. Therefore, geometry based neural network architectures can rely on both conditions and boundaries to perform regularization and avoid overfitting or vanishing gradients, which translates to balanced and efficient learning of the main properties of the dataset at hand.
- ♣ Unsupervised learning along with reinforced learning, which is not examined here, are viable alternatives to supervised learning as no annotated datasets are necessary. Moreover, knowledge is progressively extracted by the data themselves, which may bring to light latent data interdependencies or additional data structure, enhancing therefore the value of large datasets for the respective stakeholders. A necessary and crucial condition for this is to efficiently and sufficiently code the geometric properties both into the architecture as well as its implementation.
- ♣ A unifying underlying concept throughout this dissertation is digital awareness explained earlier in definition 1.3. It relies heavily on accurate and timely knowledge generated from vast and unstructured or semi-structured datasets. To this end, the role of technology is paramount, as it allows a virtual navigation through the ocean of data in a trajectory dictated by operational objectives through the formulation of an adaptive feedback loop expanding considerably the perception of human operators and reducing significantly their reaction time to unexpected errors or failures.

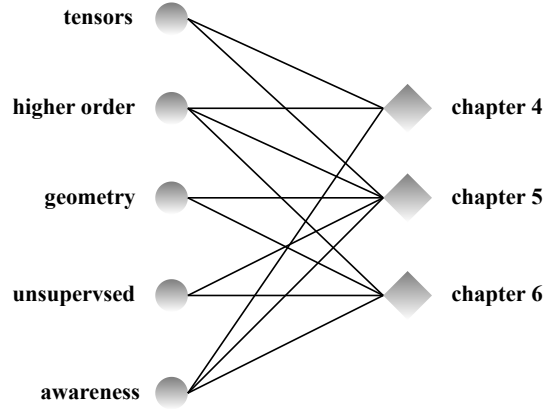
Since this dissertation is about relationships and their management, it makes perfect sense to analyse how the material presented in each of the chapters 3, 4, and 5 is interconnected. The relationships between chapters are explained as follows and moreover are shown in figure 1.5. For the topics of each research paper by the author please see appendix A', where these connections are explained in detail.

Table 1.4: Key differentiating topics in each chapter.

Chapter	Tensors	Higher order	Geometry	Unsupervised	Awareness
Chapter 3	x	x			x
Chapter 4	x	x	x	x	x
Chapter 5		x	x	x	x

In order to make the content of this dissertation accessible to a larger audience in an ever expanding and highly diverse digital world it has been written in English. The only exception to that rule is subsection 1.5 where the author extends his warmest thanks to the people whose contribution was invaluable from various aspects at various stages during the research leading to this dissertation.

Figure 1.5: Relationships between chapters.



1.3 Intended audience and notation

The intended audience of this dissertation is expected to be the equivalent of at least masters level students or seasoned practitioners, primarily because of the heavy use of technical terminology of algorithmic engineering as well as because of the extensive mathematical formalism. Moreover, knowledge of linear and tensor algebra, graph mining, signal processing, social media analysis, and deep learning is required and mathematical treatment has no been held back. However, that being written, familiarization with mathematics is not a strict requirement as most of the concepts can be intuitively understood. Needless to write at this point that of course everyone is more than welcome to read this text. With the widespread adoption of humanistic data processing systems, individuals who may be interested in the research findings reported here are smart city administrators, hospital and health infrastructure administrators, social media designers and political campaign planners, and digital content creators.

Also in this section the notation of this dissertation is presented and explained. Keeping all the symbols in a central reference point reduces redundancy and helps the reader locating quickly notation conventions. Table 1.5 summarizes the notation of this dissertation. Technical acronyms are defined the first time they are encountered in the text and are also summarized in table 1.6. Capital calligraphic letters denote random variables and spaces, depending on the context, capital boldface are reserved for matrices and tensors, lower boldface represent vectors, and finally ordinary capital and lowercase letters are scalars. Expressions from other languages, mostly Latin ones, are written in italics. Square brackets have been reserved for vectors, matrices, and tensors, whereas parentheses denote tuples.

Observe that as shown in table 1.5 the set cardinality operator and the absolute value operator for scalars have the same symbol. This, however, should not cause any confusion whatsoever to the meticulous reader since they appear in different algorithmic settings. Moreover, the use of each will be clear from the context. Also in many technical texts the same symbol denotes path length in graphs. This has been avoided by using a dedicated symbol for it in order to make notation clear.

Table 1.5: Notation synopsis.

Type	Symbol	Meaning	First in
General	$\triangleq$	Definition or equality by definition	Eq. (3.1)
	I_0	Interval $[0, 1]$	Sec. 3.3
	I_1	Interval $-I_0 \cup I_0$	Sec. 3.3
	Θ_0	Interval $[0, 2\pi)$	Sec. 5.1
Linear algebra	$\text{vec}(\mathbf{M})$	Matrix vectorization operator	Eq. (3.37)
	$\mathbf{M}_1 \otimes \mathbf{M}_2$	Kronecker tensor product	Eq. (3.9)
	$\times_k$	Tensor multiplication in the k -th dimension	Eq. (3.8)
Set theory	$\{s_1, \dots, s_n\}$	Set with elements $s_1, \dots, s_n$	Eq. (3.27)
	$(t_1, \dots, t_n)$	Tuple with elements $t_1, \dots, t_n$	Eq. (5.15)
	$ S $	Set, sequence, or tuple cardinality	Eq.
Graph theory	$\deg(v)$	Degree of vertex v	Eq. (3.55)
	$\text{edge}(u, v)$	Edge between vertices u and v	Eq. (5.16)
	$\text{edge}(i, j)$	Edge between vertices v_i and v_j	Eq. (5.18)
	$\text{prob}\{\Omega\}$	Probability of event Ω	Eq.
SOMs	$\text{loc}(x)$	Grid location for data point x	Eq. (4.18)
	$\text{invloc}(u)$	Set of data points assigned to neuron u	Eq. (4.4)
	$\text{weight}(u)$	Synaptic weights of neuron u	Eq. (4.2)
	$\text{neighb}(u_1, u_2)$	Indicator function of neighboring neurons	Eq. (4.5)
	$\text{argmin}[f(\cdot)]$	Argument minimizing function f	Eq. (4.12)

1.4 This is a live document

At the dawn of the digital age live documents are indispensable tools for effective communication where the data to be displayed, metadata such as digital signatures, and formatting instructions are intertwined. This reasoning can be carried over numerous file formats including portable document format (pdf)¹⁶. This dissertation has been written in $\text{\LaTeX}$ which allows more flexibility and offers more functionality than the original $\text{\TeX}$ ¹⁷ typesetting system developed by Donald Knuth as well as than $\text{\LaTeX}$ ¹⁸, its popular successor. This includes better support for the popular *bibtex*¹⁹ and *natbib*²⁰ packages, enabling the existence of a separate references unit for each chapter, facilitating thus bibliography entry lookup. The index was created with *makeindex*²¹, which after all these years still remains one of the best ways for index creation. To this end, all internal text references, such as table and bibliographical references as well as section and chapter mentions, are clickable links leading to the proper point inside the document. However, for aesthetic reasons and for maintaining consistency with the overall grayscale theme of this text, they have the same color with the regular font. Although this may sound inconvenient and making such links hard to spot, it prevents pages from being cluttered and instead maintains their minimalistic design. Moreover, Web links as well as any other external pointers have been disabled for security reasons. With the notable exception of figure 5.2 images are grayscale for aesthetic reasons. Wherever possible the appropriate digital object

¹⁶<https://pdfa.org/resource/iso-32000-2>

¹⁷<https://www.tug.org>

¹⁸<https://latex.org>

¹⁹<https://ctan.org/pkg/bibtex>

²⁰<https://ctan.org/pkg/natbib>

²¹<https://ctan.org/pkg/makeindex>

identifier (stylized as doi as shown in table 1.6) for bibliographic items has been added in order to speed up online bibliographical search and at the same time to improve its accuracy. Every reference to a person is accompanied by a link to their homepage, typically part of their respective home organization Web page. This is done in order to ensure unambiguous references to an ever expanding digital world full of small world phenomena and manifestations of the Tansley effect²². In table 1.6 the technical acronyms are shown.

Table 1.6: Technical acronyms synopsis.

Acronym	Meaning	First in
API	Application interface	Section 3.2
CNN	Convolutional neural network	Section 3.1
DL	Deep learning	Section 1.2
FFNN	Feedforward neural network	Section 3.2
FFT	Fast Fourier transform	Section 5.1
GSP	Graph signal processing	Section 3.2
IR	Information retrieval	Section 3.1
LMS	Least mean squares	Section 3.1
LOD	Link oriented data	Section 1.3
LS	Least squares	Section 3.3
ML	Machine learning	Section 1.2
MLDA	Multilinear discriminant analysis	Section 3.2
NLP	Natural language processing	Section 3.3
RNN	Recurrent neural network	Section 3.1
SNA	Social network analysis	Section 1.1
SVD	Singular value decomposition	Section 3.2
TDA	Tensor discriminant analysis	Section 3.2

Finally, concerning the text per se, a considerable effort was made to make the text as legible as possible. To this end, long sequences of paragraphs have been avoided since the reader is usually not inclined to read them²³. Instead, they have been replaced by restructured text, bulleted lists, tables, figures, or other means as appropriate. Along the same line of reasoning, for readability purposes wherever possible content has been moved to tables. The last line of every paragraph is terminated within the right half of a page to avoid giving a sense of incompleteness. For the same reason a page is always terminated at the bottom half of the page. A section is always terminated with a paragraph. Three lines is the minimum number for each paragraph. Figures have been added to illustrate the main points. Along a similar line of reasoning, after an equation is presented frequently an intuitive explanation is additionally given so that the reader can follow the text flow. Moreover, metatext in the form of chapter or section introduction as appropriate has been added for clarity. Figures, tables, algorithms, equations, theorems, and definitions are named before they are used and their names are not capitalized to avoid page cluttering. Figures and tables are properly credited, otherwise they were made by the author. Harvard comma has been carefully preserved [Obe11]. As stated earlier, proper technical English has been used. Inclusive and gender-neutral language has been carefully used. Finally, regarding citations, references to scientific works are placed in square brackets, whereas links to online resources are placed in footnotes to avoid cluttering the bibliographical sections.

²²<https://wholepeople.com/tansley-effect>

²³<https://www.theguardian.com/commentisfree/2018/aug/25/skim-reading-new-normal-maryanne-wolf>

Said resources have been last accessed during the writing of this dissertation, but every effort has been made to exclusively rely on major Web sites who are much less likely to remove such a resource despite the highly dynamic nature of the Web. Formatting guidelines have been taken from the authoritative resource for $\text{\TeX}$ and document preparation [KB86][GMS⁺94].

1.5 Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον επιβλέποντα της διατριβής αναπληρωτή καθηγητή του πανεπιστημίου Δυτικής Αττικής Φοίβο-Απόστολο Μυλωνά²⁴ για την αгаστή συνεργασία καθώς και για την αποδοχή του θέματος το οποίο του πρότεινα. Αρχηγέ με συγχωρείς που σε ταλαιπώρησα. Επίσης θα ήθελα να εκφράσω τις ευχαριστίες μου για την αρωγή τους και τις εποικοδομητικές προτάσεις τους στα άλλα δύο μέλη της τριμελούς επιτροπής, τον καθηγητή του πανεπιστημίου Πατρών Σπύρο Σιούτα²⁵ και την αναπληρώτρια καθηγήτρια του Ιονίου πανεπιστημίου και πρόεδρο του τμήματος Πληροφορικής Κάτια-Λήδα Κερμανίδου²⁶. Επιπροσθέτως σημαντική συνεισφορά είχαν και τα υπόλοιπα μέλη της επταμελούς επιτροπής, κυρίως η πρώην αναπληρώτρια καθηγήτρια του πανεπιστημίου Zayed και νυν διακεκριμένο μέλος του Technology Innovation Institute (TII)²⁷ στα ΗΑΕ Ελεάννα Καφέζα²⁸ με την οποία συνδιοργανώσαμε και δύο workshop^{29 30} με σχετική θεματολογία και ο καθηγητής του Δημοκριτείου πανεπιστημίου της Θράκης Λάζαρος Ηλιάδης³¹. Επιπλέον θα ήθελα να ευχαριστήσω τον καθηγητή του πανεπιστημίου της Ουτρέχτης Γιάννη Βελεγράκη³² καθώς επίσης και τον καθηγητή του ΚΤΗ Αριστείδη Γκιώνη³³ για τα εποικοδομητικά τους σχόλια. Εκτός επταμελούς επιτροπής σημαντική ήταν και η βοήθεια ακαδημαϊκών όπως του του φίλου Άρη Ηλία³⁴, όπως και με τις προσομοιώσεις στην διπλωματική μου εργασία χρόνια πριν. Ξεχωριστή μνεία αξίζει στον καθηγητή του πανεπιστημίου Πατρών Ευστράτιο Γαλλόπουλο³⁵ και φυσικά δεν θα μπορούσα επ' ουδενί να παραλείψω συγκίνηση ιδιαίτερη εγκωμιαστική, θερμή, και τιμητική αναφορά στον αναπληρωτή καθηγητή του πανεπιστημίου Πατρών Χρήστο Μακρή³⁶.

Η διατριβή αυτή πιθανότητα να είχε πολύ διαφορετική μορφή και αρκετά τρωτά σημεία χωρίς την συμβολή ορισμένων ατόμων τα οποία θέλω να ευχαριστήσω και εδώ. Αρχικά θα ήθελα να στείλω περήφανο χαιρετισμό για την τετάρτη Ιουλίου στον theodork^{37 38 39 40} και να προσθέσω no taxation without representation⁴¹, να υποβάλλω τα σέβη μου και να κλίνω ευλαβικά το γόνυ στον πανθυπερσέβαστο και πορφυρογέννητο συνεχστή των 4B tsiliman⁴², να ευχαριστήσω τους sanidas (κατά μήκος της διχοτόμου),

²⁴<https://ai-nnohub.uniwa.gr/ereynites/mylonas-foivos/>

²⁵<https://www.ceid.upatras.gr/webpages/faculty/sioutas>

²⁶<https://di.ionio.gr/gr/departement/staff/92-kermanidis/>

²⁷<https://www.tii.ae>

²⁸<https://orcid.org/0000-0001-9565-2375>

²⁹<https://dl.acm.org/doi/abs/10.1145/3459637.3482039>

³⁰<https://dl.acm.org/doi/abs/10.1145/3511808.3557937>

³¹<https://civil.duth.gr/cv/iliadis>

³²<https://www.uu.nl/staff/IVelegakis>

³³<https://www.kth.se/profile/argioni>

³⁴<https://www.ceid.upatras.gr/personnel/ilias-aristeidis/>

³⁵<https://www.ceid.upatras.gr/personnel/gallopylos-eystratios>

³⁶<https://www.ceid.upatras.gr/personnel/makris-christos>

³⁷Έπειτα από τόσα χρόνια χρησιμοποιώ πλέον τα login στο diogenis. If you know you know.

³⁸<https://www.imdb.com/title/tt0057012>

³⁹<https://www.imdb.com/title/tt0062622>

⁴⁰<https://www.imdb.com/title/tt0066921>

⁴¹https://en.wikipedia.org/wiki/No_taxation_without_representation

⁴²<https://www.imdb.com/title/tt0104257>

tsiaras, emmanuel⁴³, και krommudi για τις ατέρμονες συζητήσεις στο (παλαιό πλέον) Νέον και στο (επίσης παλαιό) υπόγειο κυλικείο του ceid ανάμεσα σε διαλέξεις και εργαστήρια, να ευχαριστήσω τον Πάνο τον αμερικάνο για τις δημοσιεύσεις τις οποίες γράψαμε και τους αλγορίθμους τους οποίους σχεδιάσαμε στην γιάφκα, τον Σωτήρη τον σάτυρο για τις τεχνολογικές και φιλοσοφικές συζητήσεις καθώς και για τις κινηματογραφικές προβολές, τον Ντίνο τον νταή για τις συζητήσεις περί προγραμματισμού και φιλοσοφίας σχεδιασμού γλωσσών, τον Νίκο την ραπτομηχανή για τις ατελείωτες συζητήσεις περί προγραμματισμού καθώς και για το σχόλιό του για τον πρώτο μου κώδικα Python μεγάλου μεγέθους⁴⁴, τον Φώτη τον υποχθόνιο για τις ατελείωτες αναζητήσεις στο πεδίο της μάθησης μηχανής, της κοινωνικής παρατήρησης, της λελογισμένης πράξης, και όχι μόνο, καθώς και τον Μιχάλη και την Βίκυ για την ανερυθρίαστα ρηξικέλευθη abstreoφιλοσοφία⁴⁵ τους η οποία διαπερνά το στιγμιαίο και ατενίζει το άπειρο καθιστώντας το άτομο αέναο κοινωνό και μυσταγωγό της υψηλής τέχνης. Επίσης (και) από εδώ να πω στον dempelis πως είχε θρυλική παρουσία στους σταθμούς εργασίας στο υπολογιστικό κέντρο⁴⁶ και φυσικά να δώσω πολλές ευχαριστίες στον marounta ο οποίος σε κρίσιμες στιγμές πάντα είχε μια καλύτερη πρόταση. Φρέσκες ιδέες είχαν ο Θανάσης και η Ξένια. Από την ενότητα αυτή δεν θα μπορούσαν να απουσιάζουν η Ασπασία, η Μελίνα, η Tamara, η Διονυσία, η Αναστασία, η Τατιάνα, η Κλεοπάτρα, η Caldera από το παγωμένο Tromsø για τις συζητήσεις για τον Harari, το πότε και πως γίνεται σωστή έξοδος, τι είναι τα ψάρια, και γιατί η φέτα δεν είναι απλά ένα λευκό τυρί, τα ατελείωτα λευκά ενεργειακά ποτά, και για την λιτανεία ενάντια στον φόβο⁴⁷, η Ιουλία, η Lisa και η απέχεθειά της για την ελληνική πίτσα, η Franziska και οι μεγάλες συζητήσεις για τις ταινίες και τον Leonard Cohen, η Isabelle από τις όχθες της Bodensee, το μικρό Caprice, η Lexi, η Ember, και η Aelita με το ελληνικής έμπνευσης όνομα και οι συζητήσεις για τον Chomsky^{48 49}. Ξεχωριστή μνεία αξίζει στην καθηγήτρια του πανεπιστημίου της Αβάνας Claudia Carricarte Naranjo για τις συζητήσεις περί της νευροβιολογικής εξάρτησης του εγκεφάλου από την καρδιά και πως αυτό αποτυπώνεται με τεχνικές νευροαπεικόνισης. Να θυμόμαστε να είμαστε EKA⁵⁰. Τέλος, ιδιαίτερες ευχαριστίες στον Ρήγα και στα σάντουιτς τα οποία κάνει. Διπλό τυρί, μπέικον, και ριγανάτο για να πάει καλά η μέρα. Κι έπειτα ένα δεύτερο για να πάει ακόμα καλύτερα.

Εν κατακλείδι, θα ήθελα να αφιερώσω την παρούσα διατριβή σε εμένα.

1.6 Bibliography

[Bau94] Jean Baudrillard. *Simulacra and simulation*. University of Michigan Press, 1994.

[BHP⁺23] Md Abul Basar, Md Faruk Hosen, Bikash Kumar Paul, Md Rakibul Hasan, SM Shamim, and Touhid Bhuyian. Identification of drug and protein-protein interaction network among stress and depression: A bioinformatics approach. *Informatics in medicine unlocked*, 37, 2023.

[Bla00] Susan J Blackmore. *The meme machine*. Oxford Paperbacks, 2000.

[CC07] Cristian S Calude and Gregory J Chaitin. *Randomness and complexity: From Leibniz to Chaitin*. World Scientific, 2007.

⁴³<https://www.imdb.com/title/tt0113277>

⁴⁴"It is the best C code I have ever seen written in Python."

⁴⁵<https://www.instagram.com/abstreo>

⁴⁶<https://www.ceid.upatras.gr/el/ypologistiko-kentro>

⁴⁷https://dune.fandom.com/wiki/Litany_Against_Fear

⁴⁸Δεν απεβίωσε όπως εσφαλμένα ειπώθηκε στις 18 Ιουνίου όσο γράφονταν αυτό το κείμενο

⁴⁹<https://apnews.com/article/noam-chomsky-alive-not-dead-5b7a1b23b8731ca311e1ec38cdc3c119>

⁵⁰<https://www.youtube.com/watch?v=x-nGRnMfsMY>

- [Ces23] Eugenio Cesario. Big data analytics and smart cities: Applications, challenges, and opportunities. *Frontiers in big data*, 6, 2023.
- [Daw16] Richard Dawkins. *The selfish gene*. Oxford University Press, 2016.
- [DKK24] Katharine Dommett, Glenn Kefford, and Simon Kruschinski. *Data-driven Campaigning and Political Parties: Five Advanced Democracies Compared*. Oxford University Press, 2024.
- [GMS⁺94] Michel Goossens, Frank Mittelbach, Alexander Samarin, et al. *The L^AT_EX companion*. Addison-Wesley Reading, 1994.
- [GTBM23] Mariia Golovianko, Vagan Terziyan, Vladyslav Branytskyi, and Diana Malyk. Industry 4.0 vs. Industry 5.0: co-existence, transition, or a hybrid. *Procedia Computer Science*, 217:102–113, 2023.
- [HAA⁺24] Noor Ul Huda, Ijaz Ahmed, Muhammad Adnan, Mansoor Ali, and Faisal Naeem. Experts and intelligent systems for smart homes’ transformation to sustainable smart cities: A comprehensive review. *Expert Systems with Applications*, 238, 2024.
- [Hof79] Douglas R Hofstadter. *Gödel, Escher, Bach*. Harvester press Hassocks, 1979.
- [Hof07] Douglas R Hofstadter. *I am a strange loop*. Basic Books, 2007.
- [IKR24] Omar S Itani, Ashish Kalra, and Amin Rostami. How does big data affect organizational financial performance in turbulent markets? The role of customer-linking and selling capabilities. *Technological Forecasting and Social Change*, 201, 2024.
- [JAM⁺23] Zohaib Jan, Farhad Ahamed, Wolfgang Mayer, Niki Patel, Georg Grossmann, Markus Stumptner, and Ana Kuusk. Artificial intelligence for Industry 4.0: Systematic review of applications, challenges, and opportunities. *Expert Systems with Applications*, 216, 2023.
- [KB86] Donald Ervin Knuth and Duane Bibby. *The T_EXbook*. Addison-Wesley Reading, 1986.
- [LWGR24] Christina Lubinski, R Daniel Wadhwani, William B Gartner, and Renee Rottner. Humanistic approaches to change: Entrepreneurship and transformation. *Business History*, 66(2):347–363, 2024.
- [MMLF24] Nijat Mehdiyev, Lea Mayer, Johannes Lahann, and Peter Fettke. Deep learning-based clustering of processes and their visual exploration: An industry 4.0 use case for small, medium-sized enterprises. *Expert Systems*, 41(2), 2024.
- [MN20] Sid Meier and Jennifer Noonan. *Sid Meier’s Memoir! A Life in Computer Games*. WW Norton & Company, 2020.
- [MT24] L Meghan Mahoney and Tang Tang. *Strategic social media: From marketing to social change*. John Wiley & Sons, 2024.
- [Obe11] David D Oberhelman. The Chicago manual of style online. *Reference Reviews*, 25(1):8–9, 2011.
- [SD24] Ana Stojanov and Ben Kei Daniel. A decade of research into the application of big data and analytics in higher education: A systematic review of the literature. *Education and Information Technologies*, 29(5), 2024.

- [SMS09] Alexander I Saichev, Yannick Malevergne, and Didier Sornette. *Theory of Zipf's law and beyond*, volume 632. Springer Science & Business Media, 2009.
- [SOOO24] Philip Olaseni Shoetan, Adedoyin Tolulope Oyewole, Chinwe Chinazo Okoye, and Onyeka Chrisanctus Ofodile. Reviewing the role of big data analytics in financial fraud detection. *Finance & Accounting Research Journal*, 6(3):384–394, 2024.
- [TLG23] Min Tang, Yi Liu, and Xun Gong. Multi-omics data mining techniques: Algorithms and software. *Methodologies of Multi-Omics Data Integration and Data Mining: Techniques and Applications*, pages 55–74, 2023.
- [Tm99] The matrix. <https://www.imdb.com/title/tt0133093/>, 1999.
- [TMP⁺24] Shinnosuke Takamichi, Hiroki Maeda, Joonyong Park, Daisuke Saito, and Hiroshi Saruwatari. Do learned speech symbols follow Zipf's law? In *ICASSP*, pages 12526–12530. IEEE, 2024.
- [VAVA⁺23] Tamara Vasilkovska, Mohit H Adhikari, Johan Van Audekerke, Somaie Salajeghe, Dorian Pustina, Roger Cachope, Haiying Tang, Longbin Liu, Ignacio Munoz-Sanjuán, Annemie Van der Linden, et al. Resting-state fMRI reveals longitudinal alterations in brain network connectivity in the zQ175DN mouse model of Huntington's disease. *Neurobiology of Disease*, 181, 2023.
- [Wie19] Norbert Wiener. *Cybernetics or Control and Communication in the Animal and the Machine*. MIT Press, 2019.
- [Wu24] Di Wu. *Data Mining with Python: Theory, Application, and Case Studies*. CRC Press, 2024.
- [XYH⁺24] Jiaxing Xu, Yunhan Yang, David Huang, Sophi Shilpa Gururajapathy, Yiping Ke, Miao Qiao, Alan Wang, Haribalan Kumar, Josh McGeown, and Eryn Kwon. Data-driven network neuroscience: On data collection and benchmark. *Advances in Neural Information Processing Systems*, 36, 2024.
- [XYP⁺24] Siheng Xiong, Yuan Yang, Ali Payani, James C Kerce, and Faramarz Fekri. Teilp: Time prediction over knowledge graphs via logical reasoning. In *AAAI*, volume 38, pages 16112–16119, 2024.
- [YJU24] Shubo Yang, Atif Jahanger, and Muhammad Usman. Examining the influence of green innovations in industrial enterprises on China's smart city development. *Technological Forecasting and Social Change*, 199, 2024.
- [YYJ24] Maolin Yang, Yuqian Yang, and Pingyu Jiang. A design method for edge-cloud collaborative product service system: A dynamic event-state knowledge graph-based approach with real case study. *International Journal of Production Research*, 62(7):2584–2605, 2024.
- [Zha24] Yizheng Zhao. Efficient computation of signature-restricted views for semantic Web ontologies. In *Web Conference*, pages 1945–1953. ACM, 2024.

Chapter 2

Executive Summary

*The computer was born to solve
problems that did not exist before*
Bill Gates (1955 –)

2.1 Summary

GRAPHS constitute the primary algorithmic vehicle for representing relationships of various types among entities and the complex interaction between them. Said entities are frequently drawn from an underlying domain having different kinds of properties and semantics. Said relationships need not necessarily be unique regarding their endpoints, but in such a case additional information is required in order to distinguish multiple edges connecting the same vertices in order to avoid ambiguity during ML as well as severe ontological errors. Graph machine learning (GML) is the field where computational combinatorics, algorithmic engineering, tensor algebra, geometry, neural networks, and ML intersect. Mining non-trivial knowledge relies heavily on graph properties such as the degree distribution, the primary eigenvector, the fundamental circuits, the adjacency matrix and the various factorizations thereof like the spectral and the polar factorizations, and patterns such as bridges and articulations. Furthermore, large scale graphs in the real world tend to exhibit properties such as recursive or modular community structure, self similarity, which allows scaling through natural and simple algorithmic mechanisms and has connections to fractals,

power laws in the degree distribution, and a multitude of connectivity patterns. Figure 1.4 is an emphasis of the self similarity of large graphs. The aforementioned properties are critical in designing advanced graph mining algorithms as efficient parallel miners, structurally aware smart agents which can freely circulate in graphs, and reliable termination conditions for iterative processed can be developed.

Given that the volume of real life graphs can reach Internet scale in terms of the size of vertices and edges, it follows immediately that appropriate digital infrastructure for storage and infrastructure should be developed. The advent of NoSQL databases at the beginning of the current decade was a major milestone as it marked in the database universe a shift from the relational paradigm, which has been designed for tabular data, to other ones. Already NoSQL databases such as Neo4j have been a mainstay for a broad spectrum of applications such as social network analysis, large omics analysis, ontological mining, geospatial and geolocation applications, as well as enterprise realm configuration analysis. In such databases graphs can be loaded as JSON or BSON documents, RDF triplets, or JSON-LD files to name just a few formats. Although graph models have their own distinct advantages for representing link oriented data, other data types like tabular structured and documents are also common and, thus, efficient transformation algorithms are necessary. Moreover, any reasonably correctly designed graph model should capture and expose the complexity of any specific graph object in the database as well as that of the underlying domain.

Humanistic data not only are important by definition, but also are gradually becoming the prevalent data category in contemporary knowledge-based civilizations. They have a plethora of important properties including a potentially enormous size, increasingly quicker generation rates, and maybe corrupted or incomplete entries. Despite these challenges mining these data is bound to provide crucial knowledge across numerous aspects of everyday life smart cities, digital health, and electronic financial transactions. Thus, algorithms handling humanistic LOD should take these properties into consideration and, wherever possible, turn them to their advantage. To this end DL is critical as will be explained later on.

This work begins with chapter 1, the general introduction of this dissertation, where the basic properties of humanistic data and these of large graphs are briefly overviewed. The formal description of humanistic data used here is given in definition 1.1 and their nature is specified. Moreover, definition 1.3 outlines digital awareness, whereas the research goals are given in objective statement 1.1. Digital awareness depends heavily on the timely extraction of critical knowledge from humanistic data in general and LOD in particular and aims at providing a deeper insight into the current situation to multiple stakeholders, ranging from on the spot operators to investors, of operations involving humanistic data like Industry 4.0.

In chapter 3 the architecture of tensor stack network (TSNs) is introduced and two applications thereof are given. TSNs rely on clustering multiple feedforward neural networks FFNNs in their layers and cross-training them in parallel in such a manner that FFNNs packed in the same layer learn during distinct and dedicated phases of the training process not only from the backpropagated errors of their own neurons, but also from those of the FFNNs in the same cluster. The first TSN application is the assessment of the structural resilience of large graphs, which is defined as the ability to keep its connectivity when a fraction of its edges is permanently lost. This is a stricter constraint than these imposed on temporal graphs as in the latter edges are only temporary deleted. Resilience is a major operational requirement of current systems working on LOD including telecommunication systems, especially mobile and wireless ones. Two TSN architectures each with two synaptic weight initialization strategies for a four combinations in total are trained with numerous synthetic Kronecker graphs containing a multitude of structural patterns for resilience classification. The reason behind selecting TSNs is that they can capture the inherently higher

order patterns in graphs. The results indicate a high correlation with the Estrada index, the odd Estrada index, and the clustering coefficient. The second application of TSNs is estimating the topological correlation coefficient between any two graphs based on their adjacency matrices when the latter are compressed with the two-dimensional discrete cosine transform (DCT2). This is tantamount to enriching the information latent in the compressed graphs with knowledge from the structural patterns of similar graphs. The results obtained from three benchmark graph sequences are encouraging as the mean square error (MSE) was low and so was the complexity once the TSNs were trained. A variety of nonlinear activation function was tested with the logistic function systematically outperforming every other one, probably because of its monotonicity, simple form, and bounded derivative. An additional advantage of the proposed methodology is that it is generic enough and does not depend on the particular semantics of the underlying domain.

Chapter 4 describes a self organizing map (SOM), which was a rarity among neural network architectures when it was proposed as it is designed for unsupervised learning tasks, for fMRI image clustering. Images were taken from an open source neuroimaging dataset. The novelty of the proposed SOM is that it operates with a tensor distance metric specifically and explicitly tailored for evaluating image similarity. SOMs, alternatively called cognitive maps, have a unique iterative training process which relies heavily on adaptively moving connections between two spaces of vastly different dimensions in such a way that the major topological properties of the higher dimension space are up to an extent preserved in the lower dimension space. Crucial to the formulation of clusters in the lower dimension space is a modified Hebbian rule which locally diffuses a fraction of the reward the SOM gains during an iteration. The way SOMs operate eliminates the need for elaborate optimization schemes as only a large number of pairwise distances needs to be computed, possibly in parallel when the underlying computational infrastructure allows so. Another advantage of SOMs is that the resulting clustering is segmentwise smooth and this local continuity allows easy visualization. This chapter focuses on a tensor distance metric for the input vector space which adds flexibility in two ways. First, clustering can be extended to higher order data such as images, graphs, matrices, and time series. Second, the resulting clusters are unions of arbitrary shapes instead of fixed ones such as rectangles in case of ℓ_1 norm or circles in case of ℓ_2 norm. The figures of merit were the topological error and the decay rate of the average location difference of the centroid clusters. Specifically, a power law and a cutoff power law were fitted to the latter and the exponent was used as the decay measure.

The focus of the chapter 5 is finding affective communities in Twitter graphs with graph neural networks (GNNs). The latter is an emerging class of connectionist models taking full advantage of the knowledge they progressively and iteratively extract from the interaction patterns between entities coming from an underlying domain. Depending on their working configuration GNNs aggregate local state information in the form of ground truth vectors to obtain robust estimates of global properties. As the knowledge contained in graphs is inherently and strongly of higher order nature, GNNs are very well capable of efficiently carrying out dimensionality reduction tasks provided that the appropriate aggregator is selected. One such task is discovering vertex sentiment polarity in the form of labels based on a representation combining structural, functional, and affective aspects. The performance of the proposed GNN has been assessed through two benchmark graphs extracted from political Twitter with a topic sampling strategy. The two topics of interest were the celebrations of the Greek 1821 Independence Revolution and the US 2020 Presidential Elections. Affective community structure was easier to find in the former graph as most accounts were tweeting about benign topics with a positive stance. On the other hand, the latter graph was echoing the polarized political climate of the time with numerous inflammatory comments, heated retorts, and considerable irony. These findings were also carried out to a GNN model where the edges between two vertices, representing the communication between them, also have attributes influencing training.

Chapter 6 concludes this dissertation by presenting the conclusions with respect to expressive power, computational complexity, space complexity, and digital awareness. Additionally, the research papers by the author are overviewed based on the year they were published and their primary field.

Finally, section A' is the appendix where the main world events which took place while the author was working towards the PhD are listed.

2.2 Keywords

Tensor algebra, multilinear algebra, higher order data, knowledge mining, tensor stack networks, self organizing maps, graph neural networks, dimensionality reduction, affective community discovery, topological properties, topology, graph geometry, graph connectivity, graph resilience, iterative learning, training process, clustering, classification, unsupervised learning, humanistic data, link oriented data.

Part II

Contributions

Chapter 3

Tensor stack networks

Your brain does not manufacture thoughts.

Your thoughts shape neural networks.

Deepak Chopra (1946 –)

3.1 Chapter introduction

SINCE their original inception in the form of perceptron in 1957 the broad class of neural network architectures constitute an integral part of both machine learning (ML) and deep learning (DL). Various members of this class, each with its own unique properties and advantages, have been proposed for efficiently performing massive, distributed, parallel, and highly non-linear processing and they have been successfully applied to problems previously thought to be intractable or without an elegant solution. The long list of these problems includes indicatively the following, but it is very far from being exhaustive:

- ♣ The exclusive or (XOR) separation problem. This is one of the most well-known problem where the two classes have non-linear separation bounds. Historically, it was one of the first problems where neural networks were applied to and demonstrated unequivocally their power.
- ♣ Adaptive signal processing. It is an important field in signal processing where the system or signal under study changes or evolves over time. Typically the former has stochastic origins, whether the latter can be driven by deterministic or stochastic mechanisms with trajectories in state spaces.

- ♣ Non-linear and noisy telecommunication channel equalization. It is an important problem especially in contemporary mobile networks and satellite connections where interference is stochastic and possibly strongly non-linear, while a large number of entities may well be involved.
- ♣ Time series prediction and smoothing. The former consists of efficiently estimating future values of an evolving time series, whereas in the latter the entire signal is known and outliers are processed according to certain regularity and continuity criteria to ensure better compression potential.
- ♣ Time series interpolation. Filling missing or corrupted values in time series is rapidly re-emerging as their size increases while tabular models gradually give way to graph, long column, and more recently vector models in order to find patterns in LOD and optimize operations.
- ♣ Large database clustering. Scalability has always been a major issue in databases and much effort in their design, particularly in data representation and its associated operations, has been diverted to handle data segments which fit only in secondary memory and transfers play a critical role.
- ♣ Classification. Annotated datasets are crucial for this task and they are harder to be created and curated as data sources generate at a quicker rate more unstructured and semi-structured data. Neural network architectures can efficiently handle big and complex datasets and can tie in with streams.
- ♣ Pattern recognition. One of the mainstays of ML and DL efficiently locating patterns which may be slightly altered or, in the graph case, not perfectly aligned. Efficient parallelism is required in most cases, and graph matching and searching algorithms frequently offer massive parallelism.

Despite the deceptive simplicity of the conceptual and functional description of most neural network architectures, the computational power of many of these architectures combined with the proper configuration and attribute engineering is considerable. This can be attributed to various factors, each of which contributes in its own way to the overall learning process, including non-linearity in both connectivity patterns as well as in processing, scalability, potential feedback and feedforward loops adding memory and recurrent information flow, neuron specialization, and decentralization. Examples of neural networks include the multilayer perceptron (MLP) [ATAA24], recurrent neural networks (RNNs) [QYZ24], convolutional neural networks (CNNs) [JT24], and self organizing maps (SOMs). The latter will be the focus of chapter 4 in conjunction with tensor distance metrics coupled with suitable weight schemes.

Tensor stack networks (TSNs) have been originally proposed for voice prediction over very long Markov chains [LCL⁺24] as well as for information retrieval (IR) scenarios [WPPQA24]. Structurally TSNs consist of layers which contain neuron tensors with information flowing serially from the input layer to the output. In the TSN architectures examined here output feedback from subsequent layers does not exist and results from each layer are linearly propagated. Within each layer processing is synchronized. The twofold algorithmic cornerstone forming the heart of the power of TSNs is the following [HHZ⁺23]:

- ♣ TSNs have the ability to natively handle multidimensional data such as images and graphs. This is crucial since implementation is simplified, results are more interpretable, and information loss is prevented in sharp contrast to traditional DL methods where vectorizing multidimensional data results in ignoring data point proximity or topological information. The latter is paramount in a number of fields and preserving it significantly increases the efficiency of schemes like TSNs.
- ♣ In TSNs processing neurons within a specific tensor processing layer are designed to learn not only from their own computational errors but also from the weighted ones of the peer neurons belonging

to the same processing layer. This in most cases leads naturally to smoother correction factors added to the synaptic weights of TSN neurons akin to the way the batch least mean squares (LMS) algorithm approximates the stochastic descent in order to train an adaptive system.

In this work the inner TSN mechanisms as well as the potential of this architecture in terms of configurability and expressive power are explored in the context of two fundamental graph mining problems, namely that of evaluating the structural resiliency of a graph based on a plethora of low order connectivity patterns and that of recovering compressed graphs with the two-dimensional discrete cosine transform (DCT2) using information from similar one in a graph sequence. Both have applications across a wide array of fields including SNA, omics and brain connectivity networks, long supply chains and logistics, computer communication networks, and transportation scenarios. Observe that higher order patterns deriving directly from the multitude of relationships inherent in humanistic data play a major role. Additionally, in the above fields graphs are appropriate models succinctly summarizing the dynamics thereof.

The two applications of TSNs, one pertaining to graph mining shown in section 3.4 and one to SNA shown in section 3.5, are described in the research objective statement 3.1 as follows. Both rely heavily on the scaling properties of TSNs and their native ability to handle two-dimensional data. The topological patterns and the activation functions have been tailored to the underlying field as well as to the particular task at hand. Despite their size and need for sparse algorithms, tensors offer advantages which cannot be obtained by conventional neural network architectures which vectorize input data. This results in the loss of spatial knowledge which indirectly codifies patterns not found in the numerical values of the input.

Objective 3.1 (Chapter research objectives) *The research contribution of this chapter is twofold:*

- ♣ *A TSN specifically designed to assess the structural resilience of a graph based on a set of fundamental connectivity patterns. The evaluation capabilities of this TSN are a function of the information available about the graph under study in terms of said set as well as on its architecture.*
- ♣ *A second TSN tailored for decompressing Twitter graphs which have been compressed with DCT2 relying on the compressed graph in the form of DCT2 coefficients of the original adjacency matrix as well as on structural information from similar graphs.*

Two important common points in the above TSN architectures are that they operate strictly on the graph structure, so graph functionality is excluded in this chapter in sharp contrast to chapter 5, and that they can incorporate external information related to the graphs under study.

The remainder of this chapter reflects its duality and it is structured as follows. In section 3.2 the recent scientific bibliography regarding tensors, TSNs, tensor signal processing, and graph resilience is briefly reviewed. For a literature review of SNA please see section 5.2. Preliminary material including tensor basic operations and least squares fit of power laws are presented in section 3.3. The TSN developed specifically to evaluate structural graph resiliency is described in section 3.4. Moreover, the TSN designed to recover compressed graph sequences based on the DCT2 coefficients of the graph adjacency matrix is explained in section 3.5. Finally, for the general conclusions of this dissertation please see chapter 6.

3.2 Previous work

Tensors [BGL15; BPT15] are direct generalizations of vectors and matrices found in linear algebra in more than two dimensions [Kol09; PF15], with dimensions of three [CQW23] and four [WYL⁺23; SZT⁺23]

being very common choices for a broad spectrum of applications [PPF14; WHH⁺23]. They have been used to detect rotation [YZL⁺24], to implement human activity models [CL23]. Tensor dimensions, also known as *modes* [BJT24; FYL⁺24] or infrequently as *fibers* [SS08] are central in modelling higher order data [W⁺18]. Tensor operations include a variety of norms, such as the Frobenius norm [LWS⁺23], which can be used for tensor completion [TYLJ24]. Other major tensor operations are contractions [SNAC16], especially for sparse tensors [XYC⁺24; KS24], and transforms such as Kruskal factorization [WOV23; BC24] and Tucker decomposition [LLWA24; PLH⁺24; CCHN24] which are generalizations of the singular value decomposition (SVD) to higher dimensions [HSZ23]. TensorFlow¹ [Pat23], covered in more detail in section 6.4.7, is a popular tensor package with a Python API [Aba16b; Aba16a]. Among others it has been used for data visualization [Won18], Gaussian processes [Mat17], and for the implementation of CNNs [RSJ23]. Software vulnerabilities in applications built on TensorFlow are examined in [FD23]. Python package *numpy*² has limited tensor handling capabilities as well as some built-in methods [HT24]. Julia³ is a Java virtual machine (JVM)-based language specifically designed for data intensive applications [Chu23; MO23]. As such, Julia has packages for tensor handling such as the ones described in [Pat24; FM24].

TSNs have been originally applied to speech recognition [DHY12; DY11; Den13], IR [DHG13], and problems with large instances [HDY13] and considerable attribute variability [Den14]. TSNs consist of clusters of feedforward neural networks (FFNNs) where each network in the cluster receives a synaptic weight update based not only on its own local error gradient but also on that of other networks in the cluster [HDY12]. Also, stacking can avoid overfitting in neural network training [GB13]. Alternatively or in conjunction with stacking Tikhonov regularization [GHO99] can be used during the training process which tantamounts to learning in the presence of noise [Bis95]. Given the potentially huge number of synaptic weights and inputs, numerical issues should be considered and appropriately countered [Pri91]. As their name suggests, tensors are their basic building blocks [BFZM18] and can naturally represent their learning layers [LCL⁺24]. Engineering scenarios for TSNs include face recognition [HHY⁺17], image classification based on sparse stacked representations [LCY15], blind sound classification [KGN⁺19], speech recognition with very long vocabularies [YDS13], and gesture interpretation [BGM23]. For a TSN toolkit see [PH15].

Tensor signal processing (TeSP) covers a broad spectrum of topics such as tensor spectral clustering [BGL15], efficient matrix classification [Sch15], feature extraction [BPT15], compressed sensing [PWD13], and clustering through higher order data tensor decomposition [LB15]. Applications include face recognition [OHCB24], anomaly detection [PPF14], network monitoring [KSM15], and object classification through multilinear discriminant analysis (MLDA) [ZWSS15] or tensor discriminant analysis (TDA). Dedicated tensor processing units (TPUs) [PAEHC⁺24] have been proposed and implemented for tasks such as matrix multiplication [GBH⁺23]. Hypergraph signal processing based on *t*-product tensor decompositions is the focus of [PPLA23]. Architectures for TeSP are explored in [Den14]. Popular DL software is MLlib for the Spark ecosystem [Men16], *keras*⁴ [GP17], *Theano*⁵ [Bea11], *Torch*⁶ [CKF11], *TensorLab*⁷ [VDDL16], *Caffe*⁸ [Jia14], and *MatConvNet*⁹ [VL15]. As the field evolves, more tools are developed, each with more

¹<https://www.tensorflow.org>

²<https://numpy.org>

³<https://julialang.org>

⁴<https://keras.io>

⁵<https://github.com/pymc-devs/pytensor>

⁶<http://torch.ch>

⁷<https://www.tensorlab.net>

⁸<https://caffe.berkeleyvision.org>

⁹<https://www.vlfeat.org/matconvnet>

sophisticated capabilities progressively.

Other neural network architectures are spiking neural networks [GSK24]. This class of neural networks simulates the physical activity of the neurons comprising the human brain [RMC⁺24], which operate on firing patterns which are correlated to external stimuli or to motor, visual, sensory, or structural knowledge about the outer world [HEW24]. Other recent popular architectures include RNNs [QYZ24] which have memory in the form of connections [DWA⁺24] between different neuron layers also with adjustable weights [XGS⁺24]. The primary characteristic is that connections in RNNs form temporal sequences which also serve as mineable patterns [DJP⁺24]. Applications include affective document classification [NPN⁺24], human activity recognition [CWYH24], and geometric matrix completion [MBB17]. CNNs [ZWZ⁺24] rely heavily on local weighted and delayed feedback [YHY⁺24] as well as on local convolution functions allowing thus processing neurons to specialize on different parts of the input data [YSA24]. Thus meaningful input space segmentations can be achieved [LL24]. Predicting the physical trajectory of social agents, whether humans or autonomous vehicles, based on local smoothness constraints [MQEC20], quantum optics [GPZZ24], and sound classification based on attributes extracted by signal processing techniques [KGN⁺19] are three applications. However, CNNs are quite popular in computer vision [KAW⁺24] and image processing [XLY⁺24]. Lastly, SOMs are covered in detail in chapter 4.

Graph resilience has been the focus of [IW11] as well as of [NG90] among many others. Analytic resilience metrics are proposed in [LCW05] for networks of evolving topology and in [VM03] for graphs with degree correlations. A comparative study of resilience metrics are [AS15] and [PYHH21]. Similar algorithmic schemes to the one presented in this chapter employ an LSTM for assessing graph resilience [TSYZ19]. Resiliency considerations for evolving graphs are given in [MG20]. An overview of resilience design strategies is given in [FYK⁺22]. Centrality as resilience is examined in [WMK⁺21]. Applications cover a wide array of technological and financial sectors such as long supply chains [TZC19], transportation networks [WX20; SAZ⁺20], brain networks [BS13], social media [BBM20], IoT [CFKS20], electric power grids [YCY21; LCCJ20; BPWV20], and manufacturing [Kus20].

3.3 Introductory material

3.3.1 Overview

The material presented in this section applies to the three technical chapters of this dissertation, namely chapters 3, 4, and 5. Specifically, tensors constitute a fundamental part of TSNs presented in this chapter as well as of the distance metric proposed in chapter 4. Power laws are critical for the analysis of the results of chapters 3 and 5. The connecting tissue so to speak of this work is explained in this section.

3.3.2 Tensors

Tensors, as given in definition 3.1, play an instrumental role in higher order knowledge mining in general as well as mining LOD in particular. If $\mathbb{S}_1, \dots, \mathbb{S}_p$ are p not necessarily distinct vector spaces of dimensions $I_1, \dots, I_p$ respectively, then T belongs to their Cartesian product. Intuitively speaking initially, a tensor T of the p -th order –or simply of order p – is a higher order vector indexed by p integers. In this dissertation the focus will be on real tensors, but depending on the underlying application this may differ.

Definition 3.1 (Tensor) A p -th order tensor is a linear mapping coupling the p vector spaces $\mathbb{S}_1, \dots, \mathbb{S}_p$ which may not be necessarily distinct simultaneously. A real tensor belongs to $\mathbb{R}^{I_1 \times I_2 \times \dots \times I_p}$.

From a probability theory perspective, tensors can be divided into the following categories depending on the nature of their elements. Please note that this taxonomy is far from being unique given the size and structural complexity of tensors as well as the wide range of their applications.

- ♣ Deterministic. The tensor entries are numerical values and do not vary over time.
- ♣ Individually stochastic where element are stochastically independent. In this case, each element has its own probabilistic properties and they may or may not be identically distributed.
- ♣ Jointly stochastic where elements have dependencies. The latter can include every element. Alternatively, elements are partitioned based on their mutual dependencies creating communities.

Additionally, another tensor categorization is the following. This is different from the preceding one and in fact both taxonomies can be combined. Thus, a deterministic tensor can have evolution mechanisms.

- ♣ Static. Once they are created, these tensors do not change.
- ♣ Evolving. They are generated by a mechanism and their elements change in various rates.

Observe that from definition 3.1 it follows directly that the size of a tensor increases multiplicatively as more dimensions are added. This leads to the elements of a tensor quickly reaching a number which is not directly manageable by in most if not all engineering scenarios involving big data. In turn this is a considerable constraint among others in computing the result of equation (3.2) of definition (3.2). At this point equation (3.1) needs to be introduced first for conceptual clarity and notational simplicity.

$$Q[P] \triangleq \begin{cases} 1, & \text{predicate } P \text{ is true} \\ 0, & \text{predicate } P \text{ is false} \end{cases} \quad (3.1)$$

With the indicator function of equation (3.1) defined, which connects the domain of logic and proposal evaluation to the numeric domain, a critical tensor metric can now be given. The tensor sparsity of definition 3.2 is the number of non-zero elements of a tensor to the product of its dimensions. For deterministic tensors it is a measure of the number of potentially interesting elements, whereas for stochastic tensors it is a realization approximation based on ergodicity assumptions of an element being non-zero.

Definition 3.2 (Tensor sparsity) The sparsity of a tensor $T \in \mathbb{S}_1^{I_1} \times \mathbb{S}_2^{I_2} \times \dots \times \mathbb{S}_p^{I_p}$ denoted by ρ_0 is defined as the fraction of the number of zero entries to the total number of elements as shown in (3.2).

$$\rho_0 \triangleq \frac{\sum_{i_1=1}^{I_1} \dots \sum_{i_p=1}^{I_p} Q[T[i_1, \dots, i_p] = 0]}{\prod_{k=1}^p I_k} \quad (3.2)$$

However intuitive it is, equation (3.2) is prone to numerical errors due to finite numerical precision, roundoff errors, catastrophic cancellation, and limitations in the implementation of floating point operations. These are aggravated by the potentially big number of variables involved. This dictates the need for a robust tensor sparsity definition. One such well-known option is definition 3.3.

Definition 3.3 (Effective tensor sparsity) *The effective sparsity $\rho(\tau_0)$ with respect to a strictly positive threshold τ_0 of a tensor $T \in \mathbb{S}_1^{I_1} \times \mathbb{S}_2^{I_2} \times \dots \times \mathbb{S}_p^{I_p}$ is the fraction of the total number of tensor elements whose absolute value with respect to an appropriately defined metric in the respective vector space equals or is below τ_0 to the total number of tensor elements as shown in equation (3.3) below.*

$$\rho_0(\tau_0) \triangleq \frac{\sum_{i_1=1}^{I_1} \dots \sum_{i_p=1}^{I_p} Q[|T[i_1, \dots, i_p]| \leq \tau_0]}{\prod_{k=1}^p I_k}, \quad \tau_0 > 0 \quad (3.3)$$

For stochastic tensors an alternative to effective sparsity is the Chebyshev inequality of equation (3.4) which leads to power law bounds and it is a second order inequality. Assuming that every element of the tensor T is stochastically independent and identically distributed with mean μ_0 and variance σ_0 , then an upper bound for the probability of a tensor element exceeding a threshold is the following.

$$\text{prob} \{ |T[i_1, \dots, i_p] - \mu_0| \geq k\sigma_0 \} \leq \frac{1}{k^2} \quad (3.4)$$

However, the aforementioned power law bound may be insufficient or very loose. The latter may especially be true if the distribution has such a shape which can yield sharper bounds. An important such case is the Gaussian distribution which yields exponential bounds through the standard error function.

For deterministic tensors various heuristic schemes also exist. For instance, the numerator of equation 3.3 can be approximated by efficient set cardinality estimators. Alternatively, stochastic approximations exist relying on large sum heuristics, random walks, or the strong or weak law of large numbers.

Similarly to the vector and matrix norms in linear algebra there exist tensor norms. One of the most common ones is the Frobenius norm of definition 3.4 which has a number of useful properties.

Definition 3.4 (Frobenius norm) *The Frobenius norm of a p -th order tensor $T \in \mathbb{S}_1^{I_1} \times \mathbb{S}_2^{I_2} \times \dots \times \mathbb{S}_p^{I_p}$ denoted by $\|T\|_F$ is defined as the square root of the sum of its squared elements where the absolute value is defined suitably over the appropriate vector space as shown in equation (3.5).*

$$\|T\|_F \triangleq \left(\sum_{i_1=1}^{I_1} \dots \sum_{i_p=1}^{I_p} |T[i_1, \dots, i_p]|^2 \right)^{\frac{1}{2}} \quad (3.5)$$

As with tensor sparsity before, the computation of the Frobenius norm following definition (3.5) requires linear time with respect to the total number of elements of the tensor. Again efficient heuristics exist for deriving tensor norms, primarily based on the Hager method for deterministic norms or on chi square schemes for stochastic tensors, the delta rule, or sums of Rice or Rayleigh distributions.

The tensor Frobenius norm has the following properties. Some of the most important ones are:

- ♣ It is everywhere differentiable. This allows for the design of iterative schemes along the direction of the steepest descent algorithm. Also it allows the derivation of upper bounds for estimator errors.
- ♣ It is absolutely scalable in the sense that $\|\alpha T\|_F = |\alpha| \|T\|_F$ where α is a scalar. This indicates that critical tensor properties do not change with scaling. This adds robustness to the tensor definition.
- ♣ It is subadditive in the sense that for two tensors $\|T_1 + T_2\|_F \leq \|T_1\|_F + \|T_2\|_F$. This allows the estimation of large tensor norms quickly through the addition of random perturbations.
- ♣ When $\|T\|_F$ is zero, then the tensor is zero and vice versa. This paves the way among others for termination criteria of iterative algorithms in conjunction with the differentiation property.

Along a similar line of reasoning the root mean square norm ρ_{rms} of a tensor can be defined. It is formally given in definition 3.5. This quantity has been inspired by the root mean velocity of molecules in a gas and it represents the average energy of a tensor element, which for humanistic applications is frequently a useful activity indicator such as the amount of hashtags or tweets in a given Twitter (**not** X) graph. Similar definitions can be made for each individual tensor dimension or even for a tensor block.

Definition 3.5 (Tensor root mean square norm) *The root mean square energy of a p -th order tensor $T \in \mathbb{S}_1^{I_1} \times \dots \times \mathbb{S}_p^{I_p}$ is defined as the square root of the sum of the squared magnitude of its elements to the total number of its elements as shown in equation (3.6) for a suitably defined absolute value metric.*

$$\rho_{rms} \triangleq \sqrt{\frac{\sum_{i_1=1}^{I_1} \dots \sum_{i_p=1}^{I_p} |T[i_1, \dots, i_p]|^2}{\prod_{k=1}^p I_k}} = \frac{\|T\|_F}{\sqrt{\prod_{k=1}^p I_k}} \quad (3.6)$$

The tensor addition is the most basic operation involving any two tensors which must have identical dimensions as it is shown in definition 3.6. This allows among others the creation of tensor spans on an algorithmic level and the merging of information in TSNs on a data processing one.

Definition 3.6 (Tensor addition) *The tensor addition of two tensors T_1 and T_2 both necessarily having the same order and the same dimensions is defined elementwise as in equation (3.7) below.*

$$(T_1 + T_2)[i_1, \dots, i_p] \triangleq T_1[i_1, \dots, i_p] + T_2[i_1, \dots, i_p] \quad (3.7)$$

The next operation is the tensor multiplication along the k -th dimension $\times_k$. Unlike the tensor addition the two operands need only be compatible in the k -th dimension and may very well be quite different in the remaining ones. Also, both tensors can very well have different orders. Definition 3.7 follows.

Definition 3.7 (Tensor multiplication) *The tensor multiplication of two tensors possibly having different orders p and p' and coupling different spaces but having the same number of elements along the k -th dimension is elementwise defined as in equation (3.8). The result may be of different order than that of both operands.*

$$(T_1 \times_k T_2)[i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_p, i'_1, \dots, i'_{k-1}, i'_{k+1}, \dots, i'_{p'}] \triangleq \sum_{i_k=1}^{I_k} T_1[i_1, \dots, i_p] T_2[i'_1, \dots, i'_{p'}] \quad (3.8)$$

Finally, the Kronecker tensor product for two tensors can be defined. In this work the Kronecker product for a pair of matrices $A \in \mathbb{R}^{p \times q}$ and $B \in \mathbb{R}^{u \times v}$ as shown in definition 3.9 will be used.

Definition 3.8 (Kronecker tensor product) *The Kronecker tensor product for any two matrices A and B of not necessarily the same dimensions is defined as in equation (3.9) below.*

$$A \otimes B \triangleq \begin{bmatrix} A[1,1]B & A[1,2]B & \dots & A[1,q]B \\ A[2,1]B & A[2,2]B & \dots & A[2,q]B \\ \vdots & \vdots & \ddots & \vdots \\ A[p,1]B & A[p,2]B & \dots & A[p,q]B \end{bmatrix} \in \mathbb{R}^{pu \times qv} \quad (3.9)$$

Since a matrix is a tensor of order two, the definition 3.8 can be directly generalized to tensors of higher order as well as to tensors of different orders. Observe that the dimensions of the resulting matrix have increased multiplicatively with respect to the dimensions of the original matrices.

Definition 3.9 will be used in chapters 3 and 4 to efficiently reshape tensors.

Definition 3.9 (Tensor vector operator) *The tensor vector operator $\text{vec}(\cdot)$ maps a p -th order tensor $T \in \mathbb{S}_1^{l_1} \times \dots \times \mathbb{S}_p^{l_p}$ to a vector v by stacking from top to bottom and from left to right the elements of T .*

3.3.3 Least squares power law fit

Power laws appear in many places in large graphs of an important class named *scale free graphs*. Such laws, in its simplest form a monomial or a variation thereof, take different forms as seen in table 3.1 depending on the underlying field and consequently have different properties. In the case of scale free graphs power laws appear in the degree distribution, the number of triangles distribution, and the primary eigenvector components. The positive exponent γ_0 plays a central role, similar to the Lyapunov exponent: When the exponent is above three, then the deterministic mean and variance of the power law are finite. When the exponent lies between two and three, then only the mean is finite. Otherwise both are infinite.

Table 3.1: Power laws.

Name	Form	Linearized
Power law	$f_k = \alpha_0 k^{-\gamma_0} = \alpha_0 \exp(-\gamma_0 \ln k)$	$\ln f_k = \ln \alpha_0 - \gamma_0 \ln k$
Cutoff power law	$f_k = \alpha_0 k^{-\gamma_0} e^{-k/\beta_0} = \alpha_0 \exp(-\gamma_0 \ln k - k/\beta_0)$	$\ln f_k = \ln \alpha_0 - \gamma_0 \ln k - k/\beta_0$
Squared power law	$f_k = \alpha_0 \exp(-\gamma_0 \ln^2 k)$	$\ln f_k = \ln \alpha_0 + \gamma_0 \ln^2 k$
Shifted power law	$f_k = \alpha_0 \exp(-\gamma_0 \ln^2 k - \beta_0)$	$\ln f_k = \ln \alpha_0 - \beta_0 + \gamma_0 \ln^2 k$
Delta power law	$f_k = \alpha_0 \exp(-\gamma_0 \ln^{1+\delta_0} k)$	$\ln f_k = \ln \alpha_0 + \gamma_0 \ln^{1+\delta_0} k$

A probabilistic model related to power laws is the DGX shown in (3.10). Notice its similarity to the Gaussian kernel, especially the dependency from a location and a scale parameter. The major exception is that of the logarithm in the exponent which essentially leads to a polynomial. However, as a direct result of this a different normalization constant is necessary which is not as easy to compute as in the Gaussian case. The DGX distribution is the inspiration of the squared power law in table 3.1.

$$f_k \triangleq \frac{A(\mu_0, \sigma_0)}{k} \exp\left(-\frac{(\ln k - \mu_0)^2}{2\sigma_0^2}\right)$$

$$A(\mu_0, \sigma) \triangleq \left(\sum_{k=1}^{+\infty} \frac{1}{k} \exp \left(-\frac{(\ln k - \mu_0)^2}{2\sigma_0^2} \right) \right)^{-1} \quad (3.10)$$

In the remaining of this subsection the exponent γ_0 will be estimated for two scenarios. The first case will be the power law shown in the first row of table 3.1. Assuming there are n available observations, then collecting them using the linearized form of the power law yields the linear system of equation (3.11).

$$\underbrace{\begin{bmatrix} \ln f_1 \\ \ln f_2 \\ \vdots \\ \ln f_n \end{bmatrix}}_{\mathbf{f}_p} = \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & -\ln 2 \\ \vdots & \vdots \\ 1 & -\ln n \end{bmatrix}}_{\mathbf{X}_p} \underbrace{\begin{bmatrix} \ln \alpha_0 \\ \gamma_0 \end{bmatrix}}_{\mathbf{b}_p} \Leftrightarrow \mathbf{f}_p = \mathbf{X}_p \mathbf{b}_p \quad (3.11)$$

As the coefficient matrix $\mathbf{X}$ is not square, one possible solution is to multiply both sides of (3.11) with its transpose. The least squares (LS) solution of (3.11) $\hat{\mathbf{b}}_p$ and the projected values based it $\hat{\mathbf{f}}_p$ are given in (3.12). The projection matrix $\mathbf{H}_p$ can be used to generate regressed values when new data are available.

$$\hat{\mathbf{b}}_p \triangleq \left(\mathbf{X}_p^T \mathbf{X}_p \right)^{-1} \mathbf{X}_p^T \mathbf{f}_p \quad \text{and} \quad \hat{\mathbf{f}}_p \triangleq \mathbf{X}_p \hat{\mathbf{b}}_p = \mathbf{X}_p \left(\mathbf{X}_p^T \mathbf{X}_p \right)^{-1} \mathbf{X}_p^T \mathbf{f}_p = \mathbf{H}_p \mathbf{f}_p \quad (3.12)$$

By construction the two by two matrix $\mathbf{X}_p^T \mathbf{X}_p$ is symmetric with non-zero determinant.

$$\mathbf{X}_p^T \mathbf{X}_p = \begin{bmatrix} n & -\sum_{k=2}^n \ln k \\ -\sum_{k=2}^n \ln k & \sum_{k=2}^n \ln^2 k \end{bmatrix} \quad \text{and} \quad \det \left(\mathbf{X}_p^T \mathbf{X}_p \right) = n \sum_{k=2}^n \ln^2 k - \left(\sum_{k=2}^n \ln k \right)^2 \quad (3.13)$$

The LS estimate of γ_0 is computed as in (3.14).

$$\hat{\gamma}_0 = \frac{(\sum_{k=2}^n \ln k)(\sum_{k=1}^n \ln f_k) - n(\sum_{k=2}^n \ln k \ln f_k)}{n \sum_{k=2}^n \ln^2 k - (\sum_{k=2}^n \ln k)^2} \quad (3.14)$$

The mean square error (MSE) J_p between the actual and the projected values is given in equation (3.15).

$$J_p \triangleq \frac{1}{n} \left\| \mathbf{f}_p - \hat{\mathbf{f}}_p \right\|_2^2 = \frac{1}{n} \left\| \mathbf{f}_p - \mathbf{X}_p \hat{\mathbf{b}}_p \right\|_2^2 = \frac{1}{n} \left\| \mathbf{f}_p - \mathbf{H}_p \mathbf{f}_p \right\|_2^2 = \frac{1}{n} \left\| (\mathbf{I}_n - \mathbf{H}_p) \mathbf{f}_p \right\|_2^2 \quad (3.15)$$

The following properties hold for $\mathbf{H}_p$. In light of the above, the geometric meaning of J_p is that it is the component of the measurements which is perpendicular to the subspace spanned by the columns of $\mathbf{H}_p$.

$$\begin{aligned} \mathbf{H}_p^2 &= \mathbf{X}_p^T \left(\mathbf{X}_p^T \mathbf{X}_p \right)^{-1} \mathbf{X}_p \mathbf{X}_p^T \left(\mathbf{X}_p^T \mathbf{X}_p \right)^{-1} \mathbf{X}_p = \mathbf{X}_p^T \left(\mathbf{X}_p^T \mathbf{X}_p \right)^{-1} \mathbf{X}_p = \mathbf{H}_p \\ (\mathbf{I}_n - \mathbf{H}_p) \mathbf{H}_p &= \mathbf{H}_p - \mathbf{H}_p^2 = \mathbf{H}_p - \mathbf{H}_p = \mathbf{O}_n \end{aligned} \quad (3.16)$$

The relative residual B of (3.17) is the fraction of power of $\mathbf{b}_p$ captured by $\mathbf{H}_p$ of equation (3.12).

$$B \triangleq \frac{\left\| (\mathbf{I}_n - \mathbf{H}_p) \mathbf{b}_p \right\|_2}{\left\| \mathbf{H}_p \mathbf{b}_p \right\|_2} \quad (3.17)$$

The exponent γ_0 for the cutoff power law of table 3.1 will now be estimated in the LS sense in a similar way. Stacking the n available equations results in the system of the equation (3.18).

$$\underbrace{\begin{bmatrix} \ln f_1 \\ \ln f_2 \\ \vdots \\ \ln f_n \end{bmatrix}}_{\mathbf{f}_c} = \underbrace{\begin{bmatrix} 1 & 0 & -1 \\ 1 & -\ln 2 & -2 \\ \vdots & \vdots & \vdots \\ 1 & -\ln n & -n \end{bmatrix}}_{\mathbf{X}_c} \underbrace{\begin{bmatrix} \ln \alpha_0 \\ \gamma_0 \\ 1/\beta_0 \end{bmatrix}}_{\mathbf{b}_c} \Leftrightarrow \mathbf{f}_c = \mathbf{X}_c \mathbf{b}_c \quad (3.18)$$

Along a similar line of reasoning the LS solution of the linear system (3.18) is that of equation (3.19).

$$\hat{\mathbf{b}}_c = \left(\mathbf{X}_c^T \mathbf{X}_c \right)^{-1} \mathbf{X}_c^T \mathbf{f}_c \quad (3.19)$$

Now the three by three coefficient matrix $\mathbf{X}_c^T \mathbf{X}_c$ has the structure of equation (3.20).

$$\mathbf{X}_c^T \mathbf{X}_c \triangleq \begin{bmatrix} n & -\sum_{k=2}^n \ln k & -\sum_{k=1}^n k \\ -\sum_{k=2}^n \ln k & \sum_{k=2}^n \ln^2 k & \sum_{k=2}^n k \ln k \\ -\sum_{k=1}^n k & \sum_{k=2}^n k \ln k & \sum_{k=1}^n k^2 \end{bmatrix} \quad (3.20)$$

Matrix $\mathbf{X}_c^0$ is made by placing $\mathbf{X}_c^T \mathbf{f}_c$ as the second column of $\mathbf{X}_c^T \mathbf{X}_c$ as in (3.21).

$$\mathbf{X}_c^0 \triangleq \begin{bmatrix} n & \sum_{k=1}^n \ln f_k & -\sum_{k=1}^n k \\ -\sum_{k=2}^n \ln k & -\sum_{k=2}^n \ln k \ln f_k & \sum_{k=2}^n k \ln k \\ -\sum_{k=1}^n k & -\sum_{k=2}^n k \ln f_k & \sum_{k=1}^n k^2 \end{bmatrix} \quad (3.21)$$

The LS estimation of γ_0 is computed as in (3.22). The determinant of three by three matrices can be computed with the Sarrus rule, which is a special case of Leibniz formula, or with Laplace expansions.

$$\gamma_0 = \frac{\det(\mathbf{X}_c^0)}{\det(\mathbf{X}_c^T \mathbf{X}_c)} \quad (3.22)$$

Finally, the sum the power law can be bounded in a closed form by equation (3.23). Depending on the number of values to be added and the exponent γ_0 tighter bounds may well exist. Changing the integral bounds may increase the accuracy of the approximation, but their estimation is not easy.

$$\sum_{k=1}^{k_0} \alpha_0 k^{-\gamma_0} \approx \alpha_0 \int_1^{k_0} x^{-\gamma_0} dx = \alpha_0 \left(1 - \frac{k_0^{1-\gamma_0}}{1-\gamma_0} \right) \quad (3.23)$$

3.3.4 Activation functions

Activation functions are integral parts of TSNs and GNNs in chapters 3 and 5 respectively. In order for their respective training process to efficiently converge, the activation function $\varphi(\cdot)$ should comply to at least the following rules in addition to be suitable for the learning task at hand.

- ♣ A thresholding behavior akin to that of human neurons should be present, even in a soft form allowing $\varphi(\cdot)$ to exhibit strong non-linear behavior. This leads to compact architectures.
- ♣ A saturation behavior can prevent unbounded responses to isolated input values and thus exclude their possible propagation throughout the network. Thus convergence is not disturbed.
- ♣ The local derivative of $\varphi(\cdot)$ should be bounded and also it should be computed or estimated. This leads to smooth weight corrections as well as a total progressive convergence.

The sigmoid or logistic (lgi) function of (3.24) is perhaps one of the most well known activation functions and has a natural interpretation in the context of the class of Verhulst population models.

$$\varphi(x; \sigma_0) \triangleq \frac{1}{1 + e^{-\sigma_0 x}} = 1 - \frac{1}{1 + e^{\sigma_0 x}}, \quad \varphi: \mathbb{R} \rightarrow \sigma_0 I_0 \quad (3.24)$$

The hyperbolic tangent of (3.25) is non-linear and smooth. Notice that compared to the logistic function presented earlier the hyperbolic tangent can take negative values and thus it has different applicability.

$$\varphi(x; \sigma_0) \triangleq \tanh(x; \sigma_0) = \frac{e^{\sigma_0 x} - e^{-\sigma_0 x}}{e^{\sigma_0 x} + e^{-\sigma_0 x}}, \quad \varphi: \mathbb{R} \rightarrow \sigma_0 I_1 \quad (3.25)$$

The smooth rectified linear unit (sReLU) is lately becoming a common option shown in (3.26). As its name suggests sReLU is smooth and allows some flexibility in handling the vanishing gradients scenario.

$$\varphi(x; \sigma_0) \triangleq \ln(e^{\sigma_0 x} + 1), \quad \varphi: \mathbb{R} \rightarrow \mathbb{R} \quad (3.26)$$

The softmax function of (3.27) which is often used in order to generate ranking scores given a set of values $S \triangleq \{s_1, \dots, s_n\}$ in the context of neural networks yields a probability for s_k .

$$\varphi(s_k) \triangleq \frac{e^{s_k}}{\sum_{j=1}^n e^{s_j}}, \quad \varphi: S \rightarrow I_0 \quad (3.27)$$

The final option is the scaled polynomial constant unit (SPOCU). It consists of a scaled difference between the evaluations in two closely spaced points of an eighth degree polynomial kernel as shown in (3.28). The constant c imposes a hard upper limit on the neuron activation function value.

$$\varphi(x; \alpha_0, \beta_0, \gamma_0) \triangleq \alpha_0 \left(h\left(\frac{x}{\gamma_0} + \beta_0\right) - h(\beta_0) \right), \quad \alpha_0, \gamma_0 > 0, \beta_0 \in (0, 1)$$

$$h(x; c) \triangleq \begin{cases} c^3(c^5 - 2x^4 + 2), & x > c \\ x^3(x^5 - 2x^4 + 2), & 0 \leq x \leq c, \\ 0, & x < 0 \end{cases} \quad c \geq 1 \quad (3.28)$$

3.3.5 Learning rate

The learning rate μ is critical for the convergence of the training process of TSNs and SOMs in chapters 3 and 4 respectively. In the general case it depends on the epoch as well as on the iteration within a given epoch if the entire training dataset is available. However, more frequently it depends only on the epoch r in order to reduce design and algorithmic complexity. Since training takes place iterations, the epoch number will be in an iteration counter and not in a vector index. Common options include the following.

Picking a fixed constant μ_0 is the simplest strategy for the learning rate. When the input dataset is known and consists of vectors, then the optimal strategy is to set μ equal to twice the reciprocal of the maximum eigenvalue of the autocorrelation matrix. However, its computation is prohibitive.

Another option is the exponential decay of (3.29). Although it favors early epochs with respect to later ones, it does so in a rather unbalanced manner. Thus, the training process may terminate prematurely and it may yield less than satisfactory results as much input knowledge has not been mined.

$$\mu^{[r]} = \alpha_0 e^{-\gamma_0 r} \quad (3.29)$$

A mild improvement comes from the inverse logarithm learning rate of (3.30). The slow growth rate of the logarithm ensures some evolution over epochs, but this formula is useful for very large datasets and often the constant in the denominator needs to be small in order to compensate for this.

$$\mu^{[r]} \triangleq \frac{1}{a_0 \log(r) + \beta_0} \quad (3.30)$$

The next logical step is equation (3.31) where the inverse linear relationship leads to quicker evolution and does not have many of the numerical problems of the inverse logarithm resulting in a smoother learning process. Still, early epochs do not have sufficient weight and for large datasets it does not scale well.

$$\mu^{[k]} = \frac{1}{\alpha_0 r + \beta_0} \quad (3.31)$$

A direct variant of (3.31) is the inverse polynomial strategy of (3.32). Its decay rate is quite flexible and potentially smooth with the ability to adapt to a broad spectrum of datasets with a variety of characteristics, but determining the right coefficients is not easy and may require resorting to numerical methods.

$$\mu^{[k]} = \frac{1}{(\alpha_0 r + \beta_0)^{\gamma_0}} \quad (3.32)$$

Finally, the cosine decay of (3.33) provides a decent tradeoff between a smooth decay rate, rewarding early epochs, and having a low number of parameters to estimate. However, the maximum number of epochs or an estimation thereof is necessary. Otherwise, a hybrid scheme can be used.

$$\mu^{[k]} = \alpha_0 \cos\left(\frac{\pi r}{2r_0}\right), \quad 0 \leq r \leq r_0 - 1 \quad (3.33)$$

3.4 Graph resiliency evaluation with TSNs

3.4.1 TSN architecture

Graph resilience or *structural coherency* is a major design consideration across a wide array of technological fields with numerous applications. Edge loss, whether random or systematic, can be attributed to design oversights or can very well be the result of natural factors. In any case, resilience must be a design criterion of top priority. Evaluating or estimating resilience may potentially be computationally intensive as it frequently involves visiting or otherwise examining a prohibitively large number of edges or vertices, depending heavily of course on the structural coherency definition used as well as on the operational parameters plus any additional constraints. For instance, in definition 3.10 the fraction of the edges which may be deleted from a graph cannot be determined universally and depends on the underlying field.

Definition 3.10 (Graph resilience) *Resilience is the ability to maintain total connectivity or maintain the original number of connected components even after a fraction of edges is lost for an indefinite time.*

Graphs provide an excellent abstract model for the study of resilience. In this context the existence of an edge depends heavily on the local connectivity properties between its end vertices. Once the graph is derived, its resilience is evaluated by a tensor stack network (TSN). TSNs are an emerging DL classification methodology with typically a FFNN topology where input is either stacked vectors or matrices, such as images or oversampled data from multiple input and multiple output (MIMO) digital communication systems [L⁺16; Loy01; L⁺14; NLM13; BG06]. As it can be seen from figure 3.1, TSNs have a modular architecture based on layers with identical structure which can be added or removed at will, offering flexibility and dynamic management of computational resources. TSNs do two dimensional learning as neurons can adjust their synaptic weights both from neurons of the same sublayer as well as those in neighboring layers.

In this section two TSN architectures will be trained to evaluate graph resilience and their performance will be evaluated based on how well they can approximate the Estrada index [Est22; WPW⁺20], the odd Estrada index [EH10], and the clustering coefficient [BAS23] of a large number of synthetic graphs. They are major resilience indicators whose computation requires considerable resources.

Definition 3.11 (Estrada index) *Let $\{\lambda_k\}_{k=1}^n$ be the eigenvalues of a graph adjacency matrix M with $\lambda_1 > \lambda_2 \geq \dots \geq \lambda_n$. Then, the Estrada index E which evaluates graph resilience is defined as:*

$$E \triangleq \sum_{k=1}^n e^{\lambda_k} = e^{\lambda_1} + e^{\lambda_2} + \dots + e^{\lambda_n} \approx e^{\sum_{k=1}^n \lambda_k} \quad (3.34)$$

The odd Estrada index E_0 combines two experimental observations concerning the original Estrada index, namely that adjacency matrix eigenvalues alternate in sign around zero and that only a few of the positive largest such eigenvalues actually contribute to graph resilience.

Definition 3.12 (Odd Estrada index) *The odd Estrada index E_0 for assessing graph resilience is defined as:*

$$E_0 \triangleq \sum_{k=1}^m \sinh \lambda_n \quad (3.35)$$

The rationale behind the odd Estrada index is that only paths of odd length should be considered, since paths of even length are likely to consist of smaller repeated paths. This would be equivalent to counting twice the contribution of some edges. One way of ensuring that only odd paths are considered is to apply an odd function to the spectrum of the adjacency matrix. Hyperbolic sinus is a numerically stable and odd function, as it can be seen from its Taylor expansion of equation (3.36).

$$\sinh x = \sum_k \frac{x^k}{k!}, \quad k \equiv 1 \pmod{2} \quad (3.36)$$

Each layer consists of two distinct parts as seen in figure 3.2. First, two weight matrices W_1 and W_2 with the synaptic weights connecting the output of the previous layer with the two sigmoid sublayers. The second part is a third order tensor U which multilinearly combines the output of the two sigmoid sublayers and generates the predictions of that layer, which will be fed as input to the next layer. Therefore, TSNS are a natural choice for higher order data such as images, graphs, and geospatial data.

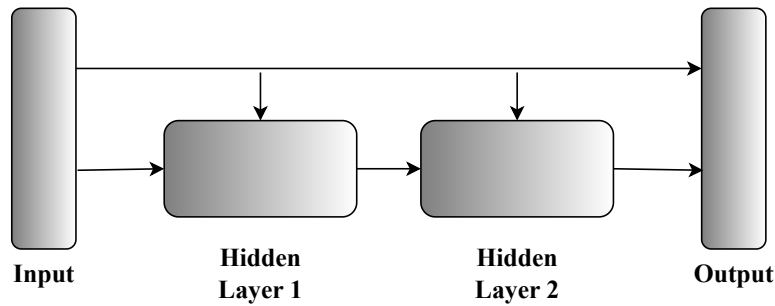


Figure 3.1: TSN architecture.

The prediction phase for a given layer ends when the predictions matrix Y are generated as seen in equation (3.37) by intermixing through tensor U the output of the sigmoid sublayers h_1 and h_2 . By definition of the Kronecker tensor product, the resulting matrix $h_1 \otimes h_2$ contains each possible product of the elements of vectors h_1 and h_2 . Each of the vectors h_1 and h_2 are the output of the two parallel sigmoid sublayers operating independently as in (3.37). Notice that these intermediate outputs are independent of each other. This allows TSNS to gradually discover and learn from two different feature sets. In equation (3.37) the non linear activation function is the sigmoid of equation (3.24) applied elementwise.

$$Y = U \times_1 (h_1 \otimes h_2) \quad \text{and} \quad \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} \varphi(W_1^T \text{vec}(X)) \\ \varphi(W_2^T \text{vec}(X)) \end{bmatrix} \quad (3.37)$$

During the training phase the local error gradients with respect to U , W_1 , and W_2 are determined. The selected solvers applied a regularization approach to avoid numerical errors or overfitting as in (3.38).

$$(A + \mu_0 I)x = b \quad (3.38)$$

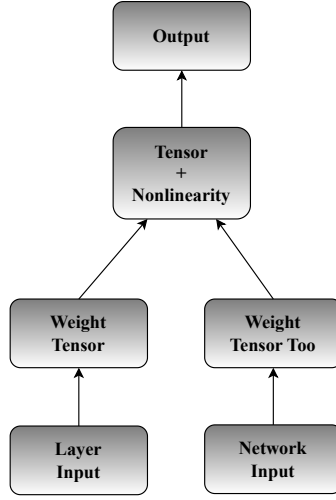


Figure 3.2: TSN layer structure.

The loss function was the binary negative cross entropy K of equation (3.39). Therein, c is the number of samples, p is the probability mass function of the training set and q the target probability mass function.

$$K \triangleq \frac{1}{c} \sum_u \sum_v q(v | u) \log p(v | u) \quad (3.39)$$

A major advantage of synthetic graphs is that they can be chosen to contain specific patterns which influence resilience. Table 3.2 summarizes the properties of the graphs used in the three datasets. Therein K_n is a complete graph, $B_{m,n}$ a bipartite graph, and C_n a cycle graph. The upper half of this table has the low resilience graphs, whereas the lower half contains the high resilience graphs. For each pattern exists two distinct yet isomorphic graphs which have it, so that TSNs can discover these patterns independently of row and column permutations of the adjacency matrix. All patterns exist in the three datasets. Graphs 1-20 and 61-80 have been assigned to training dataset, graphs 21-40 and 81-100 to training, and graphs 41-60 and 101-120 to validation. Kronecker graphs are generated in the following two steps.

- ♣ A generator graph with $|V|$ vertices and adjacency matrix M is selected.
- ♣ In each step a new graph is obtained with an adjacency matrix computed as in (3.40).

$$M^{[k]} \triangleq \begin{cases} M, & k = 0 \\ M \otimes M, & k = 1 \\ M^{[k-1]} \otimes M, & k \geq 2 \end{cases} \quad (3.40)$$

Table 3.2: Properties of the testing, training, and validation graphs.

Indices	Low resilience pattern
1-2, 21-22, 41-42	Binary tree
3-4, 23-24, 43-44	Three K_3 with 2 connecting edges
5-6, 25-26, 45-46	$B_{2,7}$ with 10 connecting edges
7-8, 27-28, 47-48	$B_{3,6}$ with 12 connecting edges
9-10, 29-30, 49-50	$B_{4,5}$ with 14 connecting edges
11-12, 31-32, 51-52	$B_{4,5}$ with 16 connecting edges
13-14, 33-34, 53-54	Two K_4 connected through a single articulation vertex
15-16, 35-36, 55-56	Non-overlapping C_4 and C_5 connected with a single edge
17-18, 37-38, 57-58	Two non-overlapping C_4 connected through a single articulation vertex
19-20, 39-40, 59-60	K_6 with 3 vertices of degree 1
Indices	High resilience pattern
61-62, 81-82, 101-102	Three overlapping C_4 with 4 edges between each cycle pair
63-64, 83-84, 103-104	Three overlapping C_5 with 6 edges between each cycle pair
65-66, 85-86, 105-106	Two K_7 with 12 connecting edges between each cycle
67-68, 87-88, 107-108	K_5 and K_4 with 8 connecting edges
69-70, 89-90, 109-110	K_5 and K_4 with 12 connecting edges
71-72, 91-92, 111-112	K_9 minus C_3
73-74, 93-94, 113-114	K_9 minus two K_3
75-76, 95-96, 115-116	Three K_3 with 8 connecting vertices
77-78, 97-98, 117-118	K_9 minus a C_4 and a C_5
78-80, 99-100, 119-120	K_9 minus a C_9

Table 3.3: Components of the two TSN architectures.

Component	Role	Arch
Data	Input graph represented as an $n \times n$ adjacency matrix	Both
$W_1^{(1)}$	Synaptic weights connecting the data layer with $H_1^{(1)}$	Both
$W_2^{(1)}$	Synaptic weights connecting the data layer with $H_2^{(1)}$	Both
$H_1^{(1)}$	First neuron stack of the first hidden layer	Both
$H_2^{(1)}$	Second neuron stack of the first hidden layer	Both
$U^{(1)}$	Tensor combining the outputs of the first hidden layer neuron stacks	Both
$W_1^{(2)}$	Synaptic weights connecting the first hidden layer with $H_1^{(2)}$	First
$W_2^{(2)}$	Synaptic weights connecting the first hidden layer with $H_2^{(2)}$	First
$H_1^{(2)}$	Upper neurons of the second hidden layer	First
$H_2^{(2)}$	Upper neurons of the second hidden layer	First
$U^{(2)}$	Tensor combining the outputs of the first hidden layer neuron stacks	First

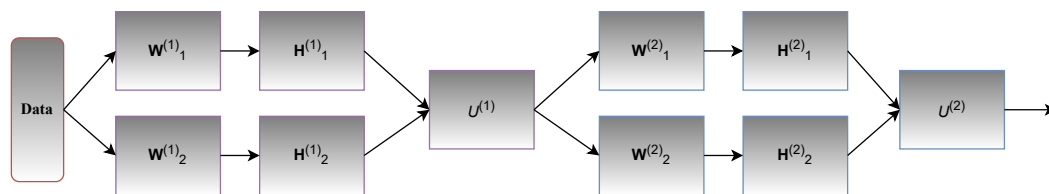


Figure 3.3: The first TSN architecture.

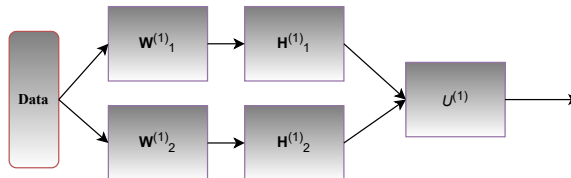


Figure 3.4: The second TSN architecture.

The first difference of the architectures is the additional hidden layer in the first architecture. Otherwise, the synaptic weights of both architectures are trained exactly in the same way, notwithstanding the different parameters and local error gradient matrices, especially those used in the additional hidden layer of the first architecture. Notice that the second architecture is rather a special case, as it has only one hidden layer. Weight initialization is the second difference as two strategies were used. The first was a Gaussian randomized strategy and the second the reciprocal of the outbound connections of the sublayer.

3.4.2 Results

Table 3.4 summarizes the contingency tables for each of the four cases, while table 3.5 summarizes the performance for them as assessed by accuracy, precision, recall, type I error, type II error, and F1 score as well as by the correlation with the Estrada index, the odd Estrada index, and the clustering coefficient. From these tables it can be inferred that two layers are preferable to only one. This can be attributed to the greater flexibility offered by two hidden layers, which can represent more nonlinear functions with increased degrees of freedom. This allows for correctly separating more matrix sets. Another observation is that initializing the weights according to a Gaussian seems to yield better results.

Table 3.4: Contingency tables for the four pairs architecture/strategy after training.

FA/FS	True	False	FA/SS	True	False	SA/FS	True	False	SA/SS	True	False
pos	20	0	pos	20	0	pos	18	2	pos	17	3
neg	19	1	neg	17	3	neg	18	2	neg	16	4

After training the validation dataset was fed to each of the four TSNs. In this way the entries of tables 3.6 and 3.7 are obtained. From the latter it can be seen that the first architecture/first option configuration and the second architecture/first options achieved the same results, whereas the other two configurations

Table 3.5: Metrics for the four TSN pairs architecture/strategy after training.

Conf	Accuracy	Type I	Type II	Prec	Recall	F1	Estrada	Odd	Cluster
fa/fs	0.9750	0.0000	0.0476	1.0000	0.9524	0.9756	0.8517	0.8445	0.8925
fa/ss	0.9250	0.0000	0.1304	1.0000	0.8696	0.9302	0.8792	0.8225	0.8633
sa/fs	0.9000	0.1000	0.1000	0.9000	0.9000	0.9000	0.8133	0.8019	0.8411
sa/ss	0.8250	0.1579	0.1905	0.8500	0.8095	0.8293	0.8001	0.7893	0.8116

had minor differences in performance, either positive or negative. This is an indication that TSNS can successfully generalize in unknown datasets, at least in datasets containing familiar connectivity patterns.

Table 3.6: Contingency tables for the four pairs architecture/strategy after validation.

FA/FS	True	False	FA/SS	True	False	SA/FS	True	False	SA/SS	True	False
pos	20	0	pos	19	1	pos	18	2	pos	17	3
neg	19	1	neg	17	3	neg	18	2	neg	17	3

3.5 Compressed graph recovery with TSNS

3.5.1 TSN architecture

In this scenario graphs whose structure has been compressed with the DCT2 are expanded with a specially designed TSN which utilizes structural information from similar graphs where similarity is determined by the topological correlation factor. This is done to exploit the increased variability and the non-linear higher order patterns. The TSN architecture is displayed in figure 3.5 and it consists of a number of components. The first components in every layer processing are the two tensor transforms of (3.41) on the layer input matrices U_I and U_L . Matrices are second order tensors, so multiplication is allowed provided they have compatible dimensions. This allows the separate processing of the input matrices.

$$U_I \triangleq H \times_1 X \quad \text{and} \quad U_L \triangleq G \times_1 S \quad (3.41)$$

Expanding (3.41) between the matrix X and the third order weight tensor H yields the intermediate matrix U_I of (3.42). Therein the matrix U_I is elementwise computed. The matrix U_L is similarly computed.

$$U_I [i_2, i_3] = (H \times_1 X) [i_2, i_3] \triangleq \sum_{i_1=1}^{I_1} H [i_1, i_2, i_3] X [i_1, i_2] \quad (3.42)$$

Table 3.7: Metrics for the four TSN pairs architecture/strategy after validation

Conf	Accuracy	Type I	Type II	Prec	Recall	F1	Estrada	Odd	Cluster
fa/fs	0.9750	0.0000	0.0476	1.0000	0.9524	0.9756	0.8311	0.8150	0.8633
fa/ss	0.9000	0.0555	0.1363	0.9500	0.8636	0.9047	0.8045	0.7944	0.8516
sa/fs	0.9000	0.1000	0.1000	0.9000	0.9000	0.9000	0.7875	0.7742	0.8241
sa/ss	0.8500	0.1500	0.1505	0.8500	0.8500	0.8500	0.7714	0.7498	0.8116

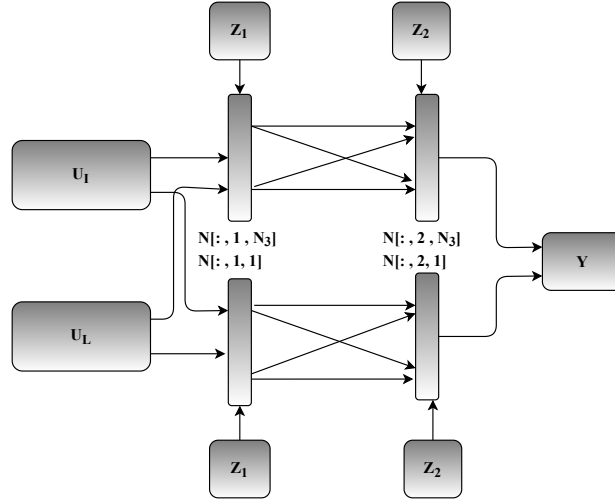


Figure 3.5: Hidden layer structure.

Once matrices U_I and U_L are computed, they are driven to N , a tensor containing an FFNN cluster and hence the layer computational core. The meaning of each dimension of $N \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is as follows:

- The first dimension is the neuron location. Consequently, $N[:, n_2, n_3]$ contains the weights of layer n_2 of network n_3 , while $N[:, :, n_3]$ contains the weights of n_3 .
- The second dimension denotes the layer location within the neuron tensor. Each layer of every network contains the same number of neurons, namely N_1 . Generally, this number may differ.
- Finally, the third dimension indicates an individual network in the cluster. By fixing this index, $N[:, :, n_3]$ is the synaptic networks internal to the n_3 -th FFNN.

Each of the N_3 networks has N_2 layers with N_1 neurons at each layer. The interaction between clustered layers is expressed by Z_1 and Z_2 . Geometrically, the packed FFNNs are placed in the area defined by the first two dimensions of N . Interactions between different layers are modeled by Z_k . Once U_L and U_I are obtained, the first layer is defined elementwise as shown in (3.43) below.

$$u \triangleq N[n_1, 1, n_3] \prod_{n'_2=1}^{N_2} U_L[n_1, n'_2] + (1 - N[n_1, 1, n_3]) \prod_{n'_2=1}^{N_2} U_I[n_1, n'_2] + \sum_{n'_1} \sum_{n'_3 \neq n_3} Z_1[n'_1, 1, n'_3] V[n'_1, 1, n'_3] \quad (3.43)$$

The result of (3.43) is routed to network n_3 and in neuron n_1 therein. It is linear in its arguments and it comprises of three distinct parts. The first is a weighted version of input matrix U_L , the second is also a weighted version of U_I , and the last term is a linear combination of the peer layer output. These factors compete with each other and eventually they form a network cluster. Once u is obtained, it is routed to

the appropriate neuron. Then input u' to neuron n_1 is similar to u as shown in (3.44).

$$u' \triangleq N[n_1, 2, n_3] \prod_{n'_2=1}^{N_2} V[n_1, 1, n'_2] + (1 - N[n_1, 2, n_3]) \prod_{n'_2=1}^{N_2} U_I[n_1, n'_2] + \sum_{n'_1} \sum_{n'_3 \neq n_3} Z_2[n'_1, 2, n'_3] V[n'_1, 2, n'_3] \quad (3.44)$$

3.5.2 Training

The error metric L assessing the progress of the TSN training process based on how different the TSN output of a given epoch and the corresponding desired response is critical. As TSN performance is evaluated based on the topology correlation coefficient, it is only logical that the TSN should yield a scalar output y . This paves the way for choosing L as the deterministic mean square error shown in (3.45). It is taken over all the C_0 pairs of the actual TSN output and the corresponding desired output $r_{d,k}$.

$$L \triangleq \frac{1}{C_0} \sum_{k=1}^{C_0} (r_{d,k} - y_k)^2 \quad (3.45)$$

Regarding the synaptic weights of the TSN, they are initialized with values drawn from a uniform distribution in I_1 . During each iteration the elements of weight tensors H and G and of the grid N are updated according to the delta rule. Moreover, the delta rule applies to N . In (3.46) $\varphi(\cdot)$ is elementwise applied.

$$\begin{aligned} H^{[k+1]} &= H^{[k]} + \mu_0^{[k+1]} X^{[k+1]} \left(\nabla_X N^{[k+1]} \right)^T \nabla_X N^{[k+1]} \\ G^{[k+1]} &= G^{[k]} + \mu_0^{[k+1]} S^{[k+1]} \left(\nabla_S N^{[k+1]} \right)^T \nabla_S N^{[k+1]} \\ N^{[k+1]} &= N^{[k]} + \mu_0^{[k+1]} (\nabla_X N + \nabla_S N) \times_1 E^{[k+1]} \end{aligned} \quad (3.46)$$

The error matrix $E^{[k+1]}$ of equation (3.46) is given by equation (3.47).

$$E^{[k]} \triangleq \begin{cases} I - 2Y^{[k]T} \varphi(Y^{[k]}), & \text{hidden layer} \\ \frac{2}{C_0} (y_k - r_{d,k}), & \text{output layer} \end{cases} \quad (3.47)$$

The local error gradient with respect to X of N and S for a hidden layer is the respective difference of the current value from the past one. The weight updates for the tensors Z_1 and Z_2 are given in (3.48):

$$\begin{aligned} Z_1^{[k+1]}[n_1, n_2, n_3] &\triangleq Z_1^{[k]}[n_1, n_2, n_3] + \mu_0^{[k+1]} u E[n_1, n_2] \\ Z_2^{[k+1]}[n_1, n_2, n_3] &\triangleq Z_2^{[k]}[n_1, n_2, n_3] + \mu_0^{[k+1]} u' E[n_1, n_2] \end{aligned} \quad (3.48)$$

3.5.3 Graph topology correlation

The topology correlation coefficient is one intuitive way to evaluate graph similarity, although certainly not the only one. Comparing the topology of two unweighted graphs is far from trivial. For two graphs without loops with n vertices one way to compute the topological correlation coefficient r_d is (3.49).

$$r_d \triangleq \frac{1}{n^2 - n} \sum_{i_1=0}^{n-1} \sum_{i_2=0}^{n-1} M_1[i_1, i_2] M_2[i_1, i_2] = \frac{1}{n^2 - n} \sum_{i=0}^{n^2-1} \text{vec}(M_1)[i] \text{vec}(M_2)[i] \quad (3.49)$$

For equation (3.49) to work the two graphs should be aligned. If the graphs contain loops, as in state modelling of transmission attempts over shared media in communication networks, only the normalization coefficient needs to be changed. The coefficient r_d counts only common edge occurrences. If common edge absences should also be counted, for instance in link prediction, then r_a of (3.50) can be used.

$$r_a \triangleq \frac{1}{n^2 - n} \sum_{i_1=0}^{n-1} \sum_{i_2=0}^{n-1} M_1 [i_1, i_2] \bar{\oplus} M_2 [i_1, i_2] = \frac{1}{n^2 - n} \sum_{i=0}^{n^2-1} \text{vec}(M_1) [i] \bar{\oplus} \text{vec}(M_2) [i] \quad (3.50)$$

In (3.50) the operator $\bar{\oplus}$ applied elementwise on the the adjacency matrices is the complementary XOR Boolean function. The proposed methodology can be applied to r_a without any loss of generality.

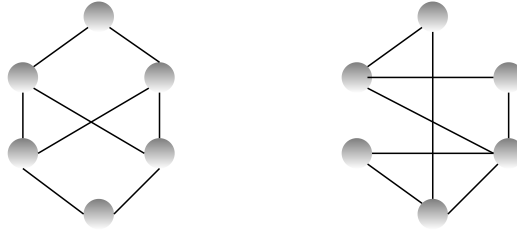


Figure 3.6: Two example input graphs.

The difference between (3.49) and (3.50) will be highlighted. In figure 3.6 two input graphs are shown whose adjacency matrices are in (3.51). The coefficients are $r_a = 1/3$ and $r_d = 2/3$. This difference can be explained by the fact that common edge absences are counted besides common edge occurrences in r_d .

$$M_1 \triangleq \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad M_2 \triangleq \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.51)$$

3.5.4 Spatial graph compression

The compressed adjacency matrix is obtained by the DCT2. By selecting only a fraction of the largest DCT2 coefficients is equivalent to preserving the major higher order structural graph properties. This is tantamount to compressing the input graph in terms on inherent spatial characteristics instead of fixed coding scheme. For a graph signal $x \in \mathbb{R}^{I_1 \times I_2}$ the DCT2 is a linear transform to the spatial frequency domain. The most common DCT2 variant is type II defined as shown in equation (3.52).

$$X [u_1, u_2] \triangleq \beta_1 [u_1] \beta_2 [u_2] \sum_{i_1=0}^{I_1-1} \sum_{i_2=0}^{I_2-1} x [i_1, i_2] \cos \omega_1 \cos \omega_2 \quad (3.52)$$

In (3.52) the spatial frequency variables ω_1 and ω_2 , also named transformed domain variables, are shown in equation (3.53). In GSP the meaning of spatial frequency is very closely connected to graph sampling. Higher spatial frequencies correspond to denser sampling as that neighborhoods of longer depths are examined. Thus DCT2 coefficients reflect structural patterns like vertex degrees and local path density. Parseval's theorem dictates that comparisons and operations in the original and DCT2 domains are equivalent. Only a fraction of the DCT2 coefficients are needed to reconstruct the major part graph.

$$\omega_1 \triangleq \frac{\pi u_1(2i_1 + 1)}{2I_1} \quad \text{and} \quad \omega_2 \triangleq \frac{\pi u_2(2i_2 + 1)}{2I_2} \quad (3.53)$$

The normalization coefficients $\beta_1 [u_1]$ and $\beta_2 [u_2]$ of equation (3.52), which ensure DCT2 is an isometric transform, take the values shown in equation (3.54):

$$\beta_1 [u_1] \triangleq \begin{cases} \sqrt{\frac{2}{I_1}}, & u_1 = 0 \\ \sqrt{\frac{1}{I_1}}, & u_1 \neq 0 \end{cases} \quad \text{and} \quad \beta_2 [u_2] \triangleq \begin{cases} \sqrt{\frac{2}{I_2}}, & u_2 = 0 \\ \sqrt{\frac{1}{I_2}}, & u_2 \neq 0 \end{cases} \quad (3.54)$$

3.5.5 Results

The proposed methodology will be assessed in terms of accuracy and complexity through three undirected graphs from the *network repository*¹⁰, namely the *inf-power*, the *yeast-protein-inter*, and the *ca-Erdos992* graphs. These were the starting points for forming the graph sequences $\langle P_k \rangle$, $\langle Y_k \rangle$, and $\langle E_k \rangle$ respectively, each having g_0 graphs. The following stochastic iterative process will be used.

- ♣ With probability p_0 an edge is added between one vertex from the ones having low degree and one vertex from those having high degrees. Both vertices are selected uniformly among their sets.
- ♣ Alternatively with probability $1 - p_0$ any two vertices of the graph are chosen with probability proportional to their locally weighted degrees as shown in equation (3.55).

$$\text{prob} \{v\} \propto \frac{\deg(v)}{\sum_u \deg(u)} \quad (3.55)$$

In (3.55) the denominator ranges over the neighborhood of v . This selection mechanism prevents the formulation of an exponential degree distribution and instead leads to a power law one. The latter implies the existence of rather large vertex subgroups with moderate degrees acting as local hubs.

In table 3.8 are shown the TSN configurations in terms of the fraction of the DCT2 coefficients P_0 used as training set and of the activation function. From left to right the activation function used is sReLU of (3.26), the logistic of (3.24), and SPOCU of (3.28). Moreover, table 3.9 has the normalized results for each TSN configuration and each graph sequence. The TSN performance metrics again from left to right are the final MSE, the number of epochs necessary to converge, the total execution time, and the total execution time minus training. The latter is important in determining whether TSNS are a viable alternative to direct DCT2 inversion. As TSNS are very CPU intensive, the wallclock time was used.

¹⁰<https://networkrepository.com>

Table 3.8: TSN configurations.

#	configuration	#	configuration	#	configuration
1	(0.1 P_0 , <i>sReLU</i>)	6	(0.1 P_0 , <i>lgi</i>)	11	(0.1 P_0 , <i>SPOCU</i>)
2	(0.15 P_0 , <i>sReLU</i>)	7	(0.15 P_0 , <i>lgi</i>)	12	(0.15 P_0 , <i>SPOCU</i>)
3	(0.2 P_0 , <i>sReLU</i>)	8	(0.2 P_0 , <i>lgi</i>)	13	(0.2 P_0 , <i>SPOCU</i>)
4	(0.25 P_0 , <i>sReLU</i>)	9	(0.25 P_0 , <i>lgi</i>)	14	(0.25 P_0 , <i>SPOCU</i>)
5	(0.3 P_0 , <i>sReLU</i>)	10	(0.3 P_0 , <i>lgi</i>)	14	(0.3 P_0 , <i>SPOCU</i>)

From the information contained in table 3.9 a number of conclusions can be drawn. The TSN configurations with the logistic activation function systematically outperform every other. Also with more DCT2 coefficients comes better performance. Still, the total number of training graph pairs remains a small fraction of the total available ones. Regarding these three configurations, there seems to be a tradeoff between MSE and the number of iterations. Fewer DCT2 coefficients result in slower convergence and hence more iterations. Additionally, the logistic function is more suited to the task, which can be explained by its stronger nonlinearity and its monotonicity. Finally, TSNs seem to learn sparse graphs more efficiently.

Table 3.9: Normalized results for each TSN configuration and benchmark sequence.

$\langle P_k \rangle$					$\langle Y_k \rangle$				$\langle Y_k \rangle$			
#	E_t	N_e	T_e	T_c	E_t	N_e	T_e	T_c	E_t	N_e	T_e	T_c
1	2.33	1.92	1.83	2.03	2.13	2.11	1.92	1.86	1.91	1.88	2.15	2.03
2	2.13	1.62	1.77	1.94	2.01	1.95	1.75	1.72	1.80	1.72	2.03	1.84
3	1.88	1.41	1.63	1.78	1.89	1.71	1.73	1.63	1.76	1.74	1.81	1.75
4	1.69	1.23	1.58	1.61	1.78	1.63	1.59	1.43	1.51	1.62	1.67	1.61
5	1.62	1.17	1.61	1.59	1.61	1.67	1.37	1.62	1.43	1.49	1.51	1.43
6	1.13	1.09	1.19	1.13	1.11	1.08	1.09	1.141	1.05	1.09	1.12	1.18
7	1.09	1.04	1.07	1.08	1.08	1	1.13	1.14	1.17	1.14	1.09	1.11
8	1	1.02	1	1	1	1.03	1	1.04	1	1.03	1	1
9	1.04	1	1.12	1.14	1.13	1.12	1.17	1	1.14	1.17	1.18	1.21
10	1.07	1.09	1.19	1.22	1.23	1.19	1.24	1.12	1.17	1.21	1.25	1.26
11	1.38	1.36	1.32	1.28	1.39	1.27	1.43	1.29	1.28	1.42	1.45	1.47
12	1.46	1.42	1.38	1.43	1.47	1.34	1.51	1.38	1.39	1.47	1.54	1.56
13	1.35	1.28	1.22	1.25	1.27	1.21	1.29	1.21	1.22	1.24	1.25	1.27
14	1.44	1.42	1.41	1.46	1.44	1.45	1.42	1.46	1.45	1.42	1.45	1.47
15	1.53	1.47	1.54	1.52	1.56	1.53	1.52	1.51	1.54	1.55	1.53	1.56

3.6 Bibliography

- [Aba16a] Martín Abadi. TensorFlow: Learning functions at scale. *ACM SIGPLAN Notices*, 51(9):1–1, 2016.
- [Aba16b] Martín et al Abadi. TensorFlow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

- [AS15] Mohammed JF Alenazi and James PG Sterbenz. Comprehensive comparison and accuracy of graph metrics in predicting network resilience. In *DRCN*, pages 157–164. IEEE, 2015.
- [ATAA24] Muthuvel Arumugam, Arunprasath Thiyagarajan, Lakshmi Adhi, and Shyamala Alagar. Crossover smell agent optimized multilayer perceptron for precise brain tumor classification on MRI images. *Expert Systems with Applications*, 238, 2024.
- [BAS23] Adil M Bagirov, Ramiz M Aliguliyev, and Nargiz Sultanova. Finding compact and well-separated clusters: Clustering using silhouette coefficients. *Pattern Recognition*, 135, 2023.
- [BBM20] Stuart J Blair, Yaxin Bi, and Maurice D Mulvenna. Aggregated topic models for increasing social media topic coherence. *Applied Intelligence*, 50(1):138–156, 2020.
- [BC24] Eimear Byrne and Giuseppe Cotardo. Constructions of perfect bases for classes of 3-tensors. *Linear Algebra and its Applications*, 683:1–30, 2024.
- [Bea11] James Bergstra and et al. Theano: Deep learning on GPUs with Python. In *NIPS BigLearning workshop*, volume 3, pages 1–48, 2011.
- [BFZM18] Yalong Bai, Jianlong Fu, Tiejun Zhao, and Tao Mei. Deep attention neural tensor network for visual question answering. In *ECCV*, pages 20–35, 2018.
- [BG06] Mehrzad Biguesh and Alex B Gershman. Training-based MIMO channel estimation: A study of estimator tradeoffs and optimal training signals. *IEEE transactions on signal processing*, 54(3):884–893, 2006.
- [BGL15] Austin R Benson, David F Gleich, and Jure Leskovec. Tensor spectral clustering for partitioning higher-order network structures. In *ICDM*, pages 118–126. SIAM, 2015.
- [BGM23] Thomas Bridgeman, Helena Gibson, and Kevin Meehan. Gesture recognition and showing a cyclist’s intent. In *AIoT*. IEEE, 2023.
- [Bis95] Chris M Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.
- [BJT24] Abdeslem Hafid Bentbib, Khalide Jbilou, and Ridwane Tahiri. n -mode minimal tensor extrapolation methods. *Numerical Algorithms*, 95(2):665–691, 2024.
- [BPT15] Johann A. Bengua, Ho N. Phien, and Hoang D. Tuan. Optimal feature extraction and classification of tensors via matrix product state decomposition. In *ICBD*, pages 669–672. IEEE, 2015.
- [BPWV20] Reetam Sen Biswas, Anamitra Pal, Trevor Werho, and Vijay Vittal. A graph theoretic approach to power system vulnerability identification. *IEEE Transactions on Power Systems*, 36(2):923–935, 2020.
- [BS13] Lisa Feldman Barrett and Ajay Bhaskar Satpute. Large-scale brain networks in affective and social neuroscience: Towards an integrative functional architecture of the brain. *Current opinion in neurobiology*, 23(3):361–372, 2013.

- [CCHN24] HanQin Cai, Zehan Chao, Longxiu Huang, and Deanna Needell. Robust tensor cur decompositions: Rapid low-tucker-rank tensor recovery with sparse corruptions. *SIAM Journal on Imaging Sciences*, 17(1):225–247, 2024.
- [CFKS20] Andrew Cullen, Pietro Ferraro, Christopher King, and Robert Shorten. On the resilience of DAG-based distributed ledgers in IoT applications. *IEEE Internet of Things Journal*, 7(8):7112–7122, 2020.
- [Chu23] Benjamin Chung. *A type system for Julia*. PhD thesis, Northeastern University, 2023.
- [CKF11] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. torch7: A MATLAB-like environment for machine learning. In *BigLearn, NIPS workshop*, 2011.
- [CL23] Chiara Contoli and Emanuele Lattanzi. A study on the application of TensorFlow compression techniques to human activity recognition. *IEEE Access*, 2023.
- [CQW23] Shawn X Cui, Yang Qiu, and Zhenghan Wang. From three dimensional manifolds to modular tensor categories. *Communications in Mathematical Physics*, 397(3):1191–1235, 2023.
- [CWYH24] Jiahao Chen, Yaxiong Wu, Chaojing Yao, and Xiao Huang. Robust motion learning for musculoskeletal robots based on a recurrent neural network and muscle synergies. *IEEE Transactions on Automation Science and Engineering*, 2024.
- [Den13] Li et al Deng. Recent advances in deep learning for speech research at Microsoft. In *ICASSP*. IEEE, 2013.
- [Den14] Li Deng. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on signal and information processing*, 3, 2014.
- [DHG13] Li Deng, Xiaodong He, and Jianfeng Gao. Deep stacking networks for information retrieval. In *ICASSP*. IEEE, 2013.
- [DHY12] Li Deng, Brian Hutchinson, and Dong Yu. Parallel training for deep stacking networks. In *Thirteenth annual conference of the International Speech Communication Association*, 2012.
- [DJP⁺24] Robertas Damaševičius, Luka Jovanovic, Aleksandar Petrovic, Miodrag Zivkovic, Nebojsa Bacanin, Dejan Jovanovic, and Milos Antonijevic. Decomposition aided attention-based recurrent neural networks for multistep ahead time-series forecasting of renewable power generation. *PeerJ Computer Science*, 10, 2024.
- [DWA⁺24] Darrel R Deo, Francis R Willett, Donald T Avansino, Leigh R Hochberg, Jaimie M Henderson, and Krishna V Shenoy. Brain control of bimanual movement enabled by recurrent neural networks. *Scientific Reports*, 14(1), 2024.
- [DY11] Li Deng and Dong Yu. Deep convex net: A scalable architecture for speech pattern classification. In *Twelfth annual conference of the International Speech Communication Association*, 2011.
- [EH10] Ernesto Estrada and Desmond J Higham. Network properties revealed through matrix functions. *SIAM Review*, 52(4):696–714, 2010.

- [Est22] Ernesto Estrada. The many facets of the Estrada indices of graphs and networks. *SeMA Journal*, 79(1):57–125, 2022.
- [FD23] Katarzyna Filus and Joanna Domańska. Software vulnerabilities in TensorFlow-based deep learning applications. *Computers & Security*, 124, 2023.
- [FM24] Carson Farmer and Hector Medina. Hyperelastics.jl: A Julia package for hyperelastic material modelling with a large collection of models. *Journal of Open Source Software*, 9(96), 2024.
- [FYK⁺22] Scott Freitas, Diyi Yang, Srijan Kumar, Hanghang Tong, and Duen Horng Chau. Graph vulnerability and robustness: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):5915–5934, 2022.
- [FYL⁺24] Shikai Fang, Xin Yu, Shibo Li, Zheng Wang, Mike Kirby, and Shandian Zhe. Streaming factor trajectory learning for temporal tensor decomposition. *Advances in Neural Information Processing Systems*, 36, 2024.
- [GB13] Alexander Grubb and J Andrew Bagnell. Stacked training for overfitting avoidance in deep networks. In *ICML workshops*, page 1, 2013.
- [GBH⁺23] Martin Ganahl, Jackson Beall, Markus Hauru, Adam GM Lewis, Tomasz Wojno, Jae Hyeon Yoo, Yijian Zou, and Guifre Vidal. Density matrix renormalization group with tensor processing units. *PRX Quantum*, 4, 2023.
- [GHO99] Gene H Golub, Per Christian Hansen, and Dianne P O’Leary. Tikhonov regularization and total least squares. *Journal on matrix analysis and applications*, 21(1):185–194, 1999.
- [GP17] Antonio Gulli and Sujit Pal. *Deep learning with keras*. PACKT Publishing Ltd, 2017.
- [GPZZ24] Li-Hua Gong, Jun-Jie Pei, Tian-Feng Zhang, and Nan-Run Zhou. Quantum convolutional neural network based on variational quantum circuits. *Optics Communications*, 550, 2024.
- [GSK24] Richard Gast, Sara A Solla, and Ann Kennedy. Neural heterogeneity controls computations in spiking neural networks. *Proceedings of the National Academy of Sciences*, 121(3), 2024.
- [HDY12] Brian Hutchinson, Li Deng, and Dong Yu. Tensor deep stacking networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1944–1957, 2012.
- [HDY13] Brian Hutchinson, Li Deng, and Dong Yu. Tensor deep stacking networks. *TPAMI*, 35(8):1944–1957, 2013.
- [HEW24] Alexander Henkes, Jason K Eshraghian, and Henning Wessels. Spiking neural networks for nonlinear regression. *Royal Society Open Science*, 11(5), 2024.
- [HHY⁺17] Guosheng Hu, Yang Hua, Yang Yuan, Zhihong Zhang, Zheng Lu, Sankha S Mukherjee, Timothy M Hospedales, Neil M Robertson, and Yongxin Yang. Attribute-enhanced face recognition with neural tensor fusion networks. In *ICCV*, pages 3744–3753, 2017.
- [HHZ⁺23] Zhi-Long Han, Ting-Zhu Huang, Xi-Le Zhao, Hao Zhang, and Yun-Yang Liu. Multi-dimensional data recovery via feature-based fully-connected tensor network decomposition. *IEEE Transactions on Big Data*, 2023.

- [HSZ23] Rungang Han, Pixu Shi, and Anru R Zhang. Guaranteed functional tensor singular value decomposition. *Journal of the American Statistical Association*, pages 1–13, 2023.
- [HT24] Philipp Holl and Nils Thuerey. ϕ -ML: Intuitive scientific computing with dimension types for Jax, PyTorch, TensorFlow & NumPy. *Journal of Open Source Software*, 9(95), 2024.
- [IW11] Wai Hung Ip and Dingwei Wang. Resilience and friability of transportation networks: Evaluation, analysis and optimization. *IEEE Systems Journal*, 5(2):189–198, 2011.
- [Jia14] Yangqing et al Jia. Caffe: Convolutional architecture for fast feature embedding. In *International conference on multimedia*, pages 675–678. ACM, 2014.
- [JT24] Hojin Jang and Frank Tong. Improved modeling of human vision by incorporating robustness to blur in convolutional neural networks. *Nature Communications*, 15(1), 2024.
- [KAW⁺24] Ayaka Katayama, Yuki Aoki, Yukako Watanabe, Jun Horiguchi, Emad A Rakha, and Tetsunari Oyama. Current status and prospects of artificial intelligence in breast cancer pathology: Convolutional neural networks to prospective vision transformers. *International Journal of Clinical Oncology*, pages 1–21, 2024.
- [KGN⁺19] Aditya Khamparia, Deepak Gupta, Nhu Gia Nguyen, Ashish Khanna, Babita Pandey, and Prayag Tiwari. Sound classification using convolutional neural network and tensor deep stacking network. *IEEE Access*, 7:7717–7727, 2019.
- [Kol09] Tamara Kolda. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.
- [KS24] Raghavendra Kanakagiri and Edgar Solomonik. Minimum cost loop nests for contraction of a sparse tensor with a tensor network. In *Symposium on Parallelism in Algorithms and Architectures*, pages 169–181. ACM, 2024.
- [KSM15] Raman Kumar, Arjun Sahni, and Divyanshu Marwah. Real time big data analytics dependence on network monitoring solutions using tensor networks and its decompositions. *Network and Complex Systems*, 5(2), 2015.
- [Kus20] Andrew Kusiak. Open manufacturing: A design-for-resilience approach. *International Journal of Production Research*, 58(15):4647–4658, 2020.
- [L⁺14] Erik G Larsson et al. Massive MIMO for next generation wireless systems. *IEEE communications magazine*, 52(2):186–195, 2014.
- [L⁺16] Shan Lin et al. ATPC: Adaptive transmission power control for wireless sensor networks. *TOSN*, 12(1):6, 2016.
- [LB15] Longzhuang Li and Douglas Boulware. High-order tensor decomposition for large-scale data analysis. In *ICBD*, pages 665–668. IEEE, 2015.
- [LCCJ20] Xin Liu, Bo Chen, Chen Chen, and Dong Jin. Electric power grid resilience with interdependencies between power and communication networks –a review. *IET Smart Grid*, 3(2):182–193, 2020.

- [LCL⁺24] Yipeng Liu, Jie Chen, Yingcong Lu, Weiting Ou, Zhen Long, and Ce Zhu. Adaptively topological tensor network for multi-view subspace clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- [LCW05] Dmitri Loguinov, Juan Casas, and Xiaoming Wang. Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience. *IEEE/ACM TON*, 13(5):1107–1120, 2005.
- [LCY15] Jun Li, Heyou Chang, and Jian Yang. Sparse deep stacking network for image classification. In *AAAI*, pages 3804–3810, 2015.
- [LL24] Jian Lian and Tianyu Liu. Lesion identification in fundus images via convolutional neural network-vision transformer. *Biomedical Signal Processing and Control*, 88, 2024.
- [LLWA24] Cheng Lyu, Qing-Long Lu, Xinhua Wu, and Constantinos Antoniou. Tucker factorization-based tensor completion for robust traffic data imputation. *Transportation research part C: Emerging technologies*, 160, 2024.
- [Loy01] Sergey L Loyka. Channel capacity of MIMO architecture using the exponential correlation matrix. *IEEE communications letters*, 5(9):369–371, 2001.
- [LWS⁺23] Xiao Peng Li, Zhi-Yong Wang, Zhang-Lei Shi, Hing Cheung So, and Nicholas D Sidiropoulos. Robust tensor completion via capped Frobenius norm. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [Mat17] De G et al Matthews. GPflow: A Gaussian process library using TensorFlow. *JMLR*, 18(1):1299–1304, 2017.
- [MBB17] Federico Monti, Michael Bronstein, and Xavier Bresson. Geometric matrix completion with recurrent multi-graph neural networks. In *NIPS*, pages 3697–3707, 2017.
- [Men16] Xiangrui et al Meng. MLlib: Machine learning in Apache Spark. *JMLR*, 17(1):1235–1241, 2016.
- [MG20] Giannis Moutsinas and Weisi Guo. Node-level resilience loss in dynamic complex networks. *Scientific reports*, 10(1), 2020.
- [MO23] Alexis Montoison and Dominique Orban. Krylov.jl: A Julia basket of hand-picked Krylov methods. *Journal of Open Source Software*, 8(89), 2023.
- [MQEC20] Abdullah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *CVPR*, pages 14424–14432, 2020.
- [NG90] Walid Najjar and J-L Gaudiot. Network resilience: A measure of network fault tolerance. *ToC*, 2(1):174–181, 1990.
- [NLM13] Hien Quoc Ngo, Erik G Larsson, and Thomas L Marzetta. Energy and spectral efficiency of very large multiuser MIMO systems. *ToC*, 61(4):1436–1449, 2013.

- [NPN⁺24] Ha-Thanh Nguyen, Manh-Kien Phi, Xuan-Bach Ngo, Vu Tran, Le-Minh Nguyen, and Minh-Phuong Tu. Attentive deep neural networks for legal document retrieval. *Artificial Intelligence and Law*, 32(1):57–86, 2024.
- [OHCB24] Cagri Ozdemir, Randy C Hoover, Kyle Caudle, and Karen Braman. Tensor discriminant analysis on Grassmann manifold with application to video based human action recognition. *International Journal of Machine Learning and Cybernetics*, pages 1–13, 2024.
- [PAEHC⁺24] Mangpo Phothilimthana, Sami Abu-El-Haija, Kaidi Cao, Bahare Fatemi, Michael Burrows, Charith Mendis, and Bryan Perozzi. TPUgraphs: A performance prediction dataset on large tensor computational graphs. *Advances in Neural Information Processing Systems*, 36, 2024.
- [Pat23] Santanu Pattanayak. *Pro deep learning with TensorFlow 2.0: A mathematical approach to advanced artificial intelligence in Python*. Springer, 2023.
- [Pat24] Radha Patel. *A System to Exploit Symmetry in Common Tensor Kernels*. PhD thesis, MIT, 2024.
- [PF15] Evangelos E. Papalexakis and Christos Faloutsos. Fast efficient and scalable core consistency diagnostic for the PARAFAC decomposition for big sparse tensors. In *ICASSP*, pages 5441–5445, 2015.
- [PH15] David Palzer and Brian Hutchinson. The tensor deep stacking network toolkit. In *IJCNN*, pages 1–5. IEEE, 2015.
- [PLH⁺24] Chenjian Pan, Chen Ling, Hongjin He, Liquan Qi, and Yanwei Xu. A low-rank and sparse enhanced Tucker decomposition approach for tensor completion. *Applied Mathematics and Computation*, 465, 2024.
- [PPF14] Evangelos E. Papalexakis, Konstantinos Pelechrinis, and Christos Faloutsos. Spotting misbehaviors in location-based social networks using tensors. In *WWW*, pages 551–552, 2014.
- [PPLA23] Karelia Pena-Pena, Daniel L Lau, and Gonzalo R Arce. T-HGSP: Hypergraph signal processing using t-product tensor decompositions. *IEEE Transactions on Signal and Information Processing over Networks*, 2023.
- [Pri91] Douglas M Priest. Algorithms for arbitrary precision floating point arithmetic. In *Tenth symposium on computer arithmetic*, pages 132–143. IEEE, 1991.
- [PWD13] Hamid Palangi, Rabab K Ward, and Li Deng. Using deep stacking network to improve structured compressed sensing with multiple measurement vectors. In *ICASSP*, pages 3337–3341, 2013.
- [PYHH21] Shouzheng Pan, Hai Yan, Jia He, and Zhengbing He. Vulnerability and resilience of transportation systems: A recent literature review. *Physica A: Statistical Mechanics and its Applications*, 581, 2021.
- [QYZ24] Zhen Qin, Songlin Yang, and Yiran Zhong. Hierarchically gated recurrent neural network for sequence modeling. *Advances in Neural Information Processing Systems*, 36, 2024.
- [RMC⁺24] Dayong Ren, Zhe Ma, Yuanpei Chen, Weihang Peng, Xiaode Liu, Yuhang Zhang, and Yufei Guo. Spiking pointnet: Spiking neural networks for point clouds. *Advances in Neural Information Processing Systems*, 36, 2024.

- [RSJ23] K Rimal, KB Shah, and AK Jha. Advanced multi-class deep learning convolution neural network approach for insect pest classification using TensorFlow. *International Journal of Environmental Science and Technology*, 20(4):4003–4016, 2023.
- [SAZ⁺20] Yalda Saadat, Bilal M Ayyub, Yanjie Zhang, Dongming Zhang, and Hongwei Huang. Resilience-based strategies for topology enhancement and recovery of metrorail transit networks. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 6(2), 2020.
- [Sch15] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.
- [SNAC16] Yang Shi, UN Niranjana, Animashree Anandkumar, and Cris Cecka. Tensor contractions with extended BLAS kernels on CPU and GPU. In *HiPC*, pages 193–202. IEEE, 2016.
- [SS08] Thomas Schultz and Hans-Peter Seidel. Estimating crossing fibers: A tensor decomposition approach. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1635–1642, 2008.
- [SZT⁺23] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *CVPR*, pages 16632–16642. IEEE/CVF, 2023.
- [TSYZ19] Jinhui Tang, Xiangbo Shu, Rui Yan, and Liyan Zhang. Coherence constrained graph LSTM for group activity recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2019.
- [TYLJ24] Zhihui Tu, Kaitao Yang, Jian Lu, and Qingtang Jiang. Tensor recovery using the tensor nuclear norm based on nonconvex and nonlinear transformations. *Signal Processing*, 219, 2024.
- [TZC19] Wen Jun Tan, Allan N Zhang, and Wentong Cai. A graph-based model to measure structural redundancy for supply chain resilience. *International Journal of Production Research*, 57(20):6385–6404, 2019.
- [VDDL16] Nico Vervliet, Otto Debaels, and Lieven De Lathauwer. TensorLab 3.0 - numerical optimization strategies for large-scale constrained and coupled matrix-tensor factorization. In *Asilomar conference on signals, systems and computers*, pages 1733–1738. IEEE, 2016.
- [VL15] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for MATLAB. In *International conference on multimedia*, pages 689–692. ACM, 2015.
- [VM03] Alexei Vázquez and Yamir Moreno. Resilience to damage of graphs with degree correlations. *Physical Review E*, 67(1):15–101, 2003.
- [W⁺18] Mengjiao Wang et al. Disentangling the modes of variation in unlabelled data. *TPAMI*, 40(11):2682–2695, 2018.
- [WHH⁺23] Minghua Wang, Danfeng Hong, Zhu Han, Jiaxin Li, Jing Yao, Lianru Gao, Bing Zhang, and Jocelyn Chanussot. Tensor decompositions for hyperspectral data processing in remote sensing: A comprehensive review. *IEEE Geoscience and Remote Sensing Magazine*, 11:26–72, 2023.

- [WMK⁺21] Zelin Wan, Yash Mahajan, Beom Woo Kang, Terrence J Moore, and Jin-Hee Cho. A survey on centrality metrics and their network resilience analysis. *IEEE Access*, 9, 2021.
- [Won18] Kanit et al Wongsuphasawat. Visualizing dataflow graphs of deep learning models in TensorFlow. *Transactions on visualization and computer graphics*, 24(1):1–12, 2018.
- [WOV23] Xinjue Wang, Esa Ollila, and Sergiy A Vorobyov. Nonnegative structured Kruskal tensor regression. In *CAMSAP*, pages 441–445. IEEE, 2023.
- [WPPQA24] Fuli Wang, Karelia Pena-Pena, Wei Qian, and Gonzalo R Arce. T-HyperGNNs: Hypergraph neural networks via tensor representations. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
- [WPW⁺20] Hong-Wei Wang, Zhong-Ren Peng, Dongsheng Wang, Yuan Meng, Tianlong Wu, Weili Sun, and Qing-Chang Lu. Evaluation and prediction of transportation resilience under extreme weather events: A diffusion graph convolutional approach. *Transportation research part C: Emerging technologies*, 115, 2020.
- [WX20] Wen-Huan Wang and Yi-Sai Xue. On the r-uniform linear hypertrees with extremal Estrada indices. *Applied Mathematics and Computation*, 377, 2020.
- [WYL⁺23] Fengyi Wu, Hang Yu, Anran Liu, Junhai Luo, and Zhenming Peng. Infrared small target detection using spatio-temporal 4d tensor train and ring unfolding. *IEEE Transactions on Geoscience and Remote Sensing*, 2023.
- [XGS⁺24] Qi Xu, Yuyuan Gao, Jiangrong Shen, Yaxin Li, Xuming Ran, Huajin Tang, and Gang Pan. Enhancing adaptive history reserving by spiking convolutional block attention module in recurrent neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.
- [XLY⁺24] Mingxuan Xiao, Yufeng Li, Xu Yan, Min Gao, and Weimin Wang. Convolutional neural network classification of cancer cytopathology images: Taking breast cancer as an example. In *International Conference on Machine Vision and Applications*, pages 145–149, 2024.
- [XYC⁺24] Guoqing Xiao, Chuanghui Yin, Yuedan Chen, Mingxing Duan, and Kenli Li. Efficient utilization of multi-threading parallelism on heterogeneous systems for sparse tensor contraction. *IEEE Transactions on Parallel and Distributed Systems*, 2024.
- [YCZY21] Shenhao Yang, Weirong Chen, Xuexia Zhang, and Weiqi Yang. A graph-based method for vulnerability analysis of renewable energy integrated power systems to cascading failures. *Reliability Engineering & System Safety*, 207, 2021.
- [YDS13] Dong Yu, Li Deng, and Frank Seide. The deep tensor neural network with applications to large vocabulary speech recognition. *Transactions on audio, speech, and language processing*, 21(2):388–396, 2013.
- [YHY⁺24] Xiaofeng Yuan, Lingfeng Huang, Lingjian Ye, Yalin Wang, Kai Wang, Chunhua Yang, Weihua Gui, and Feifan Shen. Quality prediction modeling for industrial processes using multiscale attention-based convolutional neural network. *IEEE Transactions on Cybernetics*, 2024.
- [YSA24] Magombe Yasin, Mehmet Sarigül, and Mutlu Avci. Logarithmic learning differential convolutional neural network. *Neural Networks*, 172, 2024.

- [YZL⁺24] Xue Yang, Yue Zhou, Wenlong Liao, Tao He, and Junchi Yan. Alpharotate: A rotation detection benchmark using tensorflow. In *ICASSP*, pages 9481–9485. IEEE, 2024.
- [ZWSS15] Rui Zeng, Jiasong Wu, Lotfi Senhadji, and Huazhong Shu. Tensor object classification via multilinear discriminant analysis network. In *ICASSP*, pages 1971–1975. IEEE, 2015.
- [ZWZ⁺24] Xia Zhao, Limin Wang, Yufei Zhang, Xuming Han, Muhammet Deveci, and Milan Parmar. A review of convolutional neural networks in computer vision. *AI Review*, 57(4), 2024.

Chapter 4

Self organizing maps

*Good evening, ladies and gentlemen.
My name is Orson Welles.
I am an actor.
I am a writer.
I am a producer.
I am a director.
I am a magician.
I appear onstage and on the radio.
Why are there so many of me and so few of you?
Orson Welles (1915 – 1985)*

4.1 Chapter introduction

SELF organizing maps (SOMs) or cognitive maps are a class of unsupervised neural network architectures which rely heavily on the flexibility to learn how to create and modify connections between two distinct spaces, namely the *data space* $\mathcal{V}$ and the *coordinate space* $\mathcal{C}$ [Koh90]. The notation $\mathcal{V}$ and $\mathcal{C}$ for these two distinct yet related spaces will be preserved throughout this chapter. SOMs are a rarity among neural network architectures in the sense that they perform an unsupervised learning process in sharp

contrast to the existing mainstream architectures which act in a supervised manner like the MLP. This trend for unsupervised neural networks continues with the development of other architectures such as autoencoders [CDW⁺24; BDS⁺24], generative adversarial networks (GANs) [GLZ24; OTP⁺24; BR24], and GNNs. The latter are the topic of chapter 5. As a general note, unsupervised neural networks are becoming essential for extracting knowledge from large datasets since the computational complexity of training in supervised learning architectures gradually becomes prohibitive. Moreover, with data scaling so do the problems of dimensionality, data space sparsity, and data quality as well as the ever necessary need for data annotation. This is especially true in humanistic data as explained in chapter 1.

Typically the data points of $\mathcal{V}$ come from a manifold and SOMs construct a low dimensional projection thereof in C while preserving up to an extent its primary topological properties by iteratively learning the appropriate synaptic weights by combining the following elements [Kiv96; Koh97]:

- ♣ A *distance function* defined in $\mathcal{V}$ for comparing data points and neuron synaptic weights.
- ♣ A modified, essentially somewhat relaxed, Hebbian learning rule [WKPP24; KBM24] operating in C which is a spatial function of the location of a neurons from a special neuron termed *best matching neuron* (BMN), which is different during every iteration of the learning process.
- ♣ A *proximity function* defined in C which determines which neurons are close to the BMN.
- ♣ A *weight function* which is also defined in C and decides the amount of reward the neurons deemed close to the BMN based on the proximity function should receive.

This deceptively simple mechanism which does not require extensive and elaborate optimization techniques is how SOMs deal with the high dimensionality of $\mathcal{V}$. Although SOMs are definitely not new as a concept since their inception can be traced back to [Koh90], the introduction of tensor distance metrics for evaluating data point proximity in the attribute space is the novelty introduced in this dissertation.

Neuroimaging with the progress of medical technology is emerging as a non invasive diagnostic method [ALS24]. It is a collective term for a range of technologies, each of which relying on different physical, chemical, and biological properties of the human body in order to study the central human nervous system. This is done in this chapter by integrating a tensor distance metric into the SOM. This is versatile enough to allow the SOM to learn higher order correlations between fMRI pixels and also highlights the role of geometry in determining boundaries in images. In order to evaluate the proposed SOM architecture, it has been tested with an open dataset, under the Creative Commons license from openneuro.org, containing fMRI images taken during three cognitive tasks conducted by elderly people. Since ground truth data is available, the confusion matrices have been derived for SOMs of various configurations. Moreover, the corresponding topological error rates, the activation set change rates, and the intra-cluster distance variations have been computed for each epoch during the cognitive map formulation.

The objectives of this chapter are given in research statement 4.1 below.

Objective 4.1 (Chapter research objectives) *This chapter presents an SOM with a tensor distance metric in the data space to cluster fMRI images representing different cognitive tasks.*

The remainder of this chapter is structured as follows. In section 4.2 the recent scientific literature regarding SOMs, ML and DL neuroimaging applications, and graph clustering. For the bibliography review pertaining to tensors and tensor distance metrics please see section 3.2. The proposed SOM architecture along with the associated tensor distance metric are described in subsection 4.3.1, while the training process is given in 4.3.2. The results obtained are shown in subsection 4.3.3. Finally, conclusions are in chapter 6.

4.2 Previous work

SOMs constitute a class of neurons arranged in a low dimensional grid and trained in an unsupervised manner based on a modified Hebbian rule [KKL90]. SOMs are designed to learn the topology of an underlying higher dimensional manifold [Kiv96] and to progressively create a low dimensional projection thereof where the major topological properties of the underlying manifold are approximately preserved. As such multiview clustering can be achieved [JRS⁺24]. The Hebbian learning rule is extensively discussed in [WKPP24] and its connections to iterative learning are explored in [KBM24]. SOM applications include clustering large databases [Koh90; Koh97], interpreting gene expressions [TSM⁺99], federated learning [GFA⁺24], discovering polluted water areas [GS24], and denoising industrial images [WAM19]. Fuzzy cognitive maps [SSL24] can extend the functionality of original SOMs by relaxing the hard constraints on the clustering they perform [JRMK24], whereas risk assessment for complex systems [AYZA⁺24] and cognitive-affective methods for test-retest reliability [GRS⁺24] are two more applications thereof. SOMs have been proposed as the computational model of the hippocampus in human brain [BSZ⁺24]. Cognitive maps are covered extensively in [GN24].

Neuroimaging is a prime application field for ML and DL as well as a major interdisciplinary research area on its own right [ALS24]. DL has been proposed for enhancing MRI images [YC24]. DL for Alzheimer's disease is proposed in [DHF⁺24]. An overview of magnetic resonance in medical imaging is given in [HT24]. Cross modal neuroimaging based on fusing tensor modes is the focus of [LBM⁺24]. Explainable DL for neuroimaging is discussed among others in [DMH⁺24] and in [MSH⁺24], while for strokes in [MPADB24]. Knowledge transfer from neuroimaging to psychiatry is explored in [DGP⁺24]. Wavelet processing with LSTM for brain signals is proposed in [ABJ⁺24]. Other DL techniques for neuroimaging are given in [JKV24]. The dataset for this chapter was taken from [LMM⁺19a].

Clustering discovers structure in raw data without supervision [EKM⁺24; FLH24; IAT24]. As such, it is by definition an extremely broad field as it covers topics ranging from approximate reasoning [V⁺24] and transcriptomics [YZL⁺24] to visual learning [YQH24] and visual transformers [LCW⁺24] and, hence, it cannot be covered entirely in this bibliography review section. It relies heavily on the selection of data representation [DSYA24] as well as on the distance metric [HMB24]. An overview of the field is given in [OT23], in [LRU20], and in [GND⁺20]. Graph clustering takes a number of forms depending on the application and the algorithmic approach, which can be infomation theoretic criteria [CRP⁺24], structural considerations [YWD⁺24], or linear programming [CCAL⁺24]. Graph clustering in multiple resolution levels is the topic of [TGZ⁺24], whereas algorithms for higher order graph clustering are given in [YBL18]. Image clustering also has connections to graph clustering [CLYW24], but it is also a field on its own right [GLLT24]. Recent advances include mask recovery [LLT24], unconstrained semisupervised convex symmetric matrix factorization [LWZ24], and convolutional autoencoders with separable orthogonal moments [BEOZZ24], while [WYD⁺24] and [TA24] provide an overview of the field.

4.3 Clustering cognitive tasks with SOMs

4.3.1 SOM architecture

Generally speaking an SOM is a neural network architecture performing clustering by indirectly connecting two vector spaces of vastly different dimensions, namely the data space $\mathcal{V}$ and the coordinate space or grid C . Typically the original data space $\mathcal{V}$ is generated by sampling a higher dimensional linear manifold. Each computational neuron belongs to a grid which frequently has one or two dimensions, with the latter option being way more popular. After an iterative learning process heavily involving geometry the original data space $\mathcal{V}$ has been projected to the SOM grid C where the primary topological properties of the former preserved up to an extent in the latter. During training the role of geometry is twofold:

- ◆ The topological properties the original data space $\mathcal{V}$ dictate possible regions of interest which in turn indirectly dictate cluster shapes and the boundaries between them in the resulting grid.
- ◆ The distance metric between vectors in the data space $\mathcal{V}$ determines various local geometric loci based on the linear manifold which are reflected in the clusters in the final coordinate space.

The ability of SOMs to generate this low dimensional representation of $\mathcal{V}$ in C with the latter preserving topological similarities and neighborhoods of the former typically yields smooth clusters which can be naturally used for visualization. Smoothness is crucial in the progressive formulation of the resulting cognitive map since this ensures than knowledge incorporated in the cluster structure is not destroyed in successive iterations. Therefore, convergence proceeds more or less smoothly over the iterations.

In the following the algebraic description of SOMs will be mixed with geometric concepts, since the former can be far more easier coded. Without loss of generality, the following concepts can be applied also to one dimensional grids. In figure (4.1) the operation of an SOM is conceptually shown. Observe therein that the two upper points of the original linear manifold are mapped in the same grid column as their horizontal distance is insufficient to place them in different columns. However, their vertical distance more than suffices for placing them in different rows, in this case in two successive ones.

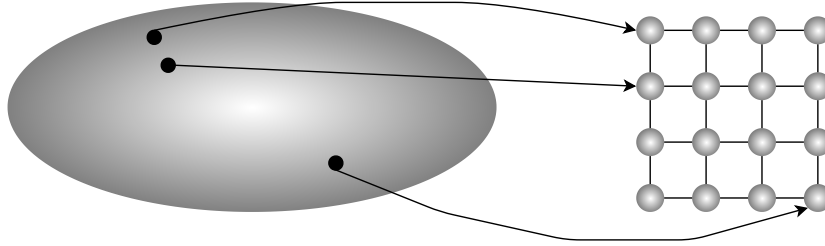


Figure 4.1: SOM projection.

In the discussion which follows let n denote the number of available data vectors or data points and d their fixed dimensionality. The latter holds true even when the data points are of variable length such as strings. In this case, d is taken to be the maximum string length with respect to its encoding.

Definition 4.1 (Input vector space) *The input space $\mathcal{V}$ is the d -dimensional space containing the set of the n available data points. Each such vector is assumed to consist of d numerical features.*

$$S \triangleq \{s_k\}_{k=1}^n \tag{4.1}$$

Definition 4.2 (Coordinate space or grid) *The coordinate space C is a $p \times q$ two dimensional discrete space.*

Each neuron u_k of the grid has a dual role as it is assigned both a location vector $\text{loc}(u_k) \in C$ and a weight vector $\text{weight}(u_k) \in \mathcal{V}$. As such, the weight vector of every neuron may very well be used in linear algebraic operations in $\mathcal{V}$ involving elements of S . These are in turn indirectly reflected in C . This connection is critical for the clustering ability of an SOM. It should be noted that, although it is straightforward to follow this connection from $\mathcal{V}$ to C , the reverse in general is not true. This is a direct consequence of the dimensionality reduction and the associated information loss.

$$u_k \leftrightarrow \begin{bmatrix} x_k \\ y_k \end{bmatrix} \in C \quad \text{and} \quad u_k \leftrightarrow \text{weight}(u_k) \triangleq \begin{bmatrix} w_k[1] \\ \vdots \\ w_k[d] \end{bmatrix} \in \mathcal{V} \quad (4.2)$$

In the general case there is no strict criterion to determine the grid size. Thus, the empirical rule of equation (4.3) proposed in [Koh90] which takes into consideration only the total number of data points and roughly connects the grid size with the entropy of the dataset at hand will be used. Moreover, the grid has equal dimensions as there is no *a priori* reason to favor one dimension over the other. The constants p_0 and q_0 can be arbitrarily selected or fit by least squares to the estimated entropy of the dataset, provided that it can be efficiently computed. Alternatively, q_0 may be multiplied by a function with a sublogarithmic growth rate so that the logarithm will remain the dominant term of the grid size as n grows.

$$\begin{aligned} p &= p_0 \lceil \log n \rceil + q_0 & \text{or} & & p &= p_0 \lceil \log n \rceil + q_0 o(\log n) \\ q &= p & & & q &= p \end{aligned} \quad (4.3)$$

Given that in the general case more than one data vectors from $\mathcal{V}$ can be mapped to the neural weights of the same neuron in the grid, the location relationship between a data vector and a neuron and the assignments of the former to the latter must be formally expressed. This is algorithmically expressed by the pair of relationships $\text{loc}(\cdot)$ and $\text{invloc}(\cdot)$ as explained in detail in definition 4.3 below.

Definition 4.3 (Neuron and data point relationship) *The relationship $\text{loc}(\cdot)$ yields the neuron a given data vector has been assigned to. Additionally, assuming that f_k data vectors from S have been assigned to u_k during the SOM training, then the relationship of equation (4.4) returns the set of these data points.*

$$\begin{aligned} \text{loc}(s_b) &\triangleq u_k \\ \text{invloc}(u_k) &\triangleq \{s_{k_1}, \dots, s_{k_{f_k}}\} \triangleq S_k \subseteq S \end{aligned} \quad (4.4)$$

The upper part of equation (4.4) given a data point s_b gives the neuron it has been assigned to. The lower given a neuron returns a set S_k consisting of data vectors and therefore it is not a function. This is a direct consequence of the clustering operation as many vectors are bound to be matched to the same neuron.

Depending on the grid shape and the adjacency rules a neuron has a varying set of neighboring ones. Although these factors influence the learning process and the resulting cluster structure, they are not reflected in the data vector $\mathcal{V}$. Nonetheless, they are an essential part of the SOM as a data processing unit. In a rectangular grid the neighboring neurons can very well form a cross, a square, a hexagon, or even a circle. In this dissertation the cross has been selected as indicated by figure 4.2. Definition 4.4 gives both the formal indicator function for determining neighbors based on which the set of neighbors is built.

Definition 4.4 (Neighboring neurons) Any neuron adjacent in C to a given neuron u_k according to the grid size and adjacency strategy is termed to be a neighbor of u_k and satisfies the indicator function of (4.5).

$$\text{neighb}(u_i, u_j) \triangleq \begin{cases} 1, & u_i \text{ and } u_j \text{ are neighboring} \\ 0, & \text{otherwise} \end{cases} \quad (4.5)$$

Furthermore, for each neuron u_i the neighborhood set $\Gamma(u_i)$ is defined as in equation (4.6).

$$\Gamma(u_i) \triangleq \{u_j \in C \mid \text{neighb}(u_i, u_j) = 1\} \quad (4.6)$$

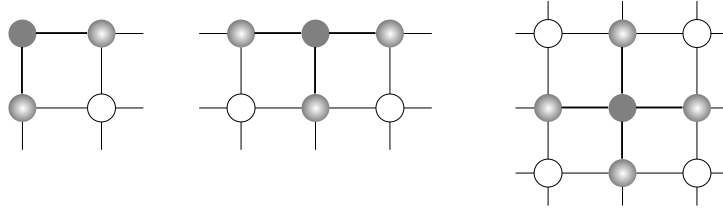


Figure 4.2: Neighboring neurons in a rectangular grid with a cross strategy.

The SOM training process relies heavily on three functions, one in $\mathcal{V}$ and two in C , in order to yield the cluster structure. As stated earlier in section 4.1 these functions are the distance metric $d(\cdot, \cdot)$ in $\mathcal{V}$, the proximity metric $g(\cdot, \cdot)$ in C , and the weight function $h(\cdot, \cdot)$. The first evaluates distances between data points and synaptic weights, the second grid distances, and the third rewards within a cluster where its maximum value is always at the cluster center. In certain scenarios $h(\cdot, \cdot)$ and $g(\cdot, \cdot)$ are related, but they do not need to be so in the general case. The weight function also plays a central role in the formation for each neuron of its proximity set of definition 4.5. Common options include the following.

The Gaussian kernel of (4.7) decays smoothly allowing the formulation of compact clusters.

$$h(u_i, u_j; \sigma_0) \triangleq \frac{1}{\sigma_0 \sqrt{2\pi}} \exp\left(-\frac{\|u_i - u_j\|_1^2}{2\sigma_0^2}\right) \quad (4.7)$$

The triangular function of (4.8) has a linear decay rate.

$$h(u_i, u_j; \gamma_0) \triangleq 1 - \frac{\|u_i - u_j\|_1}{\gamma_0} \quad (4.8)$$

The circular weight function of (4.9) forms a semicircle around neuron u_j .

$$h(u_i, u_j; \rho_0) \triangleq \frac{\sqrt{\rho_0^2 - \|u_i - u_j\|_2^2}}{\rho_0} \quad (4.9)$$

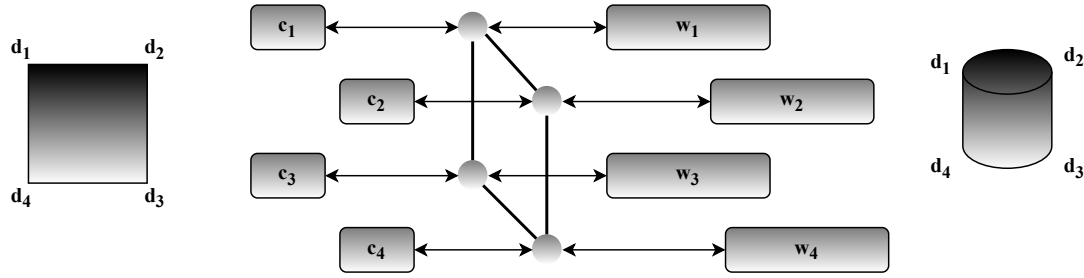


Figure 4.3: Projection of a cylinder to a square.

Definition 4.5 (Proximity set) The proximity set $\Delta(u_i; \xi_0)$ for a neuron u_i given a threshold ξ_0 is the set of neurons for which the reward function $h(\cdot, \cdot)$ remains above ξ_0 as shown in equation (4.10).

$$\Delta(u_i; \xi_0) \triangleq \{u_j \in C \mid h(u_i, u_j) \geq \xi_0\} \quad (4.10)$$

The proximity set $\Delta(u_i; \xi_0)$ of a given neuron u_i differs heavily from its neighborhood $\Gamma(u_i)$ introduced in definition (4.4). The former corresponds to the effective area of a cluster around u_i whereas the latter only to the centroid of that cluster. Therefore, $\Delta(u_i; \xi_0)$ includes $\Gamma(u_i)$ but also always extends beyond that. The limiting case where the two sets coincide is very rarely used in practice, if at all as in such as case the clusters would have literally zero periphery and clustering would reduce to moving points around the SOM grid. To this end the cluster density and the effective cluster density during epoch r are defined as in (4.11). Therein $D^{[r]}$ is the activation set of (4.13). The former expresses the size of cluster periphery to that of its centroid which is formed around neuron u_i , whereas the latter is the harmonic mean taken over all such best matching neurons. The harmonic mean is robust to numerical instabilities such as floating point perturbations and catastrophic cancellation and tends to ignore outliers.

$$\eta^{[r]}(u_i; \xi_0) \triangleq \frac{|\Delta(u_i; \xi_0)|}{|\Gamma(u_i)|} \quad \text{and} \quad \bar{\eta}^{[r]}(\xi_0) \triangleq \frac{|D^{[r]}|}{\sum_{u \in D^{[r]}} \frac{|\Gamma(u)|}{|\Delta(u; \xi_0)|}} \quad (4.11)$$

This is one of the many ways geometry shapes SOM functionality. Indeed this point should be emphasized as geometry plays an important role in SOM, as seen by the connections between $\mathcal{V}$ and C , the preservation of topology, and the interplay between the proximity and weight functions in C . Geometric constraints are typically harder to code and check for than algebraic ones since a considerably larger of data points should be checked and computations may be highly nonlinear. However, geometry offers better insight into the data space $\mathcal{V}$ and allows for meaningful visualization. Figure 4.3 depicts the projection of a cylinder in the data space to a square in the coordinate space.

4.3.2 Training process

During epoch r the best matching neuron u^* for the input vector s_k is the neuron whose weight vector is closest in $\mathcal{V}$ to s_k as evaluated by metric $d(\cdot, \cdot)$ as shown in (4.12).

$$u^* \triangleq \operatorname{argmin} [d(s_k, \text{weight}(u))] \quad (4.12)$$

Notice that the distance metric in (4.12) is computed in $\mathcal{V}$. Once it is found, u^* is added to the set $D^{[r]}$ of activated neurons for that given epoch is formed for the given epoch r as in (4.13). In case of a tie, especially in early iterations, a tie breaker is used, typically the neuron which is the farthest from every other cluster centroid or the neuron with the smallest coordinate vector. This illustrates the importance of the weight vector initialization strategy, in this particular case a multivariate Gaussian probability distribution.

$$D^{[r]} \leftarrow D^{[r]} \cup u^* \quad (4.13)$$

The synaptic weights of the best matching neuron u^* are updated in each projection of the epoch r based on a modified Hebbian update rule as per equation (4.14). Besides u^* the synaptic weights of each neuron $u' \in \Delta(u^*; \xi_0)$ are also updated as shown in the lower equation of equation (4.14). Therein please observe the role of the reward function $h(\cdot, \cdot)$ as well as that the actual reward is based on the distance between s_k and u' for the case of u' . Therefore, additional distances need to be computed depending on the size of the proximity size of u^* . In later epochs and for large clusters the same reward may be given to neurons.

$$\begin{aligned} \Delta^{[r]} \text{ weight}(u^*) &= \mu^{[r]} d(s_k, \text{weight}(u^*)) \\ \Delta^{[r]} \text{ weight}(u') &= \mu^{[r]} h(u', u^*) d(s_k, \text{weight}(u')) \end{aligned} \quad (4.14)$$

The proposed multilinear distance metric is shown in (4.15) and it includes bilinear and quadratic functions as special cases. It converts iamges to very long vectors and weighs them in the spatial frequency domain according to the relative location and value of the pixels. This allows more flexibility as different tensor components can receive different weights. The weight matrix G is defined elementwise as a first order lowpass Butterworth filter whose values depend only on the locations of the elements of s involved. The following two properties of the two dimensional discrete cosine transform (DCT2) will be utilized. First, given that images have only real values, the transformation spectrum will also consist of real values exclusively. Additionally, the latter will be sparse with nonzero values concentrated along the low spatial frequencies, making the application of transform domain filters computationally efficient.

$$\begin{aligned} J(X, Y; G) &\triangleq \left(\text{vec}(X - Y)^T G \text{vec}(X - Y) \right)^{\frac{1}{2}} = \sqrt{s^T G s} \\ s &\triangleq \text{vec}(X - Y) \end{aligned} \quad (4.15)$$

The original data points are three stacked fMRI images for each subject as shown in equation (4.16).

$$X_k \triangleq \begin{bmatrix} X_{k,1} & X_{k,2} & X_{k,3} \end{bmatrix} \quad (4.16)$$

The weight matrix G consists of the DCT2 matrix M , the filtering matrix F containing the Butterworth filter, the identity matrix I , and the 3×1 auxiliary vector $1_{3,1}$ whose elements equal 1 as shown in (4.17).

$$G \triangleq (1_{3,1} \otimes M)^T (I \otimes F) (1_{3,1} \otimes M) \quad \text{with} \quad 1_{3,1} \otimes M = \begin{bmatrix} M \\ M \\ M \end{bmatrix} \quad \text{and} \quad I \otimes F = \begin{bmatrix} F & & \\ & F & \\ & & F \end{bmatrix} \quad (4.17)$$

The advantages of using the distance metric equation (4.14) include the possibility of clustering images in more than one ways as matrix G can be extended. Different aspects can receive different weights or, alternatively, each image can be resampled by multiple feature vectors increasing flexibility and providing insight into clustering dynamics. Moreover, the proposed metric can be combined with nonlinear kernels or approximations thereof, paving therefore the way for advanced clustering schemes.

Perhaps the most common and intuitive SOM error metric is the topological error explained in definition 4.6. As local coherence preservation is of primary importance in the final cognitive map, it is only natural to measure the average probability that a data point is assigned to the periphery of a cluster.

Definition 4.6 (Topological error rate) *The topological error $T[r]$ for epoch r is the ratio of the number of data points which are mapped to $\Delta(u^*) \setminus \Gamma(u^*)$ as shown in (4.18).*

$$T[r] \triangleq \frac{1}{n} \sum_{s_k \in S} |\{\text{loc}(s_k) \notin \{u \cup \Gamma(u) \mid u \in D[r]\}\}| \quad (4.18)$$

Table 4.1: Distribution of tasks.

	T1	T2	T3	Subtotal
Neutral	6	5	5	16
Negative	6	6	5	17
Subtotal	12	11	10	33

Table 4.2: SOM configuration options.

$d(\cdot, \cdot)$	$g(\cdot, \cdot)$	$h(\cdot, \cdot)$	μ
T: tensor	G: Gaussian	G: Gaussian	S: cosine
L2: ℓ_2 norm	C: circular	C: circular	E: exponential
L1: ℓ_1 norm	R: triangle	R: triangle	P: inverse polynomial

Table 4.3: SOM configurations.

#	Configuration	#	Configuration	#	Configuration
1	$(p_0, q_0, L1, C, C, E)$	10	$(p_0, q_0, L2, C, C, E)$	19	(p_0, q_0, T, C, C, E)
2	$(p_0, q_0, L1, C, C, P)$	11	$(p_0, q_0, L2, C, C, P)$	20	(p_0, q_0, T, C, C, P)
3	$(p_0, q_0, L1, C, C, S)$	12	$(p_0, q_0, L2, C, C, S)$	21	(p_0, q_0, T, C, C, S)
4	$(p_0, q_0, L1, R, R, E)$	13	$(p_0, q_0, L2, R, R, E)$	22	(p_0, q_0, T, R, R, E)
5	$(p_0, q_0, L1, R, R, P)$	14	$(p_0, q_0, L2, R, R, P)$	23	(p_0, q_0, T, R, R, P)
6	$(p_0, q_0, L1, R, R, S)$	15	$(p_0, q_0, L2, R, R, S)$	24	(p_0, q_0, T, R, R, S)
7	$(p_0, q_0, L1, G, G, E)$	16	$(p_0, q_0, L2, G, G, E)$	25	(p_0, q_0, T, G, G, E)
8	$(p_0, q_0, L1, G, G, P)$	17	$(p_0, q_0, L2, G, G, P)$	26	(p_0, q_0, T, G, G, P)
9	$(p_0, q_0, L1, G, G, S)$	18	$(p_0, q_0, L2, G, G, S)$	27	(p_0, q_0, T, G, G, S)

4.3.3 Results

The anonymized dataset for the experiments of this section was taken from [LMM⁺19b] and it is derived from a project for studying the ageing human brain. It has been publicly uploaded to openneuro.org under the Creative Commons license¹. According to the dataset authors, the latter consists of male and female

¹dataset doi: 10.18112/openneuro.ds002366.v1.0.0.

participants, all right handed with no health problems. Each participant was presented for three seconds an image of neutral or negative affective polarity selected from the International Affective Picture System and then was issued an audio instruction to *suppress*, *maintain*, or *enhance* their emotions. These are the three cognitive tasks, respectively denoted as **T1**, **T2**, and **T3**. Table 4.1 has frequency distribution for the three cognitive tasks and the corresponding sentiment polarity. The SOM configurations used along with their numbering are shown in table 4.3. The options for the distance metric, the proximity metric, the reward function, and the learning rate are explained in table 4.2.

Table 4.4: Best (left) and average (right) confusion matrices.

Metric	Conf #	Task (%)	T1	T2	T3	T1	T2	T3
ℓ_1 norm	8	T1	78.66	8.33	13.00	77.11	11.89	11.00
		T2	10.66	75.00	14.33	11.00	74.50	14.50
		T3	4.00	7.00	79.00	5.20	8.20	76.60
ℓ_2 norm	18	Task (%)	T1	T2	T3	T1	T2	T3
		T1	79.66	8.00	12.33	78.00	7.50	14.50
		T2	9.00	81.00	10.00	7.30	79.40	13.30
proposed	27	T3	7.66	8.33	83.00	5.00	13.50	81.50
		Task (%)	T1	T2	T3	T1	T2	T3
		T1	91.01	6.99	2.00	88.50	11.16	1.33
		T2	2.33	85.33	12.33	2.00	84.00	14.00
		T3	5.00	8.66	87.33	3.00	11.50	86.50

In table 4.4 the confusion matrices for the three distance metrics tested, namely the ℓ_1 norm, the ℓ_2 norm, and that of equation (4.15). Specifically, the matrices obtained from the configuration which achieved the least misclassification rate and from aggregating every other configuration are given. The proposed metric consistently yields lower misclassification rates, which can be explained by the flexibility inherent in it. Regarding norms ℓ_2 outperforms ℓ_1 , which can be attributed to the isotropic curvature of the former. The confusion between **T2** and **T3** was consistently higher than that any other class pair.

4.4 Bibliography

- [ABJ⁺24] Njud S Alharbi, Stelios Bekiros, Hadi Jahanshahi, Jun Mou, and Qijia Yao. Spatiotemporal wavelet-domain neuroimaging of chaotic EEG seizure signals in epilepsy diagnosis and prognosis with the use of graph convolutional LSTM networks. *Chaos, Solitons & Fractals*, 181, 2024.
- [ALS24] Mohammed G Alsubaie, Suhui Luo, and Kamran Shaukat. Alzheimer’s disease detection using deep learning on neuroimaging: A systematic review. *Machine Learning and Knowledge Extraction*, 6(1):464–505, 2024.
- [AYZA⁺24] Hadi Abbasian, Reza Yousefi-Zenouz, Abdollah Amirkhani, Masoud Shirzadeh, Akbar Abdollahiasl, Shekoufeh Nikfar, Mohammadreza Siahi-Shadabad, and Abbas Kebriaeezadeh. Risk assessment for complex systems based on fuzzy cognitive maps: A case of the biopharmaceutical industry. *Complexity*, 2024(1), 2024.

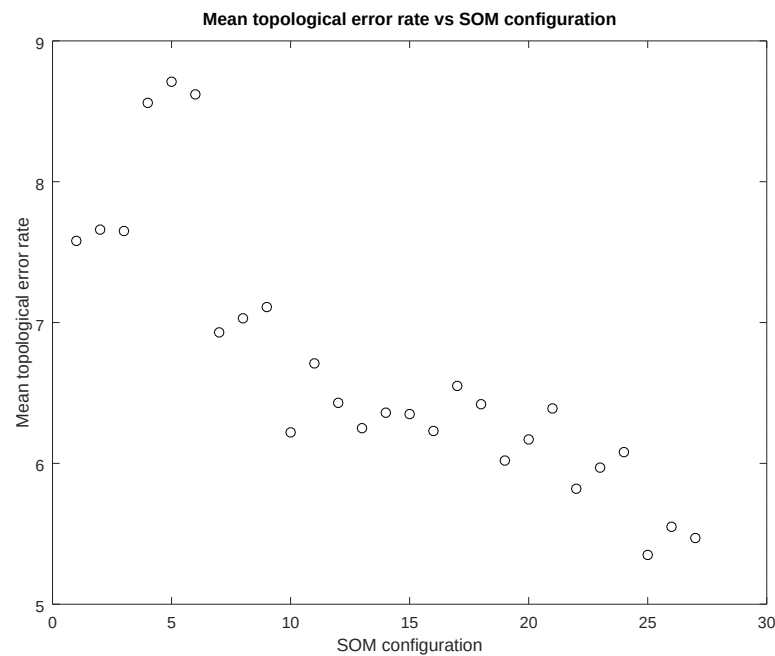


Figure 4.4: Mean topological error vs SOM configuration.

- [BDS⁺24] Kamal Berahmand, Fatemeh Daneshfar, Elaheh Sadat Salehi, Yuefeng Li, and Yue Xu. Autoencoders and their applications in machine learning: A survey. *Artificial Intelligence Review*, 57(2):28, 2024.
- [BEOZZ24] Aymane Bouali, Ilham El Ouariachi, Azeddine Zahi, and Khalid Zenkouar. Robust deep image clustering using convolutional autoencoder with separable discrete Krawtchouk and Hahn orthogonal moments. *Intelligent Systems with Applications*, 22, 2024.
- [BR24] Sayed Pedram Haeri Boroujeni and Abolfazl Razi. IC-GAN: An improved conditional generative adversarial network for RGB-to-IR image translation with applications to forest fire monitoring. *Expert Systems with Applications*, 238, 2024.
- [BSZ⁺24] Yanan Bai, Shiliang Shao, Jin Zhang, Xianzhe Zhao, Chuxi Fang, Ting Wang, Yongliang Wang, and Hai Zhao. A review of brain-inspired cognition and navigation technology for mobile robots. *Cyborg and Bionic Systems*, 2024.
- [CCAL⁺24] Nairen Cao, Vincent Cohen-Addad, Euiwoong Lee, Shi Li, Alantha Newman, and Lukas Vogl. Understanding the cluster linear program for correlation clustering. In *STOC*, pages 1605–1616. ACM, 2024.
- [CDW⁺24] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *International Journal of Computer Vision*, 132(1):208–223, 2024.

- [CLYW24] Guikun Chen, Xia Li, Yi Yang, and Wenguan Wang. Neural clustering based visual representation learning. In *CVPR*, pages 5714–5725. IEEE/CVF, 2024.
- [CRP⁺24] Chenhang Cui, Yazhou Ren, Jingyu Pu, Jiawei Li, Xiaorong Pu, Tianyi Wu, Yutao Shi, and Lifang He. A novel approach for effective multi-view clustering with information-theoretic perspective. *Advances in Neural Information Processing Systems*, 36, 2024.
- [DGP⁺24] Benoit Dufumier, Pietro Gori, Sara Petiton, Robin Louiset, Jean-François Mangin, Antoine Grigis, and Edouard Duchesnay. Exploring the potential of representation and transfer learning for anatomical neuroimaging: application to psychiatry. *NeuroImage*, 2024.
- [DHF⁺24] Yulin Dai, Yu-Chun Hsu, Brisa S Fernandes, Kai Zhang, Xiaoyang Li, Nitesh Enduru, Andi Liu, Astrid M Manuel, Xiaoqian Jiang, and Zhongming Zhao. Disentangling accelerated cognitive decline from the normal aging process and unraveling its genetic components: A neuroimaging-based deep learning approach. *Journal of Alzheimer’s Disease*, preprint(preprint):1–21, 2024.
- [DMH⁺24] Gopikrishna Deshpande, Janzaib Masood, Nguyen Huynh, Thomas Denney, and Michael N Dretsch. Interpretable deep learning for neuroimaging-based diagnostic classification. *IEEE Access*, 2024.
- [DSYA24] Fatemeh Daneshfar, Sayvan Soleymanbaigi, Pedram Yamini, and Mohammad Sadra Amini. A survey on semi-supervised graph clustering. *Engineering Applications of Artificial Intelligence*, 133, 2024.
- [EKM⁺24] Hossein Esfandiari, Amin Karbasi, Vahab Mirrokni, Grigoris Velezgas, and Felix Zhou. Replicable clustering. *Advances in Neural Information Processing Systems*, 36, 2024.
- [FLH24] Scott D Foster, Emma Lawrence, and Andrew J Hoskins. Spatially clustered survey designs. *Journal of Agricultural, Biological and Environmental Statistics*, 29(1):130–146, 2024.
- [GFA⁺24] Gad Gad, Aya Farrag, Ahmed Aboulfotouh, Khaled Bedda, Zubair Md Fadlullah, and Mostafa M Fouda. Joint self-organizing maps and knowledge distillation-based communication-efficient federated learning for resource-constrained UAV-IoT systems. *IEEE Internet of Things Journal*, 2024.
- [GLLT24] Renxiang Guan, Zihao Li, Xianju Li, and Chang Tang. Pixel-superpixel contrastive learning and pseudo-label correction for hyperspectral image clustering. In *ICASSP*, pages 6795–6799. IEEE, 2024.
- [GLZ24] Ghadeer O Ghosheh, Jin Li, and Tingting Zhu. A survey of generative adversarial networks for synthesizing structured electronic health records. *ACM Computing Surveys*, 56(6):1–34, 2024.
- [GN24] Philippe J Giabbanelli and Gonzalo Nápoles. *Fuzzy Cognitive Maps: Best Practices and Modern Methods*. Springer Nature, 2024.
- [GND⁺20] Attri Ghosal, Arunima Nandy, Amit Kumar Das, Saptarsi Goswami, and Mrityunjoy Panday. A short review on different clustering techniques and their applications. *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018*, pages 69–83, 2020.

- [GRS⁺24] Wilhelm Gros, Lisa Reuter, Julia Sprich, Dennis Schuldzinski, Julius Fenn, and Andrea Kiesel. Cognitive-affective maps (CAMs) as measurement tool—elaboration of quantitative and qualitative test-retest reliability. *Technology in Society*, 2024.
- [GS24] Vahid Gholami and Hossein Sahour. A hybrid approach of supervised self-organizing maps and genetic algorithms for predictive mapping of arsenic pollution in groundwater resources. *Exposure and Health*, 16(3):775–790, 2024.
- [HMB24] Christopher Holder, Matthew Middlehurst, and Anthony Bagnall. A review and evaluation of elastic distance functions for time series clustering. *Knowledge and Information Systems*, 66(2):765–809, 2024.
- [HT24] Akifumi Hagiwara and Masaya Takahashi. Advanced techniques for MR neuroimaging. *Magnetic Resonance in Medical Sciences*, 23(3):249–251, 2024.
- [IAT24] Ikramuddin Ikramuddin, Sandhya Avasthi, and Meenakshi Tyagi. Clustering methods for multidimensional data from social media. In *MIT Art, Design and Technology School of Computing International Conference (MITADTSociCon)*, pages 1–7. IEEE, 2024.
- [JKV24] J Jeyalakshmi, J Lokesh Kumar, and S Vishnu. Neuroimaging analysis for early diagnosis of ADNI progression with EfficientNet. In *ICCS*, pages 1555–1560. IEEE, 2024.
- [JRMK24] Ahmadali Jamali, Mohsen Rostamy-Malkhalifeh, and Reza Kargar. The novel self-organizing map combined with fuzzy C-means and K-means convolution for a soft and hard natural digital image segmentation. *AUT Journal of Mathematics and Computing*, 5(2):151–165, 2024.
- [JRS⁺24] Sadia Jamil, Eid Rehman, Tariq Shahzad, Muhammad Ishtiaq, Tehseen Mazhar, Yazeed Yasin Ghadi, and Arfan Ahmed. Multi self-organizing map (SOM) pipeline architecture for multi-view clustering. *IEEE Access*, 2024.
- [KBM24] Caleb Schultz Kisby, Saúl A Blanco, and Lawrence S Moss. What do Hebbian learners learn? reduction axioms for iterated hebbian learning. In *AAAI*, volume 38, pages 14894–14901, 2024.
- [Kiv96] Kimmo Kiviluoto. Topology preservation in self-organizing maps. In *ICNN*, volume 1, pages 294–299. IEEE, 1996.
- [KKL90] Jari Kangas, Teuvo Kohonen, and Jorma Laaksonen. Variants of self-organizing maps. *IEEE transactions on neural networks*, 1(1):93–99, 1990.
- [Koh90] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.
- [Koh97] Teuvo Kohonen. Exploration of very large databases by self-organizing maps. In *ICNN*, volume 1, pages PL1–PL6. IEEE, 1997.
- [LBM⁺24] Jeyeon Lee, Brian J Burkett, Hoon-Ki Min, Matthew L Senjem, Ellen Dicks, Nick Corriveau-Lecavalier, Carly T Mester, Heather J Wiste, Emily S Lundt, Melissa E Murray, et al. Synthesizing images of tau pathology from cross-modal neuroimaging using deep learning. *Brain*, 147(3):980–995, 2024.

- [LCW⁺24] James Liang, Yiming Cui, Qifan Wang, Tong Geng, Wenguan Wang, and Dongfang Liu. Clusterfomer: Clustering as a universal visual learner. *Advances in neural information processing systems*, 36, 2024.
- [LLT24] Xu Liang, Chen Li, and Lihua Tian. Clustering-based mask recovery for image captioning. *Neurocomputing*, 2024.
- [LMM⁺19a] William Lloyd, Jayne Morriss, Birthe Macdonald, Karin Joanknecht, Julie Sigurd, and Carien van Reekum. Longitudinal change in executive function is associated with impaired top-down frontolimbic regulation during reappraisal in older adults. *bioRxiv*, 2019.
- [LMM⁺19b] William Lloyd, Jayne Morriss, Birthe Macdonald, Karin Joanknecht, Julie Sigurd, and Carien van Reekum. Longitudinal change in executive function is associated with impaired top-down frontolimbic regulation during reappraisal in older adults. *bioRxiv*, 2019.
- [LRU20] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive data sets*. Cambridge University Press, 2020.
- [LWZ24] Wenjun Luo, Zezhong Wu, and Nan Zhou. Hypergraph-based convex semi-supervised unconstrained symmetric matrix factorization for image clustering. *Information Sciences*, 2024.
- [MPADB24] Prabha Susy Mathew, Anitha S Pillai, Ajith Abraham, and Lazzaro Di Biase. Neuroimaging and deep learning in stroke diagnosis. *Machine Learning and Deep Learning in Neuroimaging Data Analysis*, 2024.
- [MSH⁺24] Lindsay Munroe, Mariana da Silva, Faezeh Heidari, Irina Grigorescu, Simon Dahan, Emma C Robinson, Maria Deprez, and Po-Wah So. Applications of interpretable deep learning in neuroimaging: A comprehensive review. *Imaging Neuroscience*, 2024.
- [OT23] Gbeminiyi John Oyewole and George Alex Thopil. Data clustering: Application and trends. *Artificial Intelligence Review*, 56(7):6439–6475, 2023.
- [OTP⁺24] James Oldfield, Christos Tzelepis, Yannis Panagakis, Mihalis A Nicolaou, and Ioannis Patrassas. Bilinear models of parts and appearances in generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [SSL24] Lin Sha, Yabin Shao, and Yuzhen Li. A framework of fermatean fuzzy cognitive map and its extension based on Hamacher operation. *Engineering Applications of Artificial Intelligence*, 136, 2024.
- [TA24] Ganji Tejasree and Loganathan Agilandeeswari. An extensive review of hyperspectral image classification and prediction: Techniques and challenges. *Multimedia Tools and Applications*, pages 1–98, 2024.
- [TGZ⁺24] Wenxuan Tu, Renxiang Guan, Sihang Zhou, Chuan Ma, Xin Peng, Zhiping Cai, Zhe Liu, Jieren Cheng, and Xinwang Liu. Attribute-missing graph clustering network. In *AAAI*, volume 38, pages 15392–15401, 2024.
- [TSM⁺99] Pablo Tamayo, Donna Slonim, Jill Mesirov, Qing Zhu, Sutisak Kitareewan, Ethan Dmitrovsky, Eric S Lander, and Todd R Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *PNAS*, 96(6):2907–2912, 1999.

- [V⁺24] Christel Vrain et al. A review on declarative approaches for constrained clustering. *International Journal of Approximate Reasoning*, 2024.
- [WAM19] Chathurika S Wickramasinghe, Kasun Amarasinghe, and Milos Manic. Deep self-organizing maps for unsupervised image classification. *IEEE Transactions on Industrial Informatics*, 15(11):5837–5845, 2019.
- [WKPP24] Bernard Widrow, Youngsik Kim, Dookun Park, and Jose Krause Perin. Nature’s learning rule: The Hebbian-LMS algorithm. In *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, pages 11–40. Elsevier, 2024.
- [WYD⁺24] Pingxin Wang, Xibei Yang, Weiping Ding, Jianming Zhan, and Yiyu Yao. Three-way clustering: Foundations, survey and challenges. *Applied Soft Computing*, 151, 2024.
- [YBL18] Hao Yin, Austin R Benson, and Jure Leskovec. Higher-order clustering in networks. *Physical Review E*, 97(5), 2018.
- [YC24] Roh-Eul Yoo and Seung Hong Choi. Deep learning-based image enhancement techniques for fast MRI in neuroimaging. *Magnetic Resonance in Medical Sciences*, 2024.
- [YQH24] Jiawei Yao, Qi Qian, and Juhua Hu. Multi-modal proxy learning towards personalized visual multiple clustering. In *CVPR*, pages 14066–14075. IEEE/CVF, 2024.
- [YWD⁺24] Shengju Yu, Siwei Wang, Zhibin Dong, Wenxuan Tu, Suyuan Liu, Zhao Lv, Pan Li, Miao Wang, and En Zhu. A non-parametric graph clustering framework for multi-view data. In *AAAI*, volume 38, pages 16558–16567, 2024.
- [YZL⁺24] Zhiyuan Yuan, Fangyuan Zhao, Senlin Lin, Yu Zhao, Jianhua Yao, Yan Cui, Xiao-Yong Zhang, and Yi Zhao. Benchmarking spatial clustering methods with spatially resolved transcriptomics data. *Nature Methods*, 21(4):712–722, 2024.

Chapter 5

Graph neural networks

*You can't connect the dots looking forward;
you can only connect them looking backwards.
So you have to trust that the dots will somehow
connect in your future*
Steve Jobs (1955 – 2011)

5.1 Chapter introduction

CLUSTERING is one of the most important tasks in knowledge mining and in related scientific domains as it allows the discovery of structure inherent in potentially large data batches or even in entire data lakes based on a small number of assumptions typically pertaining to the nature of the data and formulated by what is sought after in the data under analysis. Although it makes perfect sense to wonder whether such properties actually exist or are erroneous generalizations of coincidental, circumstantial, or local ones. The latter in fact is a major consideration in scale-free graphs where emergence phenomena tend to appear only after a critical graph attribute such as density or vertex size reaches a certain lower threshold. That being said, the nature of the data can support some fundamental properties or even the construction of a suitably defined data space with the proper embeddings. Prime examples of this are distance metrics for categories, vectors, and sets and the density of the data space. Even simple distance metrics like the

commonly used Euclidean distance come with assumptions: The Euclidean distance implies that data points are taken from a continuous space, coordinate squares have an actual meaning (e.g. they correspond to energy or composition of edges), and straight lines are valid geometrical constructs therein. Because of the intractability of clustering in its general form, the lack of ground truth, and recently because of the sheer dataset size, a plethora of clustering schemes has been developed for various purposes [LCW⁺24].

Graph neural networks (GNNs) have recently emerged as a versatile architectural paradigm whose main properties are adaptivity, iterativity, and an orientation towards connections [LXD⁺24]. Through a message passing mechanism and a processing layer, frequently an activation function or an aggregator, in a properly configured GNN a single global state can eventually emerge from a diversity of local ground truth information stored originally at the vertices [SSYR24]. In this way is captured and exploited any major interplay in the underlying domain as long as they can be expressed in a compatible form for storing and processing. Current field research focuses on the following fundamental topics:

- ♣ The conditions necessary for the emergence of the global steady state. Since in the general case it is known that it is impossible, in fact Turing undecidable, to prove that each such state can be the final result of any type of computation done in the nodes starting with arbitrary initial ground truth vectors, it is necessary that classes of convergence be defined. This aligns with the emergence phenomena in evolving graphs under most graph evolution models.
- ♣ How fast the converge to this steady state can be achieved. This is primarily a function of the complexity of the GNN architecture in topological terms, the processing taking place in each node, and more recently of the reliability of the communication mechanism in terms of noise strength. In turn the latter depends heavily on the semantics of the underlying field.
- ♣ How robust the steady state is in the sense that once it emerges, then the GNN cannot move away from even when subject to external input under certain regularity conditions. Moreover, a related topic is that of how will the convergence rate be delayed if the ground truth vectors are incomplete, estimated, or corrupted by noise or even by numerical errors.

Recent advances in computational science and technology have rendered feasible the development of efficient methodologies, even in the form of sophisticated heuristics, for large scale humanistic data. High dimensional data are frequently generated across a broad spectrum of fields ranging from digital medicine and smart city management to finance, e-commerce, space exploration, and social media. Data dimensionality as well as variability, including missing fields, depends heavily on the underlying field. Nonetheless, the fundamental goal remains essentially the same, namely the description of time varying entities from different perspectives and possible at various levels of granularity as deemed appropriate. Social networks are no exception to this rule as they model entities, either individuals or organizations, at Internet scale. Additionally, social graphs, and any type of graphs for that matter, can inherently represent high dimensional data in a recursive and distributed way. Indeed, local information can be augmented with that of connected entities or with information obtained from the local digital communities in order to yield a better sketch of the entity under consideration. Additionally, information from entities can be combined in a bottom-up manner in order to construct a description of the community comprising of these entities.

Since the learning process in GNNs is iterative as shown in figure 5.1 and relies heavily on message passing and local aggregation, two algorithmic questions arise. First, what is the update order of the GNN vertices and, second, what is the relative weight of the local ground truth of each vertex with respect to the

information arriving from the neighborhood. Regarding the former question, there are many approaches depending on the topological configuration and the operational requirements of a given GNN.

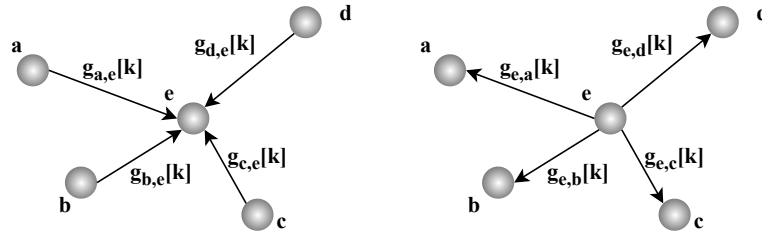


Figure 5.1: Message passing in GNNs in rounds.

Possible strategies regarding the GNN vertex update are the following. Please note that most of them are not mutually exclusive and can be combined to yield more flexible schemes.

- ♣ Input should come for every neighbor. While this may potentially impose constraints and slow GNN down, it ensures that each and every piece of information in the immediate vicinity is considered.
- ♣ Messages can come from a considerable fraction of neighbors in order to have sufficient information in a given vertex to yield a new message which in turn will be forwarded to every neighbor.
- ♣ Processing can proceed when a significant information variability or innovation has been achieved. This requires knowledge or an estimation of the respective variability of the neighborhood.

The relative weight at each vertex of the information arriving from neighbors is critical for the convergence and stability of the GNN training process. To this end, it is frequently advisable to put more weight to the local ground truth vector. In fact this is the standard update policy in *pytorch*. For an information processing perspective, this is reminiscent of the regularization technique of equation (3.38).

Another point which should be mentioned is that GNNs have close connections to graph signal processing (GSP) as information flow and processing in the former can be cast in terms of the latter. Since as mentioned earlier graphs are the primary algorithmic vehicles for inherently representing higher order data in a distributed manner while at the same time preserving any salient underlying dependencies, GNNs can be combined with the tensor distances described in detail in chapter 3 or with other schemes relying on similar principles such as higher order Fourier transforms shown in (5.1) or aggregators.

$$X(\omega_1, \dots, \omega_p) \triangleq \int_{u_1 \in \Omega_1} \dots \int_{u_p \in \Omega_p} x(u_1, \dots, u_p) e^{-i(\omega_1 u_1 + \dots + \omega_p u_p)} du_1 \dots du_p \quad (5.1)$$

In equation (5.1) $\Omega_1, \dots, \Omega_p$ are the respective domains for the variables $u_1, \dots, u_p$. A very common special case of equation (5.1) is the two-dimensional Fourier transform. In digital image processing, as in the case of chapter 4, the variables u_1 and u_2 correspond to pixel locations, whereas ω_1 and ω_2 correspond to spatial frequencies, namely . This allows the development of filters in the transform domain such as the low-pass spatial Butterworth filter [HLH24; TZQ⁺24] and other ones as well [MZH⁺24].

The connection between graphs and signal processing stems from two fundamental relationships. The first is that the graph Fourier transform is defined as the Fourier transform of its adjacency matrix as shown in equation (5.2). Therein A is the adjacency matrix of a graph and $\tilde{A}$ symbolizes its Fourier transform. Finally $\mathcal{F}[\cdot]$ is the discrete Fourier transform operator, which can be expressed as a unitary matrix.

$$\tilde{A} \triangleq \mathcal{F}[A] \quad (5.2)$$

Equation (5.2) is an inspiration for numerous GSP schemes [LYLW24; CCZ24a; WDGC24; WRR24] including graph clustering and graph filtering. Various implementations have been greatly facilitated by highly optimized versions of the Fast Fourier Transform (FFT) like the FFT West (FFTW). The discrete Fourier transform operator is the $n \times n$ unitary matrix shown below in equation (5.3).

$$F_n \triangleq \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_0 & \omega_0^2 & \omega_0^3 & \ddots & \omega_0^{n-1} \\ 1 & \omega_0^2 & \omega_0^4 & \omega_0^6 & \ddots & \omega_0^{2(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_0^{n-1} & \omega_0^{2(n-1)} & \omega_0^{3(n-1)} & \dots & \omega_0^{(n-1)^2} \end{bmatrix} \in \mathbb{C}^{n \times n} \quad (5.3)$$

In equation (5.3) the quantity ω_0 defined as in equation (5.4) below and it has an intrinsic connection to both the unitary circle and to the fluctuations of the signal under study. Under this light, the Fourier transform can be very well seen as the evaluation of a polynomial to the n roots of unity.

$$\omega_0 \triangleq \exp\left(-\frac{2\pi i}{n}\right) \in \mathbb{C} \quad (5.4)$$

Moreover, equation (5.5) is crucial for the isometry property of the Fourier transform as well as for the computational efficiency of the FFT. This can be directly attributed to symmetries in the unitary circle as well as to the formulæ for the sine and cosine of the sum of as well as of the difference between any two angles in the primary interval Θ_0 . These are not independent as the shape of the circle confines the degrees of freedom of sine and cosine. Therefore, geometry plays a role here as well similarly to chapter 4.

$$\omega_0^{pq} = \omega_0^{pq \bmod n}, \quad p, q \in \mathbb{Z} \quad (5.5)$$

The second relationship connects the Laplace transform of a matrix M to its exponential as shown in equation (5.6). This is a more general relationship in the sense that M does not have to be a graph adjacency matrix [YHL⁺24; Duo24]. The exponential matrix of a graph plays a crucial role in estimating its structural resilience since the Estrada index of a graph is defined as the trace of its exponential adjacency matrix. Equation (5.6) is also vital for evolving graphs as well as for control systems based on graphs.

$$\mathcal{L}[e^{Mt}] = (sI - M)^{-1}, \quad t \in \mathbb{R}, s \in \mathbb{C} \quad (5.6)$$

In the general case and unlike the Fourier transform, inverting the Laplace transform is not always analytically or computationally feasible. For the continuous case a plethora of numerical schemes has been developed over time capturing special cases or relying on diverse assumptions. This is indicative of the difficulty of the problem. However, as equation (5.6) makes clear, this is not an issue in a GSP context.

Recall that the exponential of a matrix M scaled with a parameter t is defined as a power series as in (5.7) and has a number of important properties such as having only strictly positive eigenvalues, a multiplicative property, commutation, and invertibility. These have major implications for control theory among others.

$$e^{Mt} \triangleq \sum_{k=0}^{+\infty} \frac{t^k}{k!} M^k = \lim_{k \rightarrow \infty} \left(I + \frac{Mt}{k} \right)^k \quad (5.7)$$

GSP can be implemented in a myriad of ways. A recent one which aims at computing efficiently estimates of a global graph state is the class of GNNs. GNNs are distributed GSP architectures relying on a coupling of unsupervised neural network topology with attention mechanisms. The latter allow different neurons to specialize on different parts and different attributes of the input graph, which consistently yields better results. Each vertex in a GNN is a processing node which starts only with a local ground truth state vector extracted from information available at the beginning of the computation and iteratively a global status is derived based on the fundamental fact that graphs contain inherently higher order information in a distributed manner as explained in chapter 1. The resulting graph global state can be subsequently further processed in order to derive global properties such as community structure.

The research objective statement 5.1 outlines the topics, scope, and methodologies shown in this chapter.

Objective 5.1 (Chapter research objectives) *This chapter presents a GNN for discovering affective coherent communities in Twitter graphs based on a model for online behavior. Moreover, a variant thereof for edge fuzzy graphs according to definition 5.1 is given.*

The remainder of this chapter is structured in the following way. In section 5.2 the recent scientific literature regarding behavioral sciences, GNN, and graph signal processing (GSP) is examined. The generic properties of GNNs including possible attribute types are described in 5.3. Then in section 5.4 a social media application is given in the form of assessing the affective coherency of two Twitter graphs structural, functional, and behavioral features. This is extended in section 5.5 where the same architecture is modified for edge fuzzy graphs. Finally, for the general conclusions of this dissertation please chapter 6.

5.2 Previous work

Graph mining aims at extracting knowledge from connection oriented data [GMK24; CZPQ24]. To this end schemas for the latter have been proposed with JSON-LD¹ being a well known one [VRC23]. JSON-LD plays a similar role to that JSON² does for text data [ABC⁺24] and BSON³ for binary data [SPS24]. Graph mining [KA24] is an integral part of the interconnected era since it lays the groundwork for numerous applications across a wide array of financial and technological fields including among others SNA [CCC⁺24], database query optimization [HCC⁺24], supply chain and logistics networks [LWL⁺24], and brain circuit analysis [LTP⁺24]. Recently DL has been added to graph mining, for instance with long short term memory (LSTM) networks [ZNN24]. In this context modeling a graph in terms of vertices, connectivity patterns, and associated features is tantamount to data model selection [FYG⁺24]. Fuzzy graphs extend classical graphs as probabilities drawn from a single distribution which may well have unknown parameters to

¹<https://json-ld.org>

²<https://www.json.org>

³<https://bsonspec.org>

be estimated [XLL⁺24]. Said distribution is closely linked to the semantics and functional nature of the underlying graph: For instance, in a transportation network edge existence probabilities can show how likely a specific road is to be blocked [FuRM24], in a computer network they may model the chance of a virus being propagated along a given link [LLH24], or in long supply chains the knowledge the flows and capacities in light of the *max-flow-min-cut* theorem [PZ24]. A recent survey about the field is [PZCL24]. Finally, community discovery is the holy grail of graph mining as can be seen in figure 5.2.

In a GSP context adjacency matrices are considered as two-dimensional signals [LMM⁺23; OFK⁺18] and signal processing techniques are then employed to extract patterns of interest [MSMR19]. An overview of the connections to deep learning are given in [CSW⁺20]. Flow based GSP is examined in [SS18]. The basic operations of GSP such as shifting and sampling are defined in [GZ17]. A graph version of the LMS adaptive filtering algorithm is presented in [HNR⁺18]. Graph filter design principles are given in [IGSS24]. Additionally graph low pass filters are discussed in [ZHW24]. Blind graph matching can be done with filtering graph signals [LSW24] and so can manifold graph signal processing [CCZ24b]. Recursive graph signal processing is explored in [SZK24]. An overview of the field is [JFG⁺24].

GNNs are connectionist models designed to operate primarily on irregular domains expressing relationships [TNZ⁺24]. They can efficiently learn global properties by iteratively updating local state vectors assigned to vertices or edges [SLG⁺24]. Compared to other neural network architectures they tend to exhibit scalability and robust generalization of local states [ZZC⁺24]. Also, GNNs exhibit scalability by taking into consideration the link oriented nature of social graphs [RCDG⁺24]. Heterogeneous GNN architectures are examined in [ZSH⁺19] and representative GNNs designed to complete versatile tasks in [YLG⁺20]. Edge labeling is proposed in [KKKY19] in the context of few-shot learning for GNNs. GNNs counting graph substructures are examined in [CCVB20]. Reinforcement learning for GNNs is examined in [KFL⁺24]. The technique of aggregated neural path in conjunction with ML tasks is described in [FZMK20]. In [FML⁺19] are given guidelines for social recommendation based on GNNs. Higher order GNNs are introduced in [MRF⁺19], capsule GNNs have been proposed in [XC18], and few shot learning scenarios in [KKKY19]. Applications include social recommendation [FML⁺19] and graph contrasting coding for learning graph representations [QCD⁺20]. Fairness in GNN operations is discussed in [JHF⁺24]. Finally, two recent comprehensive reviews on the field are [Z⁺18] and [WPC⁺20].

Social networks arguably are nowadays critical in shaping public sentiment about various social, political, cultural [DKG24], or even historical events [KSSY20], while reflecting or even distorting it at a global scale [SK⁺24]. Twitter encourages short replies and high interaction between its accounts through features such as replies, mentions, and hashtags [HS24] and has been steadily playing a central role in elections worldwide [Pac24]. A prime example is the 2016 US Republican (or GOP)⁴ party elections and later that year the US presidential elections [Cor17; Ott17]. One important DL task when mining Twitter is assigning sentiment polarity labels to accounts [MSP24]. This is done through local ground truth state vectors containing structural, functional, and affective attributes [KS24]. Another such task is mining hashtags because of their semantic significance [XZ20]. Emotions have been long identified as key factors in social network ecosystems [MSR⁺24] and determining them yields important insights [PEP⁺24]. This appears to be especially true in Twitter [MSS⁺24] where a broad spectrum of affective factors [MKS⁺24] including irony, inflammatory comments, apophlegmatic retorts, and bots with affective programming have been reported to elicit different emotional responses across accounts [MSFL17].

⁴<https://gop.com>

Behavioral attributes have recently emerged as an integral part of many recent computational systems [YLZ⁺24; CM19] including games [WLD⁺24], social media [LHE24], and recently AI systems based on large language models (LLMs) [SA24]. The connection between behavioral systems and data driven analysis is explored in [MD21], while affective gamification strategies are explored in [KSLB18]. For social networks the overall sentiment polarity has been proposed as an important coherence factor [MT24]. To this end affective models like the *emotion wheel* [WMWM20] or the *universal emotion theory* [Ekm99] are invaluable for developing efficient knowledge mining schemes [FCF⁺24]. Non linear composition composing of direction and intensity of emotions is the epicenter of [DWS⁺24]. Emotions serve specific evolutionary roles [ASS24] and have been progressively adapted to describe individual online behavior in various contexts [DG24]. An overview of behavioral systems is given in [WK17; CM19].

5.3 Introductory material

Instead of describing a generic GNN architecture as it was done for TSNs back in chapter 3, the architecture tailored for each task will be presented. The reason behind this is that GNNs are highly customized for the task at hand and, therefore, it makes sense only to describe generic properties such as local information aggregation, propagation properties, topological patterns influencing the emergence of global graph properties and the velocity of this process, or convergence criteria.

Table 5.1: Common GNN attribute types.

Type	Description	Data types
Structural	Pertaining directly to graph topology and geometry	Vectors, matrices
Functional	Define what sort of activities can take place on the graph	Vectors
Spatial	Stem from geolocation and geospatial data	Maps
Temporal	Evolve over time whether deterministically or stochastically	Time series
Affective	Come from or can potentially determine emotional state	Hashtags, speech

The most well known data types processed with GNNs are shown in the rows of table 5.1. Note that these types may well be related as various connections may arise depending on the underlying learning task as well as on the GNN configuration. Learning in GNNs proceeds in a distributed and bottom-up manner. The only available information at the beginning is a ground truth state vector of finite length which is formulated by local conditions and potentially the result of DM algorithms. Depending on the GNN geometry, the learning task, and *ad hoc* factors, state vectors can be stored at the following places:

- ♣ **Vertices:** In this case the local ground truth vectors is stored at the vertices. This is a natural choice after an extended entity recognition as the latter are mapped to vertices under most such algorithms.
- ♣ **Edges:** This is the dual case of the previous one. State vectors are formulated by conditions holding at both end vertices as a weighted mix thereof or as a single long vector representing both vertices.
- ♣ **Both:** This is the most general case and a combination of the previous two. More complex update schemes are necessary as changes can propagate in larger segments depending on the topology.

Since the GNN training process is iterative, convergence monitoring is of the utmost importance. To this end one of the mathematical cornerstone tools pertaining to the selection of the activation function $\varphi(\cdot)$ is theorem 5.1, namely the Brouwer fixed point theorem. When its conditions apply, a locally stationary point x^* can be found. This can be used in the design of convergence criteria and its velocity.

Theorem 5.1 (Brouwer fixed point theorem) *If a mapping $f: I \rightarrow I$ is continuous and the interval I is closed, then there is at least one $x^* \in I$ such that equation (5.8) holds.*

$$x^* = f(x^*) \quad (5.8)$$

In light of theorem 5.1 the result of the activation function can be thought of as the last state of a graph control system. Continuing this line of reasoning, the set of local state vectors has a trajectory in a state space which is terminated with appropriately designed convergence criteria such as the maximum relative steady state normed difference taken over all vertices or the average one if more slack is to be given to the GNN. Whenever labeled vertices are used, as in the next sections, then the number of labels changed between two successive iterations can be used. This trajectory moves a GNN progressively in each iteration from its initial state to a robust global ones. If the GNN state is to be estimated, it is to be done so in a way such that it is consistent with convergence metrics. Regarding ground truth, it does not have to factor in every single bit of information to lead to a steady state or to one out of the possible set of steady states, which is also possible, but it has to include critical knowledge to ensure convergence.

5.4 Twitter affective coherent community discovery with GNNs

5.4.1 GNN architecture

The state vector s_k associated with each vertex $v_k \in V$ has the structure shown in equation (5.9). Specifically, its first two components are structural. They are followed by five functional and three more affective features. By the very definition of these attributes, they are unsigned. Also they were selected based on recent scientific literature recommendations and their meaning is explained in table 5.2.

$$s_k \triangleq [r_{1,k} \quad r_{2,k} \quad f_{1,k} \quad \dots \quad f_{5,k} \quad a_{1,k} \quad a_{2,k} \quad a_{3,k}]^T \quad (5.9)$$

The GNN described in this section finds a label $\hat{l}_k$ for each vertex v_k pertaining to its emotional polarity, which estimates the unknown corresponding true polarity label l_k . To derive the estimated labels the local information as well as the GNN topology with the particular connectivity patterns will be used. The aggregation takes into consideration state vectors from the neighboring vertices.

Table 5.2: Vertex state attributes.

Attribute	Meaning	Attribute	Meaning
$r_{1,k}$	Number of <i>follow</i> relationships	$f_{1,k}$	Number of tweets
$r_{2,k}$	Number of inverse <i>follow</i> relationships	$f_{2,k}$	Number of retweets
$a_{1,k}$	Number of positive hashtags	$f_{3,k}$	Number of hashtags
$a_{2,k}$	Number of negative hashtags	$f_{4,k}$	Number of mentions
$a_{3,k}$	Number of neutral hashtags	$f_{5,k}$	Number of replies



Figure 5.2: Community discovery is the holy grail of graph mining.

For every vertex its corresponding initial label $l_k^{[0]}$ is numerically determined by equation (5.10). The three individual fractions determining the fractions of the hashtags with positive, neural, and negative emotional polarity. The polarity is computed with natural language processing (NLP) as shown in algorithm 1. Thus, these starting vertex labels, which are first order approximations of the true polarity sentiment, will be known as the NLP labels. Computing the affective sign of a hashtag requires determining the sign of the difference between the positive and the negative words of it as shown in algorithm 1.

$$l_k^{[0]} \triangleq \begin{cases} \frac{a_{1,k}}{a_{1,k} + a_{2,k}} & a_{1,k} > a_{2,k} \\ -\frac{a_{2,k}}{a_{1,k} + a_{2,k}} & a_{2,k} > a_{1,k} \\ 0 & a_{1,k} = a_{2,k} \end{cases} \in I_0 \quad (5.10)$$

During the j -th iteration labels are updated as in (5.11). Therein aggregation takes place over every neighboring vertex v_i of v_k . The index $*$ in the activation function argument signifies that if a label has been updated, then the most recent label is used. The local label receives as much weight as that of entire combined neighborhood, preserving stability locally. Also neighborhoods must be substantially large and sufficiently polarized to influence the outcome during only a single iteration. Algorithm 2 summarizes the operation of the GNN on a very high level with references to the appropriate equations.

$$l_k^{[j]} = \varphi \left(\frac{\beta_0}{2} \left(l_k^{[j-1]} + \sum_i \gamma_i \delta_{i,k} l_i^{[*]} \right) \right) \quad (5.11)$$

In equation (5.11) β_0 is a hyperparamter calibrating the activation function argument by expanding its domain. Additionally, γ_i is the followers-to-all ratio as shown in (5.12) and $\delta_{i,k}$ computes online influence

Algorithm 1 Assigning affective polarity to hashtags

Require: Hashtag h_0 coded as a case-sensitive string**Ensure:** Sentiment polarity of h_0 is computed

```

1: set counter  $c_0$  to zero
2: for all words  $w$  of  $h_0$  do
3:   if  $w$  is positive according to SentimentAnalysis then
4:     add one to  $c_0$ 
5:   else if  $w$  is negative then
6:     subtract one from  $c_0$ 
7:   end if
8: end for
9: return  $\text{sign}(c_0)$ 

```

and acts as a vertex weight. The first fraction denotes the fraction of the original content v_i generates by tweeting to the total content it forwards to its connections by tweeting and retweeting. Along a similar line of reasoning, the second fraction measures how many replies are specifically directed to v_i . The third fraction connects v_i with v_k through pairwise hashtag coherency regardless of affective polarity.

$$\gamma_i \triangleq \frac{r_{2,i}}{r_{1,i} + r_{2,i}} \quad \text{and} \quad \delta_{i,k} \triangleq \frac{1}{3} \left(\frac{f_{1,i}}{f_{1,i} + f_{2,i}} + \frac{f_{4,i}}{f_{4,i} + f_{5,i}} + \frac{f_{3,i}}{f_{3,i} + f_{3,k}} \right) \quad (5.12)$$

Algorithm 2 The proposed GNN

Require: State vectors as in equation (5.9)**Require:** Label change threshold η_1 **Ensure:** Robust affective polarity labels $\{l_k\}$

```

1: obtain initial labels  $\{l_k^{[0]}\}$  from algorithm 1
2: repeat
3:   for all vertices  $v_k$  do
4:     update local label  $l_k$  as in equation (5.11)
5:   end for
6: until the relative steady state difference drops under  $\eta_1$ 
7: return labels  $\{l_k\}$ 

```

5.4.2 Results

There are two benchmark graphs shown in 5.3, one sampled using hashtags about a time of festivity and another about a very polarized and heated event. The year 2021 marks the two hundred year milestone from the Greek Revolution of Independence of 1821. Although many celebrations were canceled because of the COVID-19 pandemic, Twitter activity about the anniversary soared with mostly positive comments and references to historical events along some complaints mainly about the cost of the events or the fact many of them did not take place. On the other hand, the 2020 US Presidential Elections took place in a rather charged and polarized political climate among accusations about fake news, inflammatory hashtags, and acerbic comments with Twitter playing a central role in the respective campaigns of both parties.

Table 5.3: Benchmark graph synopsis.

Property	1821	US2020
Number of vertices $ V_h $	132.317	147.881
Number of edges $ E_h $	2.225.177	2.447.224
Density ρ_h / Log-density ρ'_h	16.8170 / 1.2393	16.5486 / 1.2357
Completeness ξ_h / Log-completeness ξ'_h	$2.54E^{-4}$ / 0.6196	$2.38E^{-4}$ / 0.6173
Number of triangles T_h	446.513	489.773
Number of squares S_h	215.387	218.633
Number of cliques of size four C_h	102.044	125.806
Graph diameter L_h	10	11
Percentage of vertices reachable at $L_h - 1$	95.33%	98.17%
Percentage of vertices reachable at $L_h - 2$	93.26%	96.44%
Percentage of vertices reachable at $L_h - 3$	89.11%	91.22%
Percentage of vertices reachable at $L_h - 4$	84.73%	87.47%
Number of favorites F_h	36.994.815	42.114.509
Number of tweets W_h	17.465.844	22.773.674

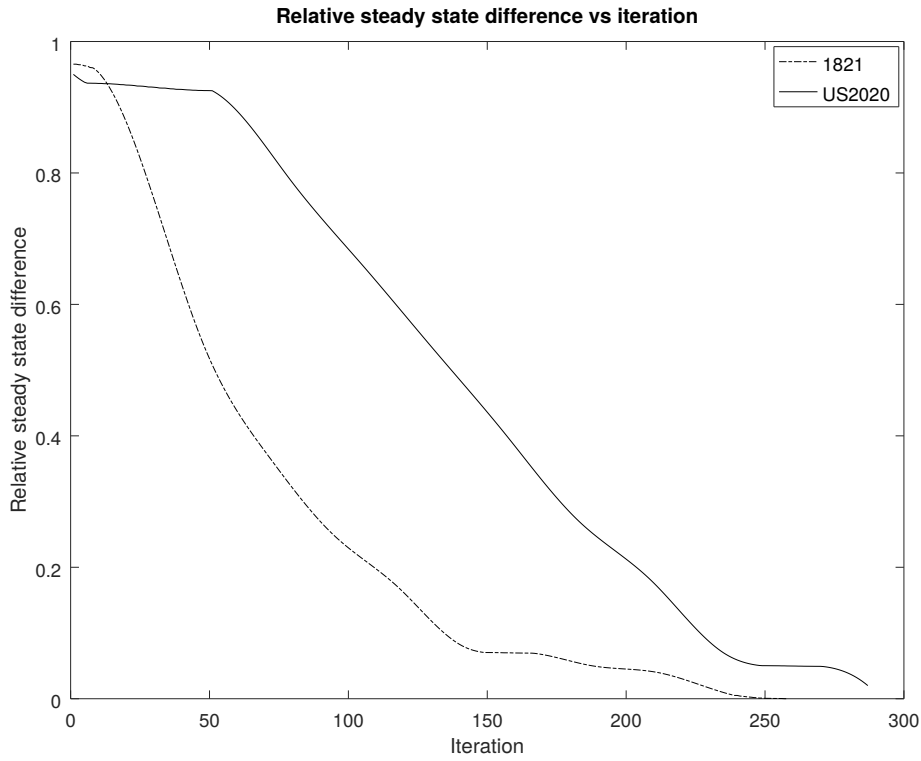


Figure 5.3: Relative steady state difference vs iteration.

In table 5.3 the graph density ρ_h is defined as the ratio of the number of the edges to the number of vertices. Because of the Euler graph formula, density is also an approximation of the average graph degree as shown in equation (5.13). The log-density ρ'_h is similarly defined as the ratio of the respective order of magnitudes of the number of edges to the number of vertices. Observe that in (5.13) the selection of the base of the logarithm is of no consequence because of the base change property of the logarithm.

$$\rho_h \triangleq \frac{|E_h|}{|V_h|} = \frac{2}{|V_h|} \sum_{v_k \in V_h} \deg(v_k) \quad \text{and} \quad \rho'_h \triangleq \frac{\ln |E_h|}{\ln |V_h|} \quad \text{with} \quad \log_b a = \frac{\log_c b}{\log_c a}, \quad abc \neq 0 \quad (5.13)$$

The completeness ξ_h of a graph is defined as the ratio of the edges of a graph to the edges of a fully connected graph with the same number of vertices as shown in equation (5.14). Also the log-completeness ξ'_h is defined in a similar fashion as the order of magnitude of the edges of these two graphs.

$$\xi_h \triangleq \frac{|E_h|}{\binom{|V_h|}{2}} = \frac{2|E_h|}{|V_h|(|V_h| - 1)} \approx \frac{2\rho_h}{|V_h|} \quad \text{and} \quad \xi'_h \triangleq \frac{\ln |E_h|}{\ln \binom{|V_h|}{2}} = \frac{\ln |E|}{\ln |V_h| + \ln (|V_h| - 1) - \ln 2} \approx \frac{\rho'_h}{2} \quad (5.14)$$

Figure 5.3 shows the GNN convergence rate for the two benchmark graphs as measured by the relative steady state difference, namely the fraction of the labels changed in each iteration. The power law and the cutoff power law of table 3.1 were fitted to these curves. The exponent and the normalized MSE are shown. It can be seen that for the 1821 graph convergence is quicker. On the contrary, convergence is slow for the US2020 graph after an initial stage of little progress. The exponents γ_0 indicate a relatively graceful reduction. Moreover, the power law yields higher values as the LS fitting yields a straight line with a steeper curve. the fractional exponents indicate fractal patterns in the distribution of hashtag frequencies such as self-similarity and heavy tail. Also, for both benchmarks the cutoff power law yields lower error.

Table 5.4: Results for the LS solutions.

Graph	Model	γ_0	MSE	Graph	Model	γ_0	MSE
1821	Power law	2.6511	1.1682	US2020	Power law	2.8617	1.1895
1821	Cutoff	2.488	1	US2020	Cutoff	2.6322	1.0184

The fraction of labels flipping sign after the GNN training in the benchmark graphs is shown in table 5.5. For the 1821 graph the NLP labels of algorithm 1 were fairly accurate, but this not the case for the US2020 graph as a considerable number of accounts flipped sign. This can be attributed to the overall positive atmosphere of the 1821 graph, whereas US2020 graph echoes a toxic and polarized political climate. This can also be seen in table 5.6 where the total sentiment polarity for the benchmark graphs is shown.

Table 5.5: Fractions for sign flips of 1821 (left) and US2020 (right) graphs.

	Positive	Negative	Positive	Negative
Positive-GNN	67.33%	32.67%	64.28%	33.72%
Negative-GNN	49.11%	50.89%	45.15%	54.85%

Table 5.6: Social graph sentiment analysis.

	Positive	Negative
1821	75.92%	24.08%
US2020	62.18%	37.82%

Given the above, the difference between algorithm 1 and the GNN can perhaps be explained in the following way in a wider context. Twitter conversations, especially the charged ones, often contain a significant number of ironic tweets which by no means are to be taken literally as their true meaning hides in the subtext. Thus, processing tweets with NLP, which focus on syntax alone, can lead to confusion in many cases. On the contrary, GNNs by aggregating local information yield a more accurate result.

5.5 GNN for edge fuzzy Twitter graphs

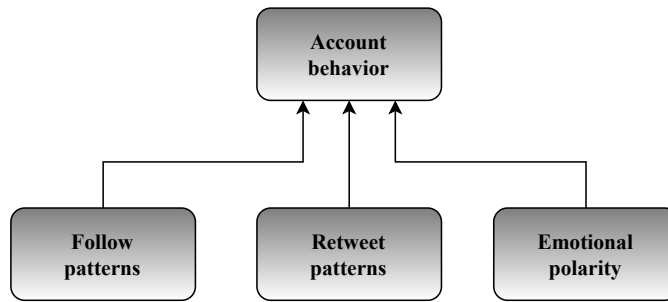


Figure 5.4: The proposed account behavior model.

In this section the emotional coherency of a social graph is extended by taking into consideration that the communication between any two accounts may well reveal information about them. Thus, the edges of a social graph seldom are a neutral conduit of information, but instead they get their own characteristics over time which can also be mined. In order to describe the proposed architecture some concepts need to be defined. First the class of edge fuzzy graphs is introduced in definition 5.1.

Definition 5.1 (Edge fuzzy graph) *An edge fuzzy graph is the ordered triplet shown in equation (5.15).*

$$G \triangleq (V, E, h) \quad (5.15)$$

The elements in (5.15) have the following meaning:

- ♣ The vertex set V . In this context each vertex corresponds to a single Twitter account through a bijection.
- ♣ The fuzzy edge set $E \subseteq V \times V$. The connectivity patterns therein reflect the social graph dynamics.
- ♣ The functional $h : E \rightarrow I_0$ maps each edge to a probability drawn from a single distribution.

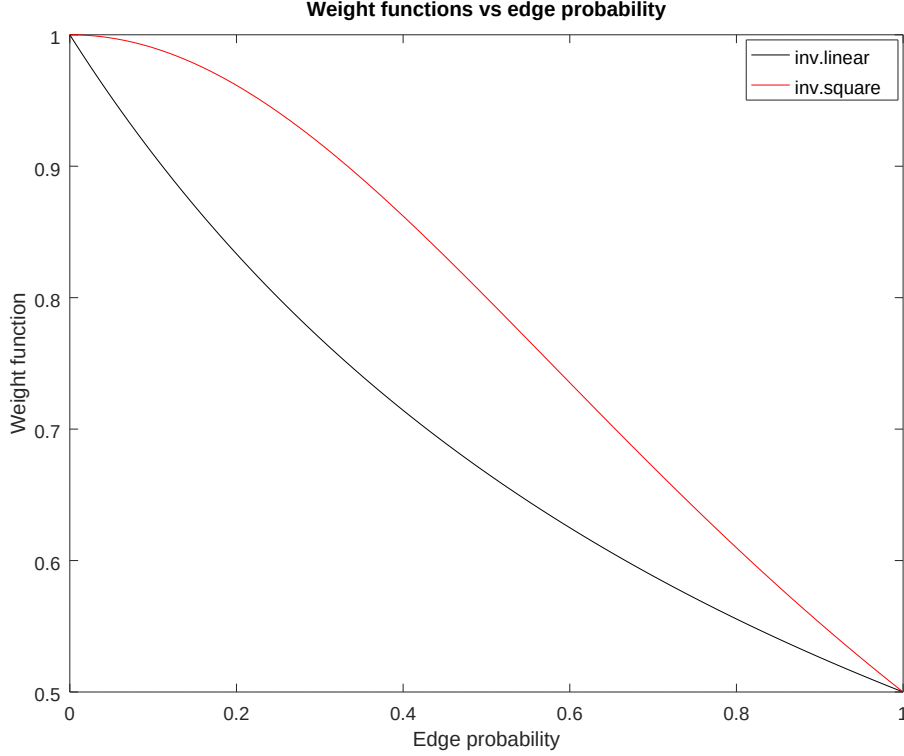


Figure 5.5: Weight functions.

The proposed GNN architecture models the communication strength between accounts as the partial uncertainty that the edge connecting them belongs to the graph, expressed through a probability. The latter depends on an attribute set and hence it is local, but in the general case it may be a function of the features of neighboring vertices as well. The model capturing the nature of an edge is shown in figure 5.4. Moreover, the probability $p_{i,j}$ for edge $e_{i,j}$ between vertices v_i and v_j is computed as in equation (5.16). Therein three factors are taken into consideration:

- ♣ A directed *follow* link from the i -th account to the j -th is indicated by the binary indicator $F_{i,j}$.
- ♣ The ratio of retweets number $r_{i,j}$ of the i -th account coming from the j -th to the total retweets R .
- ♣ The correlation factor $c_{i,j}$ of the emotional coherency of the hashtags of the i -th and j -th accounts.

$$\text{prob} \{ \text{edge} (v_i, v_j) \} \triangleq p_{i,j} = w_f F_{i,j} + w_r \frac{r_{i,j}}{R} + w_c c_{i,j} = \underbrace{\begin{bmatrix} w_f & \frac{w_r}{R} & w_c \end{bmatrix}}_{\mathbf{w}^T} \underbrace{\begin{bmatrix} F_{i,j} \\ r_{i,j} \\ c_{i,j} \end{bmatrix}}_{\mathbf{f}_{i,j}} = \mathbf{w}^T \mathbf{f}_{i,j} \quad (5.16)$$

The weights of the linear combination in (5.16) encode the sign and relative strength of each factor contribution, namely how much each factor participates to the edge probability existence and whether such participation reinforces or weakens said probability respectively. Moreover, they ensure the numerical stability of $p_{i,j}$. Equation (5.16) is a linear estimator of the true probability that the respective edge exists and it is progressively refined with each round. The coefficients w_f , w_r , and w_c act as weights of the relative contribution of each term. Although they can be adaptive depending on the messages exchanged between the i -th and j -th Twitter accounts, the elements of weight vector w received numerical values according to the semantic strength of the respective factor. Therefore, the least value is given to w_c as the emotional coherency is the weakest link between any pair of accounts, especially since it may be updated. Then w_r gets the next bigger value, as frequent retweets imply a somewhat coherent and strong connection. Finally, the largest weight is assigned to the case when there is a *follow* as it is permanent until revoked.

The sentiment estimation $l^{[r]}$ of the i -th Twitter account during the current iteration r , counted in discrete rounds in the context of this dissertation, is computed by the elements of the vector of equation (5.17). Therein the aforementioned vector consists of a triplet of emotional polarity scores. Its elements evolve from the corresponding starting ground truth vector obtained from algorithm 1.

$$l_i^{[r]} \triangleq \begin{bmatrix} n_{p,i}^{[r]} & n_{n,i}^{[r]} & n_{g,i}^{[r]} \end{bmatrix}^T \quad (5.17)$$

For each edge in the graph between vertices v_i and v_j there is a corresponding weight $\delta_{i,j}$ as shown in equation (5.18). The weight is a function of the edge probability $p_{i,j}$ and typically denotes either the difficulty of a message or flow to cross that edge as in computer or electrical networks or a critical threshold that needs to be reached like in human neurons. In table 5.7 the options used here are shown along with their main numerical and algorithmic properties. Depending on the underlying field, $f(\cdot)$ is chosen to be smooth so that it can be easily approximated by a Taylor expansion of appropriate order and that its derivative can be computed, as it is typically linked to instantaneous transfer in the case of network flow. Also the path length $\text{length}(P)$ of a path P is the sum of the weights of the edges comprising that path.

$$\delta_{i,j} \triangleq f(p_{i,j}) \quad \text{and} \quad \text{length}(p) \triangleq \sum_{\text{edge}(i,j) \in P} \delta_{i,j} \quad \text{with} \quad \text{edge}(v_i, v_j) \triangleq \text{edge}(i, j) \quad (5.18)$$

There are many options for the link function $f(\cdot)$ whose form depends on the underlying task as well as on algorithmic and numerical considerations. One of the most common options is the *inverse*, but it has the disadvantage of being almost singular close to zero, generating thus excessive weight values. An alternative to it is the *inverse linear* function, which may not decay quick enough in some engineering scenarios. The *inverse square* function does that and has a natural interpretation as a potential function, but it suffers from Runge's phenomenon, which is similar to Gibbs phenomenon in Fourier analysis.

A long graph path p in (5.18) can be approximated as in equation (5.19). Depending on the decay rate of the link function, the integral may not converge. However, for the above choices it does. In figure 5.5 the inverse linear and the inverse square are shown for I_0 . It can be inferred they are strictly decreasing and

Table 5.7: Fuzzy function options.

Name	Formula	Properties	Applications
Inverse	$\delta_{i,j} \triangleq 1/p_{i,j}$	Easily integrable Discontinuity in I_0 Resistance function Capacity function	Fuzzy graph databases
Inverse linear	$\delta_{i,j} \triangleq 1/(1 + p_{i,j})$	Easily integrable Discontinuity outside I_0 Bounded capacity function	Fuzzy graph databases
Inverse square	$\delta_{i,j} \triangleq 1/(1 + p_{i,j}^2)$	Easily integrable No discontinuity in $\mathbb{R}$ Prone to Runge's phenomenon Potential function	NLP graphs

everywhere smooth, expressing that the more likely an edge is to belong to the graph, the easier to cross.

$$\text{length}(p) \approx \int_{d_1}^{d_2} f(r) dr = \begin{cases} \ln d_2 - \ln d_1 & \text{inverse} \\ 2 \ln \frac{1 + d_2}{1 + d_1} + (d_2 - d_1) & \text{inverse linear} \\ 2(\arctan(d_2) - \arctan(d_1)) + (d_2 - d_1) & \text{inverse quadratic} \end{cases} \quad (5.19)$$

The label update mechanism of the proposed GNN architecture is given in (5.20) and it is derived from (5.11). The two differences is that the activation function is the hyperbolic tangent of (3.25) as now the argument is signed. Moreover, the fuzzy edges are factored in. The neighbor index j ranges over all inbound neighbors of the i -th vertex. It is a higher order mechanism as the state vector of a vertex is recursively updated. The iteration indicator $*$ will be used as in the preceding section.

$$l^{[t+1]} = \varphi \left(\frac{\beta_0}{2} \left(l^{[t-1]} + \sum_j \frac{\delta_{i,j}}{\Delta} l_j^{[*]} \right) \right) \quad (5.20)$$

5.5.1 Results

Table 5.8 has the normalized number of iterations as a function of the hyperparameter β_0 of (5.20) for the graphs of table 5.3. Said normalization is per graph and per weight function to illustrate the comparative effect of hyperparameter β_0 . Moreover, in order to highlight the role of the emotional attribute $c_{i,j}$ of (5.16) on the convergence rate the same GNN is run with the latter removed from the initial ground truth vectors as well. It can be seen that including $c_{i,j}$ accelerates converge. This can be explained by the the fact that the behavioral factor adds an independent dimension making easier vertex differentiation.

The above factors suggest that variability in the weight space as well as in the vertex profile increase the flexibility of the update mechanism of (5.20). This is in accordance with the standard pattern recognition

Table 5.8: Normalized number of iterations as a function of the hyperparameter β_0 .

Hyperparameter	Graph	Inv.linear	Inv.linear+beh	Inv.square	Inv.square+beh
0.5	1821	1.49	1.26	1.37	1.38
0.7		1.45	1.24	1.32	1.23
0.8		1.39	1.16	1.27	1.15
0.9		1.33	1.08	1.22	1.07
1		1.28	1	1.19	1
0.5	US2020	1.41	1.17	1.33	1.13
0.7		1.42	1.14	1.29	1.11
0.8		1.37	1.09	1.25	1.08
0.9		1.29	1.05	1.21	1.03
1		1.24	1	1.18	1

maxim stating that mapping data to a space of higher dimensionality facilitates their clustering. On the other hand, the curse of dimensionality imposes a limit on how big this new space can get. As both spaces used in this work however are low dimensional, this does not constitute a problem.

Table 5.9: Emotional distributions (pos/neu/neg) computed with the best value of β_0 .

Graph	Init	Inv.linear	Inv.linear+beh	Inv.square	Inv.square+beh
1821	0.64/0.24/0.12	0.68/0.14/0.18	0.73/0.15/0.12	0.69/0.15/0.16	0.74/0.14/0.12
US2020	0.28/0.23/0.49	0.21/0.22/0.57	0.17/0.16/0.67	0.22/0.22/0.56	0.16/0.16/0.68

5.6 Bibliography

- [ABC⁺24] Lyes Attouche, Mohamed-Amine Baazizi, Dario Colazzo, Giorgio Ghelli, Carlo Sartiani, and Stefanie Scherzinger. Validation of modern JSON schema: Formalization and complexity. *Proceedings of the ACM on Programming Languages*, 8(POPL):1451–1481, 2024.
- [ASS24] Laith Al-Shawaf and Todd K Shackelford. *The Oxford handbook of evolution and the emotions*. Oxford University Press, 2024.
- [CCC⁺24] Xinyang Chen, Jiayu Cai, Guoting Chen, Wensheng Gan, and Amaël Broustet. FCSG-Miner: Frequent closed subgraph mining in multi-graphs. *Information Sciences*, 665, 2024.
- [CCVB20] Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures? *Advances in neural information processing systems*, 33:10383–10395, 2020.
- [CCZ24a] Fei Chen, Gene Cheung, and Xue Zhang. Manifold graph signal restoration using gradient graph Laplacian regularizer. *IEEE Transactions on Signal Processing*, 2024.
- [CCZ24b] Fei Chen, Gene Cheung, and Xue Zhang. Manifold graph signal restoration using gradient graph Laplacian regularizer. *IEEE Transactions on Signal Processing*, 2024.
- [CM19] Traci M Cihon and Mark A Mattaini. Emerging cultural and behavioral systems science. *Perspectives on behavior science*, 42(4):699–711, 2019.

- [Cor17] Michael Cornfield. Empowering the party-crasher: Donald J. Trump, the first 2016 GOP presidential debate, and the Twitter marketplace for political campaigns. *Journal of Political Marketing*, 16(3-4):212–243, 2017.
- [CSW⁺20] Mark Cheung, John Shi, Oren Wright, Lavendar Y Jiang, Xujin Liu, and José MF Moura. Graph signal processing and deep learning: Convolution, pooling, and topology. *IEEE Signal Processing Magazine*, 37(6):139–149, 2020.
- [CZPQ24] Xinsheng Chen, Xiaofa Zhou, Jin Pan, and Xin Qiu. Sequential event graph mining in power grid accident tracing based on RED-GNN algorithm. In *Proceedings of the International Conference on Computing, Machine Learning and Data Science*, pages 1–7, 2024.
- [DG24] Marco Del Giudice. Afterword: Evolutionary emotion research at the crossroads. *Genetics*, 2024.
- [DKG24] Sadettin Demirel, Elif Kahraman, and Uğur Gündüz. A text mining analysis of the change in status of the Hagia Sophia on Twitter: The political discourse and its reflections on the public opinion. *Atlantic Journal of Communication*, 32(1):63–90, 2024.
- [Duo24] Javier Duoandikoetxea. *Fourier analysis*, volume 29. American Mathematical Society, 2024.
- [DWS⁺24] Sinuo Deng, Lifang Wu, Ge Shi, Lehao Xing, Meng Jian, Ye Xiang, and Ruihai Dong. Learning to compose diversified prompts for image emotion classification. *Computational Visual Media*, pages 1–15, 2024.
- [Ekm99] Paul Ekman. Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16, 1999.
- [FCF⁺24] Edoardo Fazzari, Fabio Carrara, Fabrizio Falchi, Cesare Stefanini, and Donato Romano. Using AI to decode the behavioral responses of an insect to chemical stimuli: Towards machine-animal computational technologies. *International Journal of Machine Learning and Cybernetics*, 15(5):1985–1994, 2024.
- [FML⁺19] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The WWW conference*, pages 417–426, 2019.
- [FuRM24] Haojun Fang, Ubaid ur Rehman, and Tahir Mahmood. Identification of eco-friendly transportation mode by employing complex intuitionistic fuzzy multi-criteria decision-making approach based on probability aggregation operators. *IEEE Access*, 2024.
- [FYG⁺24] Yajing Fan, Shuyang Yu, Bin Gu, Ziran Xiong, Zhou Zhai, Heng Huang, and Yi Chang. Global model selection for semi-supervised support vector machine via solution paths. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
- [FZMK20] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *WWW*, pages 2331–2341, 2020.
- [GMK24] Francesco Grassi, Giorgio Manganini, and Konstantinos Kouramas. Graph-based data mining, pattern recognition and anomaly detection for intelligent energy networks. *Computers & Industrial Engineering*, 2024.

- [GZ17] Adnan Gavili and Xiao-Ping Zhang. On the shift operator, graph frequency, and optimal filtering in graph signal processing. *IEEE Transactions on Signal Processing*, 65(23):6303–6318, 2017.
- [HCC⁺24] Cheng He, Xinyang Chen, Guoting Chen, Wensheng Gan, and Philippe Fournier-Viger. Mining credible attribute rules in dynamic attributed graphs. *Expert Systems with Applications*, 246, 2024.
- [HLH24] Xueyan Hu, Wei Liu, and Hua Huo. An intelligent network traffic prediction method based on butterworth filter and CNN–LSTM. *Computer Networks*, 240, 2024.
- [HNR⁺18] Fei Hua, Roula Nassif, Cédric Richard, Haiyan Wang, and Ali H Sayed. A preconditioned graph diffusion LMS for adaptive graph signal processing. In *EUSIPCO*, pages 111–115. IEEE, 2018.
- [HS24] Christian Howard-Sukhil. Twitter & world literature: The development of hashtag communities as a global writing practice. *New Techno Humanities*, 2024.
- [IGSS24] Elvin Isufi, Fernando Gama, David I Shuman, and Santiago Segarra. Graph filters for signal processing and machine learning on graphs. *IEEE Transactions on Signal Processing*, 2024.
- [JFG⁺24] Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, et al. A comprehensive survey on deep graph representation learning. *Neural Networks*, 2024.
- [JHF⁺24] Zhimeng Jiang, Xiaotian Han, Chao Fan, Zirui Liu, Na Zou, Ali Mostafavi, and Xia Hu. Chasing fairness in graphs: A GNN architecture perspective. In *AAAI*, volume 38, pages 21214–21222, 2024.
- [KA24] Shima Khoshraftar and Aijun An. A survey on graph representation learning methods. *ACM Transactions on Intelligent Systems and Technology*, 15(1):1–55, 2024.
- [KFL⁺24] Lecheng Kong, Jiarui Feng, Hao Liu, Dacheng Tao, Yixin Chen, and Muhan Zhang. Mag-gnn: Reinforcement learning boosted graph neural network. *Advances in Neural Information Processing Systems*, 36, 2024.
- [KKKY19] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *CVPR*, pages 11–20, 2019.
- [KS24] Mantasha Khan and Ankita Srivastava. A review on sentiment analysis of Twitter data using machine learning techniques. *International Journal of Engineering and Management Research*, 14(1):186–195, 2024.
- [KSLB18] Sangkyun Kim, Kibong Song, Barbara Lockee, and John Burton. What is gamification in learning and education? In *Gamification in learning and education*, pages 25–38. Springer, 2018.
- [KSSY20] Franziska B Keller, David Schoch, Sebastian Stier, and JungHwan Yang. Political astroturfing on Twitter: How to coordinate a disinformation campaign. *Political Communication*, 37(2):256–280, 2020.

- [LCW⁺24] James Liang, Yiming Cui, Qifan Wang, Tong Geng, Wenguan Wang, and Dongfang Liu. ClusterFomer: Clustering as a universal visual learner. *Advances in Neural Information Processing Systems*, 36, 2024.
- [LHE24] Sitan Li, Jeongmin Ham, and Matthew S Eastin. Social media users’ affective, attitudinal, and behavioral responses to virtual human emotions. *Telematics and Informatics*, 87, 2024.
- [LLH24] Xiaohong Li, Yang Li, and Hong He. Computer network virus defense with data mining-based active protection. *Scalable Computing: Practice and Experience*, 25(1):45–54, 2024.
- [LMM⁺23] Geert Leus, Antonio G Marques, José MF Moura, Antonio Ortega, and David I Shuman. Graph signal processing: History, development, impact, and outlook. *IEEE Signal Processing Magazine*, 40:49–60, 2023.
- [LSW24] Hang Liu, Anna Scaglione, and Hoi-To Wai. Blind graph matching using graph signals. *IEEE Transactions on Signal Processing*, 2024.
- [LTP⁺24] Cunbo Li, Tian Tang, Yue Pan, Lei Yang, Shuhan Zhang, Zhaojin Chen, Peiyang Li, Dongrui Gao, Huaifu Chen, Fali Li, et al. An efficient graph learning system for emotion recognition inspired by the cognitive prior graph of EEG brain network. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
- [LWL⁺24] Haojian Liang, Shaohua Wang, Huilai Li, Liang Zhou, Xueyan Zhang, and Shaowen Wang. BiGNN: Bipartite graph neural network with attention mechanism for solving multiple traveling salesman problems in urban logistics. *International Journal of Applied Earth Observation and Geoinformation*, 129, 2024.
- [LXD⁺24] Rui Liu, Pengwei Xing, Zichao Deng, Anran Li, Cuntai Guan, and Han Yu. Federated graph neural networks: Overview, techniques, and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
- [LYLW24] Ruyue Liu, Rong Yin, Yong Liu, and Weiping Wang. ASWT-SGNN: Adaptive spectral wavelet transform-based self-supervised graph neural network. In *AAAI*, volume 38, pages 13990–13998, 2024.
- [MD21] Ivan Markovsky and Florian Dörfler. Behavioral systems theory in data-driven analysis, signal processing, and control. *Annual Reviews in Control*, 2021.
- [MKS⁺24] Md Saef Ullah Miah, Md Mohsin Kabir, Talha Bin Sarwar, Mejdil Safran, Sultan Alfarhood, and MF Mridha. A multimodal approach to cross-lingual sentiment analysis with ensemble of transformer and LLM. *Scientific Reports*, 14(1), 2024.
- [MRF⁺19] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *AAAI*, volume 33, pages 4602–4609, 2019.
- [MSFL17] Bjarke Mønsted, Piotr Sapieżyński, Emilio Ferrara, and Sune Lehmann. Evidence of complex contagion of information in social media: An experiment using Twitter bots. *PLoS one*, 12(9), 2017.

- [MSMR19] Gonzalo Mateos, Santiago Segarra, Antonio G Marques, and Alejandro Ribeiro. Connecting the dots: Identifying network structure via graph signal processing. *IEEE Signal Processing Magazine*, 36(3):16–43, 2019.
- [MSP24] Satvika Marrapu, Willam Senn, and Victor Prybutok. Sentiment analysis of Twitter discourse on Omicron vaccination in the USA using VADER and BERT. *Journal of Data Science and Intelligent Systems*, 2024.
- [MSR⁺24] V Mahalakshmi, P Shenbagavalli, S Raguvaram, V Rajakumareswaran, and E Sivaraman. Twitter sentiment analysis using conditional generative adversarial network. *International Journal of Cognitive Computing in Engineering*, 5:161–169, 2024.
- [MSS⁺24] Astha Modi, Khelan Shah, Shrey Shah, Samir Patel, and Manan Shah. Sentiment analysis of Twitter feeds using flask environment: A superior application of data analysis. *Annals of Data Science*, 11(1):159–180, 2024.
- [MT24] L Meghan Mahoney and Tang Tang. *Strategic social media: From marketing to social change*. John Wiley & Sons, 2024.
- [MZH⁺24] Sugang Ma, Bo Zhao, Zhiqiang Hou, Wangsheng Yu, Lei Pu, and Xiaobao Yang. SOCF: A correlation filter for real-time UAV tracking based on spatial disturbance suppression and object saliency-aware. *Expert Systems with Applications*, 238, 2024.
- [OFK⁺18] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.
- [Ott17] Brian L Ott. The age of Twitter: Donald J. Trump and the politics of debasement. *Critical studies in media communication*, 34(1):59–68, 2017.
- [Pac24] Diogo Pacheco. Bots, elections, and controversies: Twitter insights from Brazil’s polarised elections. In *Web Conference*, pages 2651–2659. ACM, 2024.
- [PEP⁺24] D Paulraj, P Ezhumalai, Mohan Prakash, et al. A deep learning modified neural network (DLMNN) based proficient sentiment analysis technique on Twitter data. *Journal of Experimental & Theoretical Artificial Intelligence*, 36(3):415–434, 2024.
- [PZ24] Adam Polak and Maksym Zub. Learning-augmented maximum flow. *Information Processing Letters*, 2024.
- [PZCL24] Jin Peng, Bo Zhang, Lin Chen, and Hui Li. A survey on uncertain graph and uncertain network optimization. *Fuzzy Optimization and Decision Making*, 23(1):129–153, 2024.
- [QCD⁺20] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *KDD*, pages 1150–1160, 2020.
- [RCDG⁺24] Emanuele Rossi, Bertrand Charpentier, Francesco Di Giovanni, Fabrizio Frasca, Stephan Günnemann, and Michael M Bronstein. Edge directionality improves learning on heterophilic graphs. In *Learning on Graphs Conference*, pages 25–1. PMLR, 2024.

- [SA24] Reiji Suzuki and Takaya Arita. An evolutionary model of personality traits related to cooperative behavior using a large language model. *Scientific Reports*, 14, 2024.
- [SK⁺24] Prithi Samuel, Brahmadesam Viswanathan Krishna, et al. Exploration of sentiment analysis in twitter propaganda: A deep dive. *Multimedia Tools and Applications*, 83(15):44729–44751, 2024.
- [SLG⁺24] Yingxia Shao, Hongzheng Li, Xizhi Gu, Hongbo Yin, Yawen Li, Xupeng Miao, Wentao Zhang, Bin Cui, and Lei Chen. Distributed graph neural network training: A survey. *ACM Computing Surveys*, 56(8):1–39, 2024.
- [SPS24] H Shwetha, D Prajwal, and Swathi Sridharan. From MongoDB to React: Unleashing the power of MERN in e-commerce. In *ICETCS*, pages 1–6. IEEE, 2024.
- [SS18] Michael T Schaub and Santiago Segarra. Flow smoothing and denoising: Graph signal processing in the edge-space. In *GlobalSIP*, pages 735–739. IEEE, 2018.
- [SSYR24] Henry Senior, Gregory Slabaugh, Shanxin Yuan, and Luca Rossi. Graph neural networks in vision-language image understanding: A survey. *The Visual Computer*, pages 1–26, 2024.
- [SZK24] Alireza Naeimi Sadigh, Hadi Zayyani, and Mehdi Korki. A robust proportionate graph recursive least squares algorithm for adaptive graph signal recovery. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2024.
- [TNZ⁺24] Thanh-Nghia Truong, Cuong Tuan Nguyen, Richard Zanibbi, Harold Mouchère, and Masaki Nakagawa. A survey on handwritten mathematical expression recognition: The rise of encoder-decoder and GNN models. *Pattern Recognition*, 2024.
- [TZQ⁺24] Xianlun Tang, Jing Zhang, Yidan Qi, Ke Liu, Rui Li, and Huiming Wang. A spatial filter temporal graph convolutional network for decoding motor imagery EEG signals. *Expert Systems with Applications*, 238, 2024.
- [VRC23] Arthur Vercruysse, Julián Andrés Rojas, and Pieter Colpaert. Bringing IDE support to JSON-LD with the language server protocol. In *ISWC (Posters/Demos/Industry)*, 2023.
- [WDGC24] Simon Weber, Thomas Dagès, Maolin Gao, and Daniel Cremers. Finsler-Laplace-Beltrami operators with application to shape analysis. In *CVPR*, pages 3131–3140. IEEE/CVF, 2024.
- [WK17] Nick Wilkinson and Matthias Klaes. *An introduction to behavioral economics*. Macmillan International Higher Education, 2017.
- [WLD⁺24] Karen D Wang, Haoyu Liu, David DeLiema, Nick Haber, and Shima Salehi. Discovering players’ problem-solving behavioral characteristics in a puzzle game through sequence mining. In *Learning Analytics and Knowledge Conference*, pages 498–506. ACM, 2024.
- [WMWM20] Shuo Wang, Aishan Maolinyazi, Xinle Wu, and Xiaofeng Meng. Emo2Vec: Learning emotional embeddings via multi-emotion category. *ACM Transactions on Internet Technology*, 20(2):1–17, 2020.
- [WPC⁺20] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

- [WRR24] Zhiyang Wang, Luana Ruiz, and Alejandro Ribeiro. Stability to deformations of manifold filters and manifold neural networks. *IEEE Transactions on Signal Processing*, 2024.
- [XC18] Zhang Xinyi and Lihui Chen. Capsule graph neural network. In *International conference on learning representations*, 2018.
- [XLL⁺24] Jindong Xu, Kang Li, Ziyi Li, Qianpeng Chong, Haihua Xing, Qianguo Xing, and Mengying Ni. Fuzzy graph convolutional network for hyperspectral image classification. *Engineering Applications of Artificial Intelligence*, 127, 2024.
- [XZ20] Sifan Xu and Alvin Zhou. Hashtag homophily in Twitter network: Examining a controversial cause-related marketing campaign. *Computers in Human Behavior*, 102:87–96, 2020.
- [YHL⁺24] Hu Yu, Jie Huang, Lingzhi Li, Feng Zhao, et al. Deep fractional Fourier transform. *Advances in Neural Information Processing Systems*, 36, 2024.
- [YLG⁺20] Changqian Yu, Yifan Liu, Changxin Gao, Chunhua Shen, and Nong Sang. Representative graph neural network. In *ECCV*, pages 379–396. Springer, 2020.
- [YLZ⁺24] Shaokai Ye, Jessy Lauer, Mu Zhou, Alexander Mathis, and Mackenzie Mathis. AmadeusGPT: A natural language interface for interactive animal behavioral analysis. *Advances in Neural Information Processing Systems*, 36, 2024.
- [Z⁺18] Jie Zhou et al. Graph neural networks: A review of methods and applications, 2018.
- [ZHW24] Chenyue Zhang, Yiran He, and Hoi-To Wai. Detecting low pass graph signals via spectral pattern: Sampling complexity and applications. *IEEE Transactions on Signal Processing*, 2024.
- [ZNN24] Mohammad Reza Zakerzadeh, Seyedkeivan Naseri, and Payam Naseri. Modelling hysteresis in shape memory alloys using LSTM recurrent neural network. *Journal of Applied Mathematics*, 2024, 2024.
- [ZSH⁺19] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. Heterogeneous graph neural network. In *ICDM*, pages 793–803, 2019.
- [ZZC⁺24] Xin Zheng, Miao Zhang, Chunyang Chen, Soheila Molaei, Chuan Zhou, and Shirui Pan. GnnEvaluator: Evaluating gnn performance on unseen graphs without labels. *Advances in Neural Information Processing Systems*, 36, 2024.

Chapter 6

Conclusions

When a rule is extremely complex, that which conforms to it passes for irregular (random)
Gottfried Wilhelm Leibniz (1646 – 1716)
“Discourse on metaphysics” (1686)

6.1 Chapter introduction

THIS chapter is not of technical nature in a strict sense since neither algorithms nor results are presented. Instead, the purpose of this chapter, as given in objective statement 6.1 below, is to provide a general overview of the work presented in the preceding three chapters, namely chapters 3, 4, and 5 which constitute the very heart of this dissertation.

Objective 6.1 (Chapter objectives) *The twofold objective of this chapter is the following:*

- ♣ *The conclusions of this dissertation are collected, categorized, presented, and extensively commented on.*
- ♣ *The research papers written by the author as well as the distinctions earned are shown.*

In this way the reader can better understand the topics of this work from a much broader and wider perspective, especially when it comes to topics requiring algorithmic sophistication, cutting edge programming with advanced software tools, or both as such skills are necessary for higher order knowledge mining. Additionally, inspiration for future research directions may well come from the material presented here. Finally, interested undergraduate, masters students, or people seeking a second career^{1 2 3 4 5} may get a better understanding of the requirements in this field.

One way to characterize the neural network architectures presented in this dissertation is the taxonomy given in table 6.1, which is by no means unique. This shows the diversity of the architectures as well as their potential for further evolution and for being tailored to a broad spectrum of computational tasks.

Table 6.1: Architecture taxonomy.

Architecture	Properties
Tensor stack networks	Supervised learning Regular grid Centralized learning Natively handling tensor data
Self organizing maps	Unsupervised learning Regular grid Distributed learning Tensor metrics required for tensor data
Graph neural networks	Unsupervised learning Irregular grid Distributed learning Tensor metrics required for tensor data

The remainder of this chapter is structured as explained below. In section 6.2 general references about related research topics are given. Said papers were written by the author. Then, in section 6.3 the general conclusions regarding higher order methodologies for mining humanistic data can be drawn based on the technical material presented in the previous chapters. Therein emphasis is placed on the directions of *modelling power* or *expressive power* as described in subsection 6.3.1, computational complexity as discussed in subsection 6.3.2, space complexity as explained in subsection 6.3.3, and digital awareness which is the focal point of subsection 6.3.4. The research contributions of the author as well as the figures of merit regarding this work are given in more detail in section 6.4. Finally, the prospects and the potential of the higher order methods in the near future are given in section 6.5. The preceding description of the structure of this chapter underline its special nature and function.

¹<https://www.theguardian.com/money/2012/apr/20/rise-of-second-job>

²<https://www.forbes.com/sites/tracybrower/2024/03/12/best-jobs-to-consider-when-making-a-career-change>

³<https://www.monster.com/career-advice/article/how-second-careers-start-hot-jobs>

⁴<https://www.weforum.org/agenda/2023/05/workers-multiple-careers-jobs-skills>

⁵<https://hbr.org/1983/a-second-career-the-possible-dream>

6.2 Previous work

The bibliography regarding the topics covered in this dissertation has been given already in the respective bibliographical sections of the previous chapters. The purpose of this section is to briefly describe research papers by the author which have not been covered in the previous chapters, but they are relevant nonetheless. Their scope covers the entire category spectrum of academic literature and ranges from high level surveys to articles and conference papers, their common point being higher order methodologies. Please note that the papers presented in this subsection represent research milestones, excluding the papers the chapters of this dissertation are based on as seen on table 6.2. For an evaluation of the relevancy of the topic of every paper by the author to this work please see section 6.4.

Table 6.2: Research papers forming the basis for each chapter.

Chapter	Papers
Chapter 3	[DM20] and [DKMI21]
Chapter 4	[DGMS20b]
Chapter 5	[DGMS20a] and [DKMS21a]

Graph mining is one of the foundations of this dissertation. As such, many of the research papers presented here pertain to it. Chronologically the first work is [KD14] which presents a space efficient data structure for evolving graphs supporting vertex and edge attributes based on splay trees. The next research milestone in this direction is [DBM15] where higher order graph centrality metrics have been implemented in Java whereas the graph was stored in Neo4j. Then in [DKT16b] system theory concepts such as polynomial decay has been applied to evaluate graph centrality metrics and a polynomial model has been fitted in the LS sense to centrality scores. Also [DKM16] proposes a set cardinality estimator which has been applied to biological data, but it is generic enough to be applied to other data types as well including graphs. For instance, cardinality estimators can quickly approximate vertex degrees in large graphs. Approximating a directed graph with an undirected one under sparsity constraints based on the polar decomposition of the adjacency matrix is explored in [DKMS21b]. Finally, in [DM22a] a genetic algorithm for quickly finding graph communities based on the Boolean semiring factorization of the adjacency matrix.

The first major SNA paper by the author is [DKMS17] where higher order vertex centrality metrics are implemented and analyzed. In [DKT17a] random walkers for fuzzy graphs are defined and used for graph clustering. Analysing the factors influencing brand loyalty for Twitter accounts is the focus of [KDK⁺19]. Searching in LinkedIn for trusted candidates was explored in [DKMAK20]. An algorithmic scheme for decompressing Twitter graphs while adding information learned by TSNs from similar graphs is described in [DKMI21]. Discovering sentiment potential in Twitter graphs is done in [DKMS21d]. Finally, clustering graphs based on structural, functional, and cognitive attributes with SOMs is the topic of [DGSM22].

Behavioral considerations were explored in [DGMS20c], where the influence of behavioral analytics in computer science is studied. In [DVM20a] the players of a cultural game have been classified based on the Bartle taxonomy, which has been used as the ground truth, using game features as well as cognitive attributes. Clustering fMRI images representing cognitive tasks is the central point of [DGMS20b].

6.3 Conclusions

6.3.1 Modeling power

Finding good models for humanistic data and, for that matter, for every type of data is more an art than a science as a myriad of factors should be taken into consideration. Tensors are at the core of higher order methods. Algorithms relying on them are more flexible in the sense that the more than one pair of variables can be used, as is the case with linear algebra. Although the latter could model some special cases, especially bilinear and quadratic functions, and with some tweaking maybe a few more cases or for certain special input, tensor algebra can naturally handle data with more complex structure such as graphs [DGMS20a; DKMS21b; DKMS17; KD14; Dra16], matrices [DKMAK20; DM22a; DKMS21d], images [DGMS20b], sets with relationships defined over its elements [DKM16], and documents [DKK⁺17].

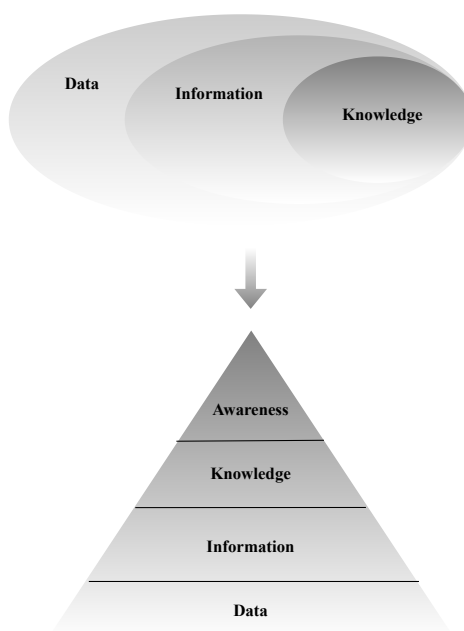


Figure 6.1: Proposed model.

Accurately capturing the interplay between the independent variables can lead not only to more efficient implementations with respect to code complexity, but also it can provide the intuition and inspiration necessary to develop groundbreaking algorithms relying on purely on tensors and the properties found in higher dimensions in order to yield novel results. The natural inclusion of multiple factors in tensors additionally allows the expression of non linear relationships suitably linearized through expansions such as Taylor or Volterra series, possibly in iterative schemes [DKMS17; DMS19]. The latter rely on linear kernels and have been applied to DSP including blind channel equalization and multiple input and multiple output (MIMO) digital telecommunications scenarios.

6.3.2 Computational complexity

Although the term *computational complexity* is broad enough to include a wide array of factors contributing to a computational task, in the context of this work the focus will be on the computational task per se in the sense of operation cost, whether these operations take place on a CPU or on a GPU or even on a dedicated TPU. Although this is a rather strong assumption given the vast performance differences stemming from the different technological paradigms and implementation limitations of these three platforms, it allows placing emphasis on the computational core of the problems at hand and thus a focus on algorithmic and implementation complexity. Additionally, this sort of complexity is perhaps the oldest performance benchmark for nearly all computational tasks and virtually all audiences are familiar with it. By definition, the approach used here to evaluate computational complexity excludes other common metrics such as communication and synchronization complexity, memory requirements and transfers across the memory hierarchy, code complexity, and networking bottlenecks.

Scale is instrumental, especially in graph mining and graph ML. Tensor factorizations and tensor based neural network architectures can efficiently create lower dimension projections [DGMS20b] of the data space which can be mined in order to yield approximate results with respect to some quality criterion [DVM20a]. Then these initial results can be progressively improved as more data arrive [KD14]. In this way not only meaningful results from large humanistic graphs can be obtained with reasonable computational cost. Extending this reasoning, results of various resolutions can be derived which, depending on the underlying engineering scenario, can be satisfactory.

6.3.3 Space complexity

With the recent advent of big data memory requirements both in terms of sheer data volume and of seeking data across memory hierarchy have become a major consideration. Memory requirements for large graphs are no exception to this as not only vertices, but also edges must be stored. For sparse or evolving graphs this is a major challenge since algorithmic schemes exploiting structural patterns or graph dynamics. Tensor factorizations yield sparse graph representations which can be stored in memory hierarchy levels closer to the actual processing units. Moreover, symmetries in these schemes can further decrease memory requirements. In this way the traffic between memory hierarchy levels is relatively low. However, this heavily depends on the optimality criterion for tensor factorizations, as the latter critically influences the form and sparsity of the factored tensor form [DM22a]. Along a similar line of reasoning the tensors in TSNs [DM20] and the tensor distance metric for SOMs [DGMS20b] can reduce memory requirements since they can yield lower dimension projections of the input tensors or, in certain cases, they can operate directly on them [DKMI21]. Additionally, clustering tensors before processing can have an effect to the main information processing similarly to the effect data preprocessing has on existing ML pipelines.

6.3.4 Digital awareness

The proposed model changes the relationship between data, information, and knowledge by introducing digital awareness as shown in figure 6.1. The traditional view is to pass from a very large amount of raw data to information and then to knowledge. The latter is typically non trivial information and it can be determined by a number of criteria, whether algorithmic or functional. This work proposes a pyramid structure where the dependency of each layer from the ones below it can be immediately seen. Additionally, digital awareness has been added to the top of the pyramid since it relies on refined knowledge extracted gradually from raw available data.

Digital awareness can be considerably increased because of the factors mentioned above. The intuitive or even the counterintuitive formulas can be inspiring and help gain a better understanding of the underlying field. Concrete examples shown in this work include:

- ♣ The clustering of fMRI images taken during the execution of various cognitive tasks [DGMS20a]. This can be attributed to the higher order high spatial correlation of pixels in brain imaging due to the high resolution of fMRI images as well as to the high degree of brain resiliency and reconfigurable rewiring. As shown in ...
- ♣ Understanding the resiliency of communication networks in terms of explainable and intuitive structural patterns as well as of the interplay thereof can assist human operators in quickly locating soft spots in critical digital infrastructure as shown in section 3.4.2.
- ♣ In the context of SNA efficiently discovering emotionally resilient Twitter communities can help social media operators and content managers to create tailored content, achieving thus a higher degree of content diffusion and visibility [DGMS20c; DGK20]. Moreover, finding affective coherent Twitter communities sheds light into community structure as well as to the ways ideas and messages pass through social media.

An important distinction must be made at this point. Digital awareness pertains to the enhancement of human perception through digital technology per definition 1.3. This is considerably different from the concept of explainable AI, which is currently the focus of intense interdisciplinary research. The latter is about designing efficient yet transparent AI whose results can be readily explained from human operators and stakeholders. Therefore, it focuses on algorithms, whereas digital awareness focuses on humans.

6.4 Contributions

6.4.1 Overview

In this section the research contributions by the author and the various distinctions of the work done in the context of this dissertation are explained here. Said research papers are categorized with respect to the field their primary topic pertains to and their relevance to this dissertation is assessed. Moreover, there is a short chronological analysis of the time these papers were published, so that the research flow, so to speak, can be understood. For the milestones in the research behind this dissertation, please see section 6.2.

6.4.2 Distinctions

To ensure the quality of this dissertation a number of steps were taken by the author in coordination with the academic advisor, including a detailed initial research proposal and a research plan which was annually updated. These are detailed below in addition to the distinctions which resulted from them, highlighting the level of effort put in this work and the consistency the research plan was implemented with.

- ♣ Each research paper included not only in the main text of this dissertation, but also cited therein were submitted exclusively to dblp⁶ indexed venues. This ensures a rigorous review process as well as high visibility. Moreover, easy and reliable access to these papers is guaranteed.

⁶www.dblp.org

- ♣ A language description file for Cypher⁷, the declarative query language Neo4j based on ASCII art^{8 9}, suitable for use with the *listings*¹⁰ package for L^AT_EX. The latter package allows the inclusion of source code snippets in L^AT_EX and X_LL^AT_EX documents, increasing thus their expression power.
- ♣ In 2018 the author won a Titan Xp 3000 GPU in a PhD competition. Said competition was based on the research proposals and the prospective applications of each PhD candidate and it was on a global scale. This acknowledges the research potential of the topic of this dissertation.
- ♣ A figure of merit is that [DKK⁺17] was one of the reference works of the Kaggle¹¹ COVID-19 scientific literature competition^{12 13}. Said competition has been organized in 2021, namely at the height of the pandemic and has been indexed by the ACM digital library¹⁴.
- ♣ Another figure of merit is the listing of [DLS⁺19] and [DTSM18] in the Wiki page¹⁵ of the online encyclopedia of integer sequences (OEIS)¹⁶.
- ♣ The author was the main organizer of two workshops pertaining to the topics of this dissertation. Said workshops were part of CIKM in 2021¹⁷ and 2022¹⁸ organized by ACM¹⁹. The activities of this workshop series are described in the respective editorials [DK21] and [DK22c].
- ♣ In 2022 the article [DKMAK22] submitted to IJAIT journal²⁰ won the Nikolaos Bourbakis award^{21 22}. This article was based on the work of the author. Also see figure 6.2.
- ♣ Additionally, the above workshop has been described in an article [DK22b] to ACM SIGIR Forum²³.
- ♣ At the time this dissertation was written, the author had an h-index of 19 as reported by Google Scholar²⁴.
- ♣ Finally, at this point it should be highlighted that the material presented here has been already published in its entirety in peer-reviewed conferences and journals. Therefore, an additional quality safeguard has been intentionally added to the content.

⁷It is a reference to the secondary antagonist of *The Matrix*.

⁸https://en.wikipedia.org/wiki/ASCII_art

⁹<https://www.asciart.eu>

¹⁰<https://ctan.org/pkg/listings>

¹¹<https://www.kaggle.com>

¹²<https://deepai.org/publication/covid-19-multidimensional-kaggle-literature-organization>

¹³<https://dl.acm.org/doi/10.1145/3469096.3474927>

¹⁴<https://dl.acm.org>

¹⁵<https://oeis.org/wiki/CiteD>

¹⁶<https://oeis.org>

¹⁷<https://www.cikm2021.org/programme/workshops>

¹⁸<https://www.cikm2022.org/workshops>

¹⁹www.acm.org

²⁰<https://www.worldscientific.com/worldscinet/ijait>

²¹<https://www.worldscientific.com/doi/10.1142/S0218213023820018>

²²<https://ionio.gr/en/news/28223>

²³<https://sigir.org/forum/issues/december-2022/>

²⁴<https://scholar.google.com/citations?user=QJOAYikAAAAJ>

In the remaining of this section the research papers by the author are summarized according to their respective fields. Therefore, this appendix serves as a central point for the titles of these papers as well as their respective overviews and their relevance to the topic of this work as described in the research proposal [Dra17]. They have been collected to a dedicated location such that the flow of the main text is not interrupted. Moreover, the reader can easily find the reference of a specific paper. It is noted that the bibliography section of this appendix covers the following two types of contributions:

- ♣ Research papers written by the author as partial fulfillment of the requirements for obtaining the PhD. The number of papers written every year during the PhD studies is shown in table 6.3.
- ♣ Contributions related to the above papers. They are papers which contributed to the PhD progress, but could not have been included in this dissertation due to space limitations.

The decision to include the contributions of the second category in the bibliography section of the appendix was taken on the grounds that although each research paper of the first category can undoubtedly be understood on its own right, knowledge of the preceding papers can give context information as well as intuition to the reader. In any case, the papers of the second category were not factored in the results shown in table 6.3, where the number of contributions per year is shown as stated earlier.

Table 6.3: Contributions per year.

2017	2018	2019	2020	2021	2022	2023	2024	Total
4	1	2	10	7	3	4	1	32

Another way to look at the contributions is to group them by field. In table 6.4 the same contributions are now shown with respect to their primary field. From the amount of research papers it follows that the focus was almost equally divided on abstract topics, namely deep learning and graph mining, and on applications, namely social network analysis and behavioral sciences. This balanced approach was in part inspired by the need to demonstrate the viability and feasibility of the proposed techniques. This in turn generated new questions based on observations of the application of the latter in practice, effectively becoming the respective central points of more contributions. In other words, this was an upwards spiral which has already bore and, most importantly, still keeps bearing fruits. This is also consistent with the applied nature of computer science. Moreover, it is worth mentioning that both application fields not only are relatively new, but also connect computer science with other sciences such as sociology and psychology. This clearly illustrates the interdisciplinary nature of the research behind this PhD.

Table 6.4: Number of contributions per field.

Deep learning	Graph mining	Social networks	Behavioral	Total
8	9	9	6	32

In order to evaluate the global research interest of the four fields of table 6.4, the four entries of these table were queried against Google Scholar²⁵ for the duration of this PhD. The results are shown in table 6.5 and they indicate a significant interest in overall.

²⁵Access date: June 24th 2024.

Table 6.5: Scientific works per field per year.

Year	Deep learning	Graph mining	Social networks	Behavioral sciences
2017	685K	183K	1.77M	1.18M
2018	1.1M	201K	1.7M	1.13M
2019	1.18M	183K	1.43M	967K
2020	1.11M	190K	1.4M	835K
2021	827K	161K	1.05M	611K
2022	496K	131K	0.57M	395K
2023	261K	76.2K	0.34M	243K
2024 (8 mo.)	156K	42.5K	0.19M	181K
Total	5.815M	1.28M	8.45M	5.54M

Along a similar line of reasoning the term *humanistic data* was used as input to Google Scholar²⁶. The results are shown in table 6.6. Because of the interdisciplinary nature of these data, these results were not constrained to computer science research papers.

Table 6.6: Scientific works for humanistic data per year.

2017	2018	2019	2020	2021	2022	2023	2024	Total
42.3K	42.2K	44.3K	44.6K	43.8K	43.5K	32.9K	16.4K	310K

Please note there is an intrinsic relationship between the topics listed above which not only leads to an overall thematic coherence of this dissertation, but also results in a unified and easily understood story summarizing the greater context of mining graphs created from humanistic data. Therefore, the reader can benefit in a multitude of ways. Additionally, true to its topic in a meta level, the structure of this text also exhibits a higher order structure expressed as a sequence of pairwise conceptual interactions. Specifically, the following connections were explored between the research papers of tables 6.7, 6.8, 6.9, and 6.10:

- ♣ DL methodologies are versatile enough to be applied to the rest of the abovementioned fields. This unifies algorithmically and technologically the topics of this work in a single and coherent narrative.
- ♣ Graph mining techniques besides being of interest on their own right they can be directly applied to SNA. Moreover, they rely heavily on DL to natively and efficiently handle graphs.
- ♣ SNA is currently perhaps the most well known application of graph mining and the source of the largest graph datasets, some of which are publicly accessible through digital repositories like SNAP²⁷.
- ♣ Behavioral sciences can benefit from graph mining and DL in order to discover behavioral patterns. Additionally, they can be applied to SNA in the case of affective communities.

The remainder of this section is structured as follows. In section 6.4.3 contributions pertaining to DL including GNNs and tensor distance metrics. Section 6.4.4 contains the research papers relevant to graph

²⁶Access date as before.

²⁷<http://snap.stanford.edu>

mining. They are more abstract contributions with a broad spectrum of applications. In the related section 6.4.5 the contributions to SNA, some of them based on the background work on graph mining, are explained. Works with a behavioral focus are collected in section 6.4.6. Finally, the software used in this work is mentioned in section 6.4.7.

6.4.3 Deep learning

The field of DL is currently one of the major drivers for a wide array of technological and financial fields including medical imaging, digital content creation, e-commerce, autonomous vehicles, smart contracts, and robotics to name only a few. Especially humanistic data processing, which is characterized by the challenges described in chapter 1, has been made considerably easier through advanced and disruptive DL algorithmic breakthroughs with various deep architectures including GNNs, TSNs, CNNs, RNNs, LSTMs, autoencoders, GANs, and more recently temporal networks and transformers. DL methodologies tailored for or applied to humanistic data tend to have one or more of the following characteristics:

- ◆ The extraction of non-trivial latent knowledge from sources generating massive human activity related data. The majority thereof comes from online social networks, in particular Twitter and more recently Reddit. Both social media offer easy and intuitive APIs and additionally have the least number of constraints as to the volume of the data collected and as to the time window for doing so. Although the methodologies developed for mining social media heavily rely on the properties of the underlying medium, they can often be adapted to other fields also built around linked data. This is particularly true for techniques exploiting primarily structural aspects such as connectivity patterns.
- ◆ Methodologies designed for interconnected data may well result in the discovery of new dependencies, leading in turn to more accurate and reliable models for the application field. This is an especially important property for humanistic data which are frequently created by highly non linear processes. Although the latter are typically inaccessible or at least not readily observable, their working dynamics and complexity are reflected in the various higher order and memory patterns inherent in the data. Consequently, algorithmic mining strategies must have comparable complexity if meaningful results are to be eventually obtained from higher order humanistic data.
- ◆ When a data model is readily available from the problem description or it is inferred or even partially estimated from observations, then intelligent or adaptive control techniques can be employed in order to predict the trajectory of the humanistic data source in an appropriately selected state space. This can subsequently be the basis for further processing such as similarity comparisons, selecting between alternative trajectories, or deducing analytically or numerically whether said trajectory can reach a specific point and, if so, after how much time. Therefore, such models can capture essential aspects of the underlying dynamics and translate them to invaluable performance metrics.
- ◆ Given the sheer data volume in most realistic scenaria, dimensionality reduction is a major goal. Linear algebraic decompositions such as SVD or polar factorization achieve exactly that and so do tensor decompositions. The latter allow the creation of one or possibly more efficient and concise data representations lending themselves to subsequent tractable processing. This includes among others sorting, interpolation, extrapolation, filtering, clustering, as well as various queries including temporal and spatial ones. Moreover, depending on the nature of the application, tailored and efficient aggregators can be built.

Deep neural network architectures fit the above points by offering considerable reconfiguration capabilities. Since these architectures have, as stated earlier, a high degree of adaptability, they are one of the learning mainstays. Another DL driver is the highly efficient code of contemporary languages which are suitable for data driven scenarios such as Julia. Since many of such languages are interpreted, self referential code can be written in order to decrease development time and generate template code. This can be achieved in a variety of methodologies including code reflection and polymorphic code, both natively supported in Julia. Here it is worth mentioning that code reflection is a powerful programming technique since the days of DOS viruses including (c)Brain²⁸ and the polymorphic²⁹ and metamorphic³⁰ ones.

In table 6.7 are shown the research papers covering recent DL topics. In particular, there can be seen a variety of contributions whose common denominator is the use of a DL methodology in order to discover knowledge in humanistic data and mine it efficiently.

Table 6.7: Deep learning contributions.

Work	Relevance
[KDM24]	GNNs for Industry 4.0 process graphs
[DGSM22]	Tensor distances combined with an SOM architecture for recommendation
[DM22a]	Genetic algorithm for Boolean semiring matrix factorization
[DKMS21b]	Iterative matrix polar decomposition for approximating directed Twitter graphs
[DK21]	A variant of [DGMS20a] for fuzzy Twitter graphs
[DKMS21a]	Overview of DL challenges in the behavioral sciences field
[DGMS20a]	GNN for evaluating the affective coherence of Twitter graphs
[DGMS20b]	Tensor distance metrics for task classification from fMRI images
Total	8

In table 6.7 the first research paper listed, namely [KDM24], focuses on GNNs for Industry 4.0 process graphs. The latter frequently have multiple edges with different labels between the same pair of vertices. By exploiting the coherence between sets of labeled edges, the accuracy in recovering noisy processes graphs increases significantly. The GNN architectures proposed in [DKMS21a] and [DGMS20a] discover affective coherent Twitter communities. Tensor distance metrics are the focus of [DGSM22] and [DGMS20b]. Approximating directed graphs with undirected ones based on density through polar factorization is at the epicenter of [DKMS21b]. Discovering graph communities based on the Boolean semiring factorization of the graph adjacency matrix is explored in [DM22a]. Finally, DL methodologies with affective applications are surveyed in [DKMS21a].

6.4.4 Graph mining

Graph mining plays a crucial role, quite frequently in the form of an analytics stack for very large graphs, across numerous applications. The topics of the research papers listed in table 6.8 are quite representative of the multitude of possible applications.

²⁸[https://en.wikipedia.com/wiki/Brain_\(computer_virus\)](https://en.wikipedia.com/wiki/Brain_(computer_virus))

²⁹<https://www.kaspersky.com/resource-center/definitions/what-is-a-polymorphic-virus>

³⁰<https://www.kaspersky.com/resource-center/definitions/metamorphic-virus>

Table 6.8: Graph mining contributions.

Work	Relevance
[KD14]	Space efficient data structure for persistent graphs
[DLTM18]	Graph resilience metric computation with parallel computing
[Dra16]	Higher order centrality metrics in Java over Neo4j
[DBM15]	Tensor fusion of centrality metrics
[DKMI21]	Twitter graph decompression with TSNs
[DM20]	Graph resiliency evaluation with TSNs
[DGK20]	Graph community discovery algorithms with Neo4j
[DKMAK20]	LinkedIn search algorithms for finding trusted candidates
[DK20]	Topological graph correlation coefficient
Total	9

The focus of [DBM15] is using the tensor adjacency matrix to combine Twitter influence metrics to a higher order one, which is continued in [Dra16]. Therein tensor fibers play an important role. Adjacency tensors are used to combine structural attributes with existing knowledge from similar graphs for decompressing Twitter graphs in [DKMI21]. Along a similar line of reasoning tensors are used to represent multilayer graphs built from LinkedIn attributes to find trusted candidates in [DKMAK20]. Multilayer graphs can be represented by adjacency third order tensors in a very similar way ordinary graphs can be represented by adjacency matrices. In the former case the edges of a single graph layer form one tensor layer, typically along the first and second dimension, namely along rows and columns respectively, leaving thus the third tensor dimension, frequently thought of consisting of fibers, for layer indexing. However, it should be emphasized that there are other ways to algebraically represent multilayer graphs. Graph resiliency is the topic of both [DM20] and [DLTM18]. In the former TSNs are used to assess structural resiliency as shown in chapter 3, whereas in the latter a closed form structural metric is proposed. Approximations to the topological correlation coefficient are explored in [DK20]. Finally, in [KD14] a novel technique for fully persistent graphs based on splay trees is presented. It is generic enough to be implemented in diverse computational environments and development stacks. Moreover, it lays the groundwork for a Java graph processing framework, currently under development.

6.4.5 Social network analysis

The advent of interconnected era has been marked by the rapid development of a global transportation and communication networks similar only perhaps to those which facilitated the consolidation of the British empire over vast land and sea masses in the Victorian era³¹. Arguably it is being currently witnessed the bottom up creation of a planetary digital nervous system, driven primarily by the insatiable human need for unobstructed communication. Content creation and dissemination has reached unsurpassed levels and it fuels, along with other factors, the technological civilization. Table 6.9 lists the relevant papers.

Finding influential Twitter accounts based on a multitude attributes provided by Apache Storm is the focus of [SDK⁺21]. Selecting databases for storing social graphs is facilitated using the recommendations proposed in [MDMS21]. Higher order properties of humanistic data are exploited in both [DKMS17] and [DGGS20]. In the former Twitter composite centrality metrics based on existing ones are proposed, while

³¹https://en.wikipedia.org/wiki/Victorian_era

Table 6.9: Social network contributions.

Work	Relevance
[MDMS21]	Recommending database architectures for social mining tasks
[SDK ⁺ 21]	Discovering influential Twitter accounts with Apache Storm attributes
[DSK ⁺ 20]	Spatiosocial tensor clustering to discover linguistic Twitter communities
[DGGS20]	Multicriterial community discovery in Twitter
[KDK ⁺ 19]	Prediction of brand loyalty for Twitter verified accounts
[DKT16b]	Characterizing Twitter influence with system theory concepts
[DKMS17]	Higher order account centrality metrics
[DKT17a]	Random walkers for clustering fuzzy graphs
[DST ⁺ 17]	Application of tensors to ontologies
Total	9

in the latter a community discovery scheme based on multiple criteria fused in a tensor is given. Clustering fuzzy graphs with the help of random walkers whose probability of crossing an edge is a function of the probability of that edge belonging to the graph is explained in [DKT17a]. The brand loyalty of verified vs unverified Twitter accounts is the subject of [KDK⁺19]. Using system theory concepts to find influential Twitter accounts is explored in [DKT16b]. Clustering a third order tensor representing various linguistic and structural attributes of Twitter accounts yields spatiosocial communities in [DSK⁺20]. Lastly, tensors can be used to extend ontologies in [DST⁺17].

6.4.6 Computational behavioral sciences

Behavioral sciences (BH) and their computational aspects have lately emerged as a focal point of interdisciplinary research. Moreover, they are a prime source of humanistic data of high variability, large size, and with higher order dependencies.

Table 6.10: Behavioral science contributions.

Work	Relevance
[DVMS21]	Affective applications of graph mining
[DVM20a]	Player classification based on the Bartle taxonomy with behavioral attributes
[DGMS20c]	Study of the convergence of behavioral science and cultural content
[DPSP19]	Speech emotion recognition
[DKK ⁺ 17]	Biomedical document tensor clustering
[DM22c]	Behavioral boosting of contemporary educational trends
Total	6

The Bartle taxonomy is currently the basic player categorization scheme which has been extensively used in the gaming industry for developing analytics and it is rooted in behavioral patterns [DVM20a]. An overview of graph mining to SNA from an affective standpoint is the topic of [DVMS21]. A more extensive survey is [DGMS20c]. Speech emotion recognition is the focus of [DPSP19]. Third order tensors extending the term-document matrix are clustered in [DKK⁺17]. Finally, the behavioral aspects of current digital education trends are explored in [DM22c].

6.4.7 Computational tools

Python³² has evolved over the years to a *lingua franca* in the fields data science and MLd due to its widespread adoption. The primary reasons for this are the following:

- ♣ The vast collection of libraries and supporting utilities such as package managers.
- ♣ The short development and debugging times.
- ♣ The easy code and dependency management through virtual environments.
- ♣ The use of Python in nearly all computer science university departments.

TensorFlow³³ is a low level library with Python API among others. TensorFlow can naturally accept and handle tensors of multiple dimensions, including sparse ones, and create operation execution graphs. Sparse tensors are critical in scalable DL algorithms as explained in subsection 6.3.3. Another popular package is PyTorch³⁴, which offers numerous DL methods.

Some of the latest parts of this work have been written in Julia in the form of applications relying on its wide array of built-in methods, native support for multidimensional arrays, and rarely on third party packages written for graph mining. Julia has a persistent reputation for having a strong and committed community frequently engaging in vigorous testing. Additionally, there is a wide choice of ways for learning the language, most of which can be found at the central Julia Web site³⁵, including videos and hands-on programming exercises. Finally, the similarity of the name and syntax of Julia methods with their MATLAB counterparts makes transition from the latter to the former easier.

Neo4j³⁶ is one of the leading graph databases as well as one of the prominent examples of NoSQL collection of database paradigms. Neo4j offers both functional capabilities and declarative ones. The latter are done through Cypher, an ASCII art based language which naturally expresses simple graph patterns. One of the most popular Python drivers for Neo4j is *py2neo*.

6.5 Prospects

The topic of this dissertation is the development of higher order knowledge mining techniques for graphs created by humanistic data. The inherently distributed nature of information in such graphs plus the challenges of mining it including potentially missing values, growing scale, and evolving nature make the invention of such techniques difficult. Given that humanistic data are literally everywhere where human activity can be found and moreover that such data can often be easily and naturally treated as graphs, the value in extracting tangible and working knowledge from such data is tremendous as generally shown in chapter 1 and additionally explained in chapters 3, 4, and 5.

³²<https://www.python.org>

³³<https://www.tensorflow.org>

³⁴<https://pytorch.org>

³⁵www.julialang.org

³⁶<https://neo4j.org>

- ♣ Tensors allow for better understanding of the interaction between multiple factors as well as separately analyzing each. This is especially important in humanistic data like images and graphs where spatial, temporal, and functional proximity modes are equally important. Decoupling these modes in the general case is naturally done with tensors. However, the transformations achieving this can be expensive. Thus, efficient or approximate versions thereof need to be found.
- ♣ Preprocessing with the intent of sparsification is paramount for efficient tensor processing, whether in a TeSP or in a TSN setting. A suitably designed preprocessing scheme can save a considerable number of either operations or block moves across the memory hierarchy or even across clusters. Preprocessing cost is thus justified, especially when it is run only once against multiple data tensors.
- ♣ Knowledge transfer from DSP can lead to adaptive filtering specifically for tensor data. For instance, adjacency matrix factorization similar to that of Lee-Seung iterative scheme or polar factorization can be developed. Alternatively transform domain algorithms based on the Laplace tensor of a graph can be instrumental in analyzing humanistic data.
- ♣ Distributed large scale implementations of the above in distributed digital ecosystems like Spark³⁷ taking advantage of the directed acyclic graph model can be the computational core of the future massive graph processing systems. Such systems should be built with link oriented data in mind from the beginning and provide native support for graph data structures.
- ♣ Regarding GNNs, their scaling capabilities should be further investigated along with any advantages memory effects may have to offer, as locality is not the only major graph property. Moreover, the effect of strategic vertex drop out instead of randomly doing so can be explored.
- ♣ Aggregation models in GNNs operating at various scales and selecting different attribute sets or sketches thereof should be developed. This should lead to GNNs simultaneously viewing different aspects of the data and, thus, yielding results depending on the processing scale.

Given the above, the research prospects in the field look bright.

Easter eggs: There are in in the text. They are not so well hidden, so that the reader can spot them.

If you have read up to here, thank you!

³⁷<https://spark.apache.org>

International Journal on Artificial Intelligence Tools
 Vol. 32, No. 3 (2023) 2382001 (1 page)
 © World Scientific Publishing Company
 DOI: 10.1142/S0218213023820018



Winners of Nikolaos Bourbakis Award for 2022

IJAIT committee for the Nikolaos Bourbakis Award decided, for implementation of the award for 2022, to evaluate all regular papers published in IJAIT in 2022 as well as those in the special issue dedicated to ICTAI-20 conference.

The committee based its evaluation on the comments of the reviewers, at a first stage, and on their content (problem under study, approach followed, mathematical modeling), at a second stage. Finally, the committee decided that the following paper is the winner of Nikolaos Bourbakis award for 2022:

- **G. Drakopoulos, E. Kafeza, P. Mylonas and H. Al Katheeri**, Higher Order Trust Ranking of LinkedIn Accounts with Iterative Matrix Methods, *Int. J. Artif. Intell. Tools*, Vol. 31, No. 7, 2260001-1–23, November 2022, doi: 10.1142/S0218213022600016.

The US\$500 cash award will be shared among the authors of the paper.

Congratulations to all authors of the paper!

Figure 6.2: Announcement of the 2022 Bourbakis award.

6.6 Bibliography

- [ADK⁺20a] Athanasios Alexopoulos, Georgios Drakopoulos, Andreas Kanavos, Phivos Mylonas, and Gerasimos Vonitsanos. Two-step classification with SVD preprocessing of distributed massive datasets in Apache Spark. *Algorithms*, 13(3), 2020.
- [ADK⁺20b] Athanasios Alexopoulos, Georgios Drakopoulos, Andreas Kanavos, Spyros Sioutas, and Gerasimos Vonitsanos. Parametric evaluation of collaborative filtering over Apache Spark. In *SEEDA-CECNSM*. IEEE, 2020.
- [DB15] Georgios Drakopoulos and Aikaterini Baroutiadi. Implementing centrality measures for Neo4j. In *8th ECESCON*. ECESCON, 2015.
- [DBM15] Georgios Drakopoulos, Aikaterini Baroutiadi, and Vasileios Megalooikonomou. Higher order graph centrality measures for Neo4j. In *IISA*. IEEE, 2015.
- [DBM24] Georgios Drakopoulos, Georgios Bardis, and Phivos Mylonas. Power iteration graph clustering with funtools higher order methods. In *SMAP*. IEEE, 2024.
- [DGGS20] Drakopoulos Drakopoulos, Konstantinos C. Giotopoulos, Ioanna Giannoukou, and Spyros Sioutas. Unsupervised discovery of semantically aware communities with tensor Kruskal decomposition: A case study in Twitter. In *SMAP*. IEEE, 2020.
- [DGK20] Georgios Drakopoulos, Panagiotis Gourgarris, and Andreas Kanavos. Graph communities in Neo4j. *EVOS*, 11(3):397–407, 2020.
- [DGKM16] Georgios Drakopoulos, Panagiotis Gourgarris, Andreas Kanavos, and Christos Makris. A fuzzy graph framework for initializing k-means. *IJAIT*, 25(6):1–21, 2016.
- [DGMS20a] Georgios Drakopoulos, Ioanna Giannoukou, Phivos Mylonas, and Spyros Sioutas. A graph neural network for assessing the affective coherence of Twitter graphs. In *IEEE Big Data*, pages 3618–3627. IEEE, 2020.
- [DGMS20b] Georgios Drakopoulos, Ioanna Giannoukou, Phivos Mylonas, and Spyros Sioutas. On tensor distances for self organizing maps: Clustering cognitive tasks. In *DEXA*, volume 12392 of *LNCS*, pages 195–210. Springer, 2020.
- [DGMS20c] Georgios Drakopoulos, Ioanna Giannoukou, Phivos Mylonas, and Spyros Sioutas. The converging triangle of cultural content, cognitive science, and behavioral economics. In *MHDW*, volume 585 of *IFIP Advances in Information and Communication Technology*, pages 200–212. Springer, 2020.
- [DGSM22] Georgios Drakopoulos, Ioanna Giannoukou, Spyros Sioutas, and Phivos Mylonas. Self organizing maps for cultural content delivery. *NCAA*, 31(7), 2022.
- [DK16] Georgios Drakopoulos and Andreas Kanavos. Tensor-based document retrieval over Neo4j with an application to PubMed mining. In *IISA*. IEEE, 2016.
- [DK20] Georgios Drakopoulos and Eleanna Kafeza. One dimensional cross-correlation methods for deterministic and stochastic graph signals with a Twitter application in Julia. In *SEEDA-CECNSM*. IEEE, 2020.

- [DK21] Georgios Drakopoulos and Eleanna Kafeza. THECOG - Transforms in behavioral and affective computing. In *CIKM*, pages 4876–4877. ACM, 2021.
- [DK22a] Georgios Drakopoulos and Eleanna Kafeza. Estimating the MBTI of Twitter accounts with graph neural networks over Neo4j. In *THECOG*. CEUR-WS, 2022.
- [DK22b] Georgios Drakopoulos and Eleanna Kafeza. Report on the 2nd international workshop on transforms in behavioral and affective computing (THECOG 2022) at CIKM 2022. *SIGIR Forum*, 56(2), December 2022.
- [DK22c] Georgios Drakopoulos and Eleanna Kafeza. THECOG 2022 - Transforms in behavioral and affective computing (Revisited). In *CIKM*. ACM, 2022.
- [DKAK19] Georgios Drakopoulos, Eleanna Kafeza, and Haseena Al Katheeri. Proof systems in blockchains: A survey. In *SEEDA-CECNSM*. IEEE, 2019.
- [DKG⁺22] Georgios Drakopoulos, Eleanna Kafeza, Ioanna Giannoukou, Phivos Mylonas, and Spyros Sioutas. Simulating blockchain consensus protocols in Julia: Proof of work vs proof of stake. In *MHDW*. Springer, 2022.
- [DKK⁺17] Georgios Drakopoulos, Andreas Kanavos, Ioannis Karydis, Spyros Sioutas, and Aristidis G. Vrahatis. Tensor-based semantically-aware topic clustering of biomedical documents. *Computation*, 5(3), 2017.
- [DKK⁺22] Georgios Drakopoulos, Aristeidis Karras, Christos Karras, Konstantinos C Giotopoulos, Phivos Mylonas, and Spyros Sioutas. Decomposing Twitter graphs based on hashtag trajectories: Mining and clustering paths over MongoDB. In *SETN*. ACM, 2022.
- [DKM16] Georgios Drakopoulos, Stavros Kontopoulos, and Christos Makris. Eventually consistent cardinality estimation with applications in biodata mining. In *SAC*. ACM, 2016.
- [DKMAK20] Georgios Drakopoulos, Eleanna Kafeza, Phivos Mylonas, and Haseena Al Katheeri. Building trusted startup teams from LinkedIn attributes: A higher order probabilistic analysis. In *ICTAI*, pages 867–874. IEEE, 2020.
- [DKMAK22] Georgios Drakopoulos, Eleanna Kafeza, Phivos Mylonas, and Haseena Al Katheeri. Higher order trust ranking of LinkedIn accounts with iterative matrix methods. *IJAIT*, 31(7), 2022.
- [DKMI21] Georgios Drakopoulos, Eleanna Kafeza, Phivos Mylonas, and Lazaros Iliadis. Transform-based graph topology similarity metrics. *NCAA*, 33(23):16363–16375, 2021.
- [DKMM15] Georgios Drakopoulos, Andreas Kanavos, Christos Makris, and Vasileios Megalooikonomou. On converting community detection algorithms for fuzzy graphs in Neo4j. In *CIMA*. IEEE, 2015.
- [DKMM17] Georgios Drakopoulos, Andreas Kanavos, Christos Makris, and Vasileios Megalooikonomou. Comparing algorithmic principles for fuzzy graph communities over Neo4j. In I. Hatzilygeroudis, V. Palade, and J. Prentzas, editors, *Advances in Combining Intelligent Methods*. Springer, 2017.

- [DKMP20] Georgios Drakopoulos, Andreas Kanavos, Phivos Mylonas, and Panagiotis Pintelas. Extending fuzzy cognitive maps with tensor-based distance metrics. *Mathematics*, 8(11), 2020.
- [DKMS17] Georgios Drakopoulos, Andreas Kanavos, Phivos Mylonas, and Spyros Sioutas. Defining and evaluating Twitter influence metrics: A higher order approach in Neo4j. *SNAM*, 71(1), 2017.
- [DKMS21a] Georgios Drakopoulos, Eleanna Kafeza, Phivos Mylonas, and Spyros Sioutas. A graph neural network for fuzzy Twitter graphs. In Gao Cong and Maya Ramanath, editors, *CIKM companion volume*, volume 3052. CEUR-WS.org, 2021.
- [DKMS21b] Georgios Drakopoulos, Eleanna Kafeza, Phivos Mylonas, and Spyros Sioutas. Approximate high dimensional graph mining with matrix polar factorization: A Twitter application. In *IEEE Big Data*, pages 4441–4449. IEEE, 2021.
- [DKMS21c] Georgios Drakopoulos, Eleanna Kafeza, Phivos Mylonas, and Spyros Sioutas. Process mining analytics for Industry 4.0 with graph signal processing. In *WEBIST*, pages 553–560. SCITEPRESS, 2021.
- [DKMS21d] Georgios Drakopoulos, Andreas Kanavos, Phivos Mylonas, and Spyros Sioutas. Discovering sentiment potential in Twitter conversations with Hilbert-Huang spectrum. *EVOS*, 12(1):3–17, 2021.
- [DKP⁺20] Georgios Drakopoulos, Andreas Kanavos, Konstantinos Paximadis, Ilias Aristidis, Christos Makris, and Phivos Mylonas. Computing massive trust analytics for Twitter using Apache Spark with account self-assessment. In *WEBIST*, pages 403–414. SCITEPRESS, 2020.
- [DKT16a] Georgios Drakopoulos, Andreas Kanavos, and Athanasios Tsakalidis. A Neo4j implementation of fuzzy random walkers. In *SETN*. ACM, 2016.
- [DKT16b] Georgios Drakopoulos, Andreas Kanavos, and Athanasios Tsakalidis. Evaluating Twitter influence ranking with system theory. In *WEBIST*. SCITEPRESS, 2016.
- [DKT17a] Georgios Drakopoulos, Andreas Kanavos, and Konstantinos Tsakalidis. Fuzzy random walkers with second order bounds: An asymmetric analysis. *Algorithms*, 10(2), 2017.
- [DKT⁺17b] Georgios Drakopoulos, Andreas Kanavos, Dimitrios Tsolis, Phivos Mylonas, and Spyros Sioutas. Towards a framework for tensor ontologies over Neo4j: Representations and operations. In *IISA*. IEEE, 2017.
- [DLS⁺19] Georgios Drakopoulos, Xenophon Liapakis, Evaggelos Spyrou, Giannis Tzimas, Phivos Mylonas, and Spyros Sioutas. Computing long sequences of consecutive Fibonacci integers with TensorFlow. In *MHDW*, volume 560 of *IFIP Advances in Information and Communication Technology*, pages 150–160. Springer, 2019.
- [DLTM18] Georgios Drakopoulos, Xenophon Liapakis, Giannis Tzimas, and Mylonas Mylonas. A graph resilience metric based on paths: Higher order analytics with GPU. In *ICTAI*, pages 884–891. IEEE, 2018.
- [DM15] Georgios Drakopoulos and Vasileios Megalooikonomou. On the weight sparsity of multilayer perceptrons. In *IISA*. IEEE, 2015.

- [DM16a] Georgios Drakopoulos and Vasileios Megalooikonomou. An adaptive higher order scheduling policy with an application to biosignal processing. In *SSCI*. IEEE, 2016.
- [DM16b] Georgios Drakopoulos and Vasileios Megalooikonomou. Regularizing large biosignals with finite differences. In *IISA*. IEEE, 2016.
- [DM20] Georgios Drakopoulos and Phivos Mylonas. Evaluating graph resilience with tensor stack networks: A keras implementation. *NCAA*, 32(9):4161–4176, 2020.
- [DM22a] Georgios Drakopoulos and Phivos Mylonas. A genetic algorithm for Boolean semiring matrix factorization with applications to graph mining. In *Big Data*. IEEE, 2022.
- [DM22b] Georgios Drakopoulos and Phivos Mylonas. Extreme learning machines for efficient speech emotion estimation in Julia. In *THECOG*. CEUR-CS, 2022.
- [DM22c] Georgios Drakopoulos and Phivos Mylonas. Facilitating current higher education trends with behavioral strategies. In *SMAP*. IEEE, 2022.
- [DM23] Georgios Drakopoulos and Phivos Mylonas. Higher order probabilistic analysis of network trajectories of intelligent agents in Thespian. In *SEEDA-CECNSM*. IEEE, 2023.
- [DM24] Georgios Drakopoulos and Phivos Mylonas. Clustering MBTI personalities with graph filters and self organizing maps over Pinecone. In *Big Data*. IEEE, 2024. submitted.
- [DMK15] Georgios Drakopoulos, Elisseos Mavratzotis, and Konstantinos Kouvelis. Developing a heuristic framework for recommending wargame implementations. In *T TSAAMS*, 2015.
- [DML⁺18] Georgios Drakopoulos, Michail Marountas, Xenophon Liapakis, Giannis Tzimas, Phivos Mylonas, and Spyros Sioutas. Blockchain for mobile health applications: Acceleration with GPU computing. In Panayiotis Vlamos, editor, *GeNeDis 2018*. Springer, 2018.
- [DMS18] Georgios Drakopoulos, Phivos Mylonas, and Spyros Sioutas. An architecture for cooperative mobile health applications. In Panayiotis Vlamos, editor, *GeNeDis 2018*. Springer, 2018.
- [DMS19] Georgios Drakopoulos, Phivos Mylonas, and Spyros Sioutas. A case of adaptive nonlinear system identification with third order tensors in TensorFlow. In *INISTA*. IEEE, 2019.
- [DPSP19] Georgios Drakopoulos, George Pikramenos, Evangelos D. Spyrou, and Stavros J. Perantonis. Emotion recognition from speech: A survey. In *WEBIST*. SCITEPRESS, 2019.
- [Dra16] Georgios Drakopoulos. Tensor fusion of social structural and functional analytics over Neo4j. In *IISA*. IEEE, 2016.
- [Dra17] Georgios Drakopoulos. Knowledge mining with tensor algebra. Technical report, Ionian University, 2017.
- [Dra23] Georgios Drakopoulos. PhD final report. Technical report, Ionian University, 2023.
- [DSK⁺20] Georgios Drakopoulos, Foteini Stathopoulou, Andreas Kanavos, Michael Paraskevas, Giannis Tzimas, Phivos Mylonas, and Lazaros Iliadis. A genetic algorithm for spatio-social tensor clustering: Exploiting TensorFlow potential. *EVOS*, 11(3):491–501, 2020.

- [DSM19] Georgios Drakopoulos, Evaggelos Spyrou, and Phivos Mylonas. Tensor clustering: A review. In *SMAP*. IEEE, 2019.
- [DST⁺17] Georgios Drakopoulos, Foteini Stathopoulou, Giannis Tzimas, Michael Paraskevas, Phivos Mylonas, and Spyros Sioutas. A genetic algorithm for discovering linguistic communities in spatio-social tensors with an application to trilingual Luxembourg. In *MHDW*. Springer, 2017.
- [DSVM19] Georgios Drakopoulos, Evaggelos Spyrou, Yorghos Voutos, and Phivos Mylonas. A semantically annotated JSON metadata structure for open linked cultural data in Neo4j. In *PCI*. ACM, 2019.
- [DTSM18] Georgios Drakopoulos, Dimitrios Tsolis, Antonia Stefani, and Phivos Mylonas. The biomolecular computation paradigm: A survey in massive biological computation. In *MHDW*, volume 520 of *IFIP Advances in Information and Communication Technology*, pages 157–167. Springer, 2018.
- [DVM20a] Georgios Drakopoulos, Yorghos Voutos, and Phivos Mylonas. Annotation-assisted clustering of player profiles in cultural games: A case for tensor analytics in Julia. *BDCC*, 4(4), 2020.
- [DVM20b] Georgios Drakopoulos, Yorghos Voutos, and Phivos Mylonas. Recent advances on ontology similarity metrics: A survey. In *SEEDA-CECNSM*. IEEE, 2020.
- [DVMS21] Georgios Drakopoulos, Yorghos Voutos, Phivos Mylonas, and Spyros Sioutas. Motivating item annotations in cultural portals with UI/UX based on behavioral economics. In *IISA*. IEEE, 2021.
- [KD14] Stavros Kontopoulos and Georgios Drakopoulos. A space efficient scheme for graph representation. In *ICTAI*, pages 299–303. IEEE, 2014.
- [KDK⁺19] Ioanna Kyriazidou, Georgios Drakopoulos, Andreas Kanavos, Christos Makris, and Phivos Mylonas. Towards predicting mentions to verified Twitter accounts: Building prediction models over MongoDB with Keras. In *WEBIST*. SCITEPRESS, 2019.
- [KDM24] Eleanna Kafeza, Georgios Drakopoulos, and Phivos Mylonas. Graph neural networks in PyTorch for link prediction in Industry 4.0 process graphs. In *AIAI*, IFIP. Springer, 2024.
- [KDT17] Andreas Kanavos, Georgios Drakopoulos, and Athanasios Tsakalidis. Graph community discovery algorithms in Neo4j with a regularization-based evaluation metric. In *WEBIST*, pages 403–410. SCITEPRESS, 2017.
- [MDI13] Nikolaos Mavridis, Georgios Drakopoulos, and Konstantinos Iliopoulos. Enriching qualitative trajectory calculus: Analysis, reconstruction, and extension to 3D. Technical report, NCSR Democritus, 2013.
- [MDMS21] Michael Marountas, Georgios Drakopoulos, Phivos Mylonas, and Spyros Sioutas. Recommending database architectures for social queries: A Twitter case study. In *AIAI*. Springer, 2021.
- [MDV⁺14] Elissaios Mavratzotis, Georgios Drakopoulos, Athanasios Voulodimos, Aggelos Vatikalos, Konstantinos Kouvelis, Sotiris Papadopoulos, and Michail Sakelariou. Evaluating UAV impact in the tactical context of a mechanized infantry scout platoon through military simulation

- software. In Nikolaos Daras, editor, *Applications of Mathematics and Informatics in Science and Engineering*, pages 161–172. Springer, 2014.
- [NDMS23] Michalis Nikolaou, Georgios Drakopoulos, Phivos Mylonas, and Spyros Sioutas. Intelligent agents with graph mining for link prediction over Neo4j. In *WEBIST*, pages 504–511. SCITEPRESS, 2023.
- [SDK⁺21] Christina Saravanos, Georgios Drakopoulos, Andreas Kanavos, Eleanna Kafeza, and Christos Makris. Discovering influential Twitter authors via clustering and ranking on Apache Storm. In *IISA*. IEEE, 2021.
- [TPDM16] Fotios Tagkalakis, Aimilia Papagiannaki, Georgios Drakopoulos, and Vasileios Megalooikonomou. Augmenting fMRI-generated brain connectivity with temporal information. In *IISA*. IEEE, 2016.
- [VDC⁺21] Yorghos Voutos, Georgios Drakopoulos, Georgios Chrysovitsiotis, Zoi Zachou, Dimitris Kikidis, Efthymios Kyrodimos, and Themis P. Exarchos. Multimodal lip-reading for tracheostomy patients in the Greek language. In *SMAP*. IEEE, 2021.
- [VDC⁺22] Yorghos Voutos, Georgios Drakopoulos, Georgios Chrysovitsiotis, Zoi Zachou, Dimitris Kikidis, Efthymios Kyrodimos, and Themis P. Exarchos. Multimodal lip-reading for tracheostomy patients in the Greek language. *Computers*, 11(3), 2022.
- [VDM19] Yorghos Voutos, Georgios Drakopoulos, and Phivos Mylonas. Smart agriculture: An open field for smart contracts. In *SEEDA-CECNSM*. IEEE, 2019.
- [VDM20] Yorghos Voutos, Georgios Drakopoulos, and Phivos Mylonas. Metadata-enriched discovery of aspect similarities between cultural objects. In *SMAP*. IEEE, 2020.

Appendix A'

World events

Progress is impossible without change; and those who cannot change their minds cannot change anything.
George Bernard Shaw (1856 – 1950)

A'.1 Appendix introduction

THE primary purpose of this appendix is to provide a brief summary of the major world events which took place as the author understood them while writing the research papers mentioned in chapter 6 or this text itself. This serves as a reminder that the world moves on and evolves also outside university labs. These events are listed in tables A'.1 and A'.2. Links to some of these events are included^{1 2 3 4 5 6 7}. Moreover, the movies, books, and research papers in the bibliography section are general recommendations and they have been a major source on inspiration for the author.

¹https://en.wikipedia.org/wiki/Monroe_Doctrine

²<https://edition.cnn.com/2022/10/10/business/william-shatner-new-book-boldly-go-scen/index.html>

³<https://variety.com/2022/tv/news/william-shatner-space-boldly-go-excerpt-1235395113>

⁴<https://time.com/6318343/jimmy-carter-birthday-99-grandson>

⁵<https://www.youtube.com/watch?v=fbvs4fVRaf8>

⁶<https://apnews.com/article/trump-trial-deliberations-jury-testimony-verdict-85558c6d08efb434d05b694364470aa0>

⁷<https://www.govinfo.gov/app/details/GPO-J6-REPORT/context>

Table A'.1: Major events in the world.

Year	Month	Event
2017	March	Nintendo Switch is released
		SpaceX completes the first rocket reflight
		Brexit negotiations start
	May	WannaCry ransomware attack
		Ariana Grande concert ISIS bombing attack
	June	ISIS London bridge attack
	August	<i>The Great American Eclipse</i>
2018	December	US recognizes Jerusalem as Israel's capital
	December	TIME person of the year: Silence breakers
	January	US federal government shutdown
		<i>Blue Blood Moon Eclipse</i>
	March	Sergei and Yulia Skripal poisoning
		Xi Jinping becomes president for life in China
	April	NASA launches Transiting Exoplanet Survey Satellite
	May	ETA is dissolved in Spain
		General Data Protection Regulation (GDPR) is enacted in EU
	June	Treaty of Prespes signed between Hellas and North Macedonia
		Saudi Arabia allows women to drive
	August	Apple is the first public company with a market capitalization of one trillion
	December	Civil unrest in Paris
		Second US federal government shutdown
		TIME person of the year: The guardians
2019	January	Jair Bolsonaro becomes president of Brazil
	March	Venezuela expels the German ambassador
		EU fines Google €1.49 billion over anti-trust violations
	April	Julian Assange is arrested after seven years in the Ecuador embassy in the UK
		Japanese emperor Akihito abdicates from his throne
	May	Gulf of Oman incident
	November	Public impeachment hearings against Donald Trump
	December	First COVID-19 case in Wuhan
2020	December	TIME person of the year: Greta Thunberg
	January	Impeachment trial against Donald Trump begins
		Wuhan is quarantined
		UK and Gibraltar officially withdraw from EU
	March	Yahoo! time capsule from 2006 is opened
	May	Protests after the murder of George Floyd in the US
	July	Prominent Twitter (not X) accounts hacked to promote cryptocurrency schemes
	November	US Presidential elections. Joe Biden wins
	December	TIME person of the year: President Joe Biden, Vice President Kamala Harris
2021	January	J6 insurrection
	April	Russia warns NATO not to send troops to Ukraine
	June	G7 agrees to a global 15% corporate tax
	September	AUKUS trilateral security agreement
		Olaf Scholz is the new German chancellor
	October	Captain Kirk actually flies in space
		Windows 11 are released
2021	December	New Caledonia independence referendum
		TIME person of the year: Elon Musk

Table A'.2: Major events in the world (continued).

Year	Month	Event
2022	January	The five UN Security Council members jointly declare a nuclear won cannot be won
	February	Russia and China express concerns over NATO expansion Russia invades Ukraine
	May	The Event Horizon Telescope captures the supermassive black hole of Milky Way
	October	Elon Musk buys Twitter (not X) for \$44 billion
	December	TIME person of the year: Volodymyr Zelensky, the spirit of Ukraine
2023	February	Russia withdraws from New START agreement with the US Train crash in Tempi, Thessaly, Hellas
	March	OpenAI launches GPT-4 UBS buys Credit Suisse for CHF 3 billion Israeli judicial reform protests
	April	Finland joins NATO Nuclear power in Germany ends after half a century SpaceX's Starship rocket launches. And explodes
	May	First Republic bank in the US fails and is auctioned Elections in Hellas
	June	More elections in Hellas A boat carrying immigrants capsizes off the coast of Peloponnese Titan submersible implosion
	July	EU and the US sign a data communication agreements
	October	President Jimmy Carter turns 99 Coach Giannis Ioannidis passes away <i>Music festival</i> event
	November	First global AI safety summit in London
	December	TIME person of the year: Taylor Swift
2024	January	Egypt, Ethiopia, Iran, and UAE become BRICS members Japan becomes the fifth country to do soft landing on the Moon
	March	Sweden joins NATO AI Act is passed in the EU
	May	Donald Trump is convicted of felony crimes
	June	European elections Twitter (not X) makes <i>like</i> private Mark Hamill reacts to making <i>like</i> private

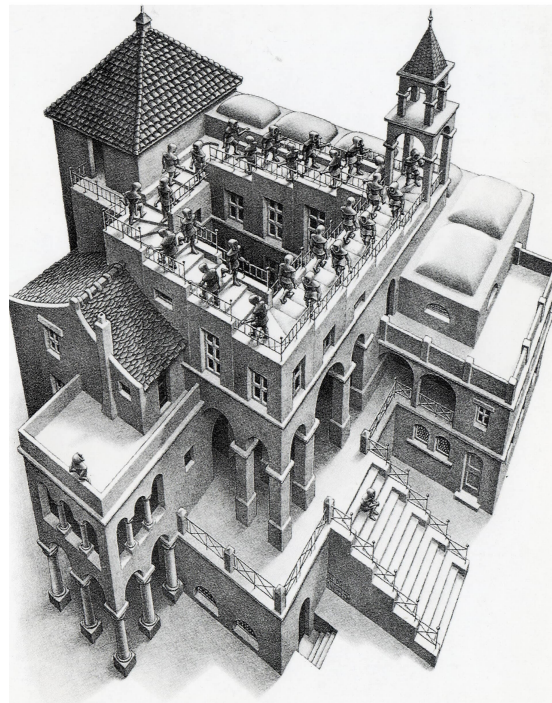


Figure A'.1: Stairs – M.C. Escher.

A'.2 Appendix bibliography

- [20068] 2001: A space Odyssey. <https://www.imdb.com/title/tt0062622/>, 1968.
- [A c71] A clockwork orange. <https://www.imdb.com/title/tt0066921/>, 1971.
- [Abb84] Edwin Abbott. *Flatland*. Seeley and co., 1884.
- [Ali79] Alien. <https://www.imdb.com/title/tt0078748/>, 1979.
- [Ali86] Aliens. <https://www.imdb.com/title/tt0090605/>, 1986.
- [Ap00] American psycho. <https://www.imdb.com/title/tt0144084/>, 2000.
- [Apo79] Apocalypse now. <https://www.imdb.com/title/tt0078788/>, 1979.
- [BGO48] Eric Arthur Blair (George Orwell). 1984. Secker and Warburg, 1948.
- [Bla82] Blade runner. <https://www.imdb.com/title/tt0083658/>, 1982.
- [Bra53] Ray Bradbury. *Fahrenheit 451*. Ballantine books, 1953.
- [Bro96] Broken arrow. <https://www.imdb.com/title/tt0115759/>, 1996.
- [Bro98] Christopher R Browning. *Ordinary Men: Reserve Police Battalion 101 and the Final Solution in Poland*. Penguin books Ltd, 1998.
- [Brz97] Zbigniew Brzezinski. *The grand chessboard: American primacy and its geostrategic imperatives*. Basic Books, 1997.
- [Brz12] Zbigniew Brzezinski. *Strategic Vision: America and the Crisis of Global Power*. Basic books, 2012.
- [BS05] Kai Bird and Martin Jay Sherwin. *American Prometheus: The Triumph and Tragedy of J. Robert Oppenheimer*. Alfred A. Knopf, 2005.
- [BSK82] Richard Bachman (Stephen King). *The running man*. Signet books, 1982.
- [BSM97] Zbigniew Brzezinski, Brent Scowcroft, and Richard Murphy. Differentiated containment. *Foreign Affairs*, pages 20–30, 1997.
- [Car65] Lewis Carroll. *Alice's Adventures in Wonderland*. Macmillan, 1865.
- [Car09] Peter Carlson. *K Blows Top: A Cold War Comic Interlude, Starring Nikita Khrushchev, America's Most Unlikely Tourist*. PublicAffairs, 2009.
- [Cha77] Gregory J Chaitin. Algorithmic information theory. *IBM Journal of research and development*, 21(4):350–359, 1977.
- [CHW75] Michel Crozier, Samuel P Huntington, and Joji Watanuki. *The Crisis of Democracy: On the Governability of Democracies*. NYU University Press, 1975.

- [Cit95] Citizen X. <https://www.imdb.com/title/tt0112681/>, 1995.
- [Cle94] Clerks. <https://www.imdb.com/title/tt0109445/>, 1994.
- [Col04] Collateral. <https://www.imdb.com/title/tt0369339/>, 2004.
- [Con97] Con Air. <https://www.imdb.com/title/tt0118880/>, 1997.
- [Coy87] Harold Coyle. *Team Yankee*. Presidio Press, 1987.
- [Cri69] Michael Crichton. *The Andromeda strain*. Knopf, 1969.
- [Cri95] Crimson tide. <https://www.imdb.com/title/>, 1995.
- [Das81a] Das Boot. <https://www.imdb.com/title/tt0082096/>, 1981.
- [Das81b] Das Experiment. <https://www.imdb.com/title/tt0250258/>, 1981.
- [Dic56] Philip K Dick. Minority report. In *Fantastic universe*. Leo Margulies, 1956.
- [Dic62] Philip K Dick. *The man on the high castle*. Putnam, 1962.
- [DMdgS22] Doctor Mabuse der große Spieler. <https://www.imdb.com/title/tt0013086/>, 1922.
- [Dr.64] Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb. <https://www.imdb.com/title/tt0057012/>, 1964.
- [dT40] Alexis de Tocqueville. *Democracy in America*. Saunders and Otley, 1840.
- [Fig99] Fight club. <https://www.imdb.com/title/tt0137523/>, 1999.
- [Ful87] Full metal jacket. <https://www.imdb.com/title/tt0093058/>, 1987.
- [Gho17] Ghost in the shell. <https://www.imdb.com/title/tt1219827/>, 2017.
- [Gib77] Edward Gibbon. *The History of the Decline and Fall of the Roman Empire*. Strahan and Cadell, 1777.
- [Gib84] William Gibson. *Neuromancer*. Ace, 1984.
- [Gq99] Galaxy quest. <https://www.imdb.com/title/tt0177789/>, 1999.
- [Gra18] David Graeber. *Bullshit jobs: A theory*. Simon and Schuster, 2018.
- [Gro62] Leslie Groves. *Now It Can Be Told: The Story Of The Manhattan Project*. Da Capo Press, 1962.
- [Hac95] Hackers. <https://www.imdb.com/title/tt0113243/>, 1995.
- [Her65] Frank Herbert. *Dune*. Chilton Books, 1965.
- [Hof07] Douglas R Hofstadter. *I am a strange loop*. Basic Books, 2007.
- [Hut96] Samuel P Huntington. *The clash of civilizations and the remaking of world order*. Simon and Schuster, 1996.

- [Hux31] Aldus Huxley. *Brave new world*. Chatto and Windus, 1931.
- [Irr02] Irreversible. <https://www.imdb.com/title/tt0290673/>, 2002.
- [Jd91] Terminator II: Judgment day. <https://www.imdb.com/title/tt0103064/>, 1991.
- [Kaf25] Franz Kafka. *The trial*. Verlag die Schmiede, 1925.
- [Kaf26] Franz Kafka. *The castle*. Kurt Wolff, 1926.
- [Kaf27] Franz Kafka. *Amerika*. Kurt Wolff, 1927.
- [KB86] Donald Ervin Knuth and Duane Bibby. *The T_EXbook*. Addison-Wesley Reading, 1986.
- [Lew16] Michael Lewis. *The Undoing Project: A Friendship That Changed Our Minds*. W. W. Norton, 2016.
- [Mag73] Magnum Force. <https://www.imdb.com/title/tt0070355/>, 1973.
- [Mal95] Mallrats. <https://www.imdb.com/title/tt0113749/>, 1995.
- [Met27] Metropolis. <https://www.imdb.com/title/tt0017136/>, 1927.
- [MN20] Sid Meier and Jennifer Noonan. *Sid Meier's Memoir! A Life in Computer Games*. WW Norton & Company, 2020.
- [Nig47] Nightmare alley. <https://www.imdb.com/title/tt0039661/>, 1947.
- [Nig91] Night on Earth. <https://www.imdb.com/title/tt0102536/>, 1991.
- [Opp23] Oppenheimer. <https://www.imdb.com/title/tt15398776/>, 2023.
- [Pla86] Platoon. <https://www.imdb.com/title/tt0091763/>, 1986.
- [Res92] Reservoir dogs. <https://www.imdb.com/title/tt0105236/>, 1992.
- [Rob87] Robocop. <https://www.imdb.com/title/tt0093870/>, 1987.
- [Rus48] Bertrand Russell. *History of Western philosophy*. George Allen and Unwin Ltd, 1948.
- [RWB43] Arturo Rosenblueth, Norbert Wiener, and Julian Bigelow. Behavior, purpose and teleology. *Philosophy of science*, 10(1):18–24, 1943.
- [Sd95] Strange days. <https://www.imdb.com/title/tt0114558/>, 1995.
- [Se795] Se7en. <https://www.imdb.com/title/tt0114369/>, 1995.
- [Shi60] William L Shirer. *The Rise and Fall of the Third Reich: A History of Nazi Germany*. Simon and Schuster, 1960.
- [SJ60] Arthur Schlesinger Jr. *The politics of upheaval*. Houghton Mifflin, 1960.
- [SJ99] Arthur Schlesinger Jr. *The cycles of American history*. Mariner books, 1999.

- [SJ02] Arthur Schlesinger Jr. *A Thousand Days: John F. Kennedy in the White House*. Houghton Mifflin, 2002.
- [SJ04] Arthur Schlesinger Jr. *The imperial presidency*. Houghton Mifflin, 2004.
- [Sol72] Solaris. <https://www.imdb.com/title/tt0069293/>, 1972.
- [SP98] Uwe Schöning and Randall Prium. *Gems of theoretical computer science*. Springer, 1998.
- [Sta79] Stalker. <https://www.imdb.com/title/tt0079944/>, 1979.
- [StTmp79] Star trek: The motion picture. <https://www.imdb.com/title/tt0079945/>, 1979.
- [Taj50] The asphalt jungle. <https://www.imdb.com/title/tt0090605/>, 1950.
- [Tc69] The cremator. <https://www.imdb.com/title/tt0063633/>, 1969.
- [TDf86] The Delta force. <https://www.imdb.com/title/tt0090927/>, 1986.
- [TEN20] TENET. <https://www.imdb.com/title/tt6723592/>, 2020.
- [Ter84] Terminator. <https://www.imdb.com/title/tt0088247/>, 1984.
- [The40] The great dictator. <https://www.imdb.com/title/tt0032553/>, 1940.
- [The68] The party. <https://www.imdb.com/title/tt0063415/>, 1968.
- [The88] They live. <https://www.imdb.com/title/tt0096256/>, 1988.
- [The95a] The net. <https://www.imdb.com/title/tt0113957/>, 1995.
- [The95b] The usual suspects. <https://www.imdb.com/title/tt0114814/>, 1995.
- [The98] The big Lebowski. <https://www.imdb.com/title/tt0118715/>, 1998.
- [The99] The thirteenth floor. <https://www.imdb.com/title/tt0139809/>, 1999.
- [ThfRO90] The hunt for Red October. <https://www.imdb.com/title/tt0099810/>, 1990.
- [Tm95] Twelve monkeys. <https://www.imdb.com/title/tt0114746/>, 1995.
- [TmwsLV62] The man who shot Liberty Valance. <https://www.imdb.com/title/tt0056217/>, 1962.
- [Twe57] Twelve angry men. <https://www.imdb.com/title/tt0050083/>, 1957.
- [Twe97] Twelve angry men. <https://www.imdb.com/title/tt0118528/>, 1997.
- [War83] War Games. <https://www.imdb.com/title/tt0086567/>, 1983.
- [Wie49] Norbert Wiener. *Extrapolation, interpolation, and smoothing of stationary time series with engineering applications*. MIT press, 1949.
- [Wie54] Norbert Wiener. *The human use of human beings*. Houghton Mifflin, 2nd edition, 1954.
- [Wie66] Norbert Wiener. *Nonlinear problems in random theory*. MIT press, 1966.
- [Wil69] Wild bunch. <https://www.imdb.com/title/tt0065214/>, 1969.
- [Win99] Wing commander. <https://www.imdb.com/title/tt0131646/>, 1999.

This page has been intentionally left blank

This page has been intentionally left blank

Index

- API, 6, 90
- ASCII art, 83
- autoencoder, 38, 39

- behavioral sciences, 85, 89
- best matching neuron, *see* BMN
- BMN, 38
- Boolean semiring, 79, 87
- BSON, 57
- Butterworth filter, 55

- Caffe, 6
- clustering, 39
- CNN, 4, 6, 7, 86
- cognitive maps, *see* SOM
- computational complexity, 78
- convolutional neural networks, *see* CNN
- coordinate space, 37
- Cypher, 83

- data
 - higher order, 6
 - humanistic, xiv–xvi, xx, 5, 38, 54, 78, 80, 85–89, 91
 - linked, xvi
 - proximity, 4, 38
 - space, 37, 81
- DCT2, 5, 44
- deep learning, *see* DL
- digital awareness, xviii, 78, 82
- discrete cosine transform, *see* DCT2
- DL, 3, 4, 39, 58, 85–87
- DSP, 80, 91

- emotion wheel, 59
- explainable AI, 82

- expressive power, *see* modelling power

- Fast Fourier Transform, *see* FFT
- feedforward neural networks, *see* FFNN
- FFNN, 6, 16
- FFT, 56
- FFT West, *see* FFTW
- FFTW, 56

- GAN, 38
- generative adversarial networks, *see* GAN
- graph, xvi, xvii, 4, 5, 91
 - adjacency
 - matrix, 5, 56, 87
 - tensor, xxi, 88
 - clustering, 39, 56
 - clustering coefficient, 16
 - compression, xxii, 5
 - Estrada index, 16, 56
 - evolving, xvii, 7, 54, 79, 81
 - filter, 58
 - flitering, 56
 - Fourier transform, 56
 - geometry, xvi
 - GNN, 38, 54, 57, 58, 85–87, 91
 - GSP, 55–58
 - humanistic, 81
 - knowledge mining, xvii
 - Laplace transform, 56
 - mining, xvi, 5, 57, 79, 81, 85–89
 - ML, 81
 - neural networks, *see* GNN
 - odd Estrada index, 16
 - recursion, xvii
 - resilience, 5, 7, 16
 - self similarity, xvii

- structural coherency, *see* graph resilience
- topology, xvi, 4, 7, 43, 57
- hashtag, xx
- Hebbian learning rule, 38
- Hilbert spaces, xvi
- hypergraph, 6
- Industry 4.0, xiv, xviii
- Internet of Things, *see* IoT
- IoT, xiv, 7
- IR, 6
- Java, 79, 88
 - virtual machine, *see* JVM
- JSON, 57
- JSON-LD, 57
- Julia, 6, 87, 90
- JVM, 6
- keras, 6
- knowledge mining, xvi, xxi, 78
- knowledge nining, xvi
- large language model, 59
- least mean squares, *see* LMS
- Lindenmayer system, xvii
- linear algebra, xvi, 80
- LLM, 59
- LMS, 5
- long short term memory, *see* LSTM
- LS, 79
- LSTM, 7, 39, 57
- machine learning, *see* ML
- Markov chain, 4
- MatConvNet, 6
- max-flow-min-cut theorem, 58
- MIMO, 80
- ML, xxi, 3, 39, 81
- MLDA, *see* TDA
- MLlib, 6
- MLP, 38
- modelling power, 78
- multilayer graphs, 88
- multilayer perceptron, 4
- multilinear
 - discriminant analysis, *see* TDA
- Neo4j, xvi, 79, 83, 88, 90
- neuroimaging, 38, 39
- NoSQL, 90
- numpy, 6
- pattern recognition, xxi
- polar decomposition, 87
- polar factorization, 91
- portable document format, xxv
- PR, 4
- Python, 6, 90
- PyTorch, 90
- random variable, xxiv
- reasoners, xviii
- recurrent neural networks, *see* RNN
- Reddit, xviii
- RNN, 4, 7, 86
- self organizing maps, *see* SOM
- single sign on services, *see* SSOS
- singular value decomposition, *see* SVD
- Sierpinski gasket, xvii
- SNA, xviii, xxii, 5, 7, 57, 79, 82, 85, 86, 89
- social media, xiv, 59
- social network analysis, *see* SNA
- SOM, xxv, 4, 7, 37–40, 42, 43, 45, 79, 81
- space complexity, 78
- spatial filter, 55
- spiking neural networks, 7
- SSOS, xviii
- SVD, 6, 86
- Taylor series, 80
- tensor, xxiv, 4, 5, 39, 80, 90, 91
 - algebra, xxi, 80
 - clustering, 81, 89
 - completion, 6
 - decomposition, 6, 86
 - dimension, 6
 - discriminant analysis, *see* TDA
 - distance metric, 4, 38, 39, 81, 85, 87
 - factorization, 81
 - fiber, 6, 88
 - Kruskal factorization, 6

- mode, 6, 39, 91
- norm, 6
 - Frobenius, 6
- processing units, *see* TPU
- signal processing, 5, *see* TeSP
- stack network, *see* TSN
- TDA, 6
- TeSP, 6
- TPU, 6, 81
- TSN, 4, 5, 16, 79, 81, 86
- Tucker decomposition, 6
- TensorFlow, 6, 90
- TensorLab, 6
- TeSP, 91
- Theano, 6
- Tikhonov regularization, 6
- Torch7, 6
- TSN, 88, 91
- Twitter, xviii, 5, 87–89
- universal emotion theory, 59
- Volterra series, 80
- wavelet, 39
- Zipf law, xviii
- άλγεβρα τανυστών, xxxii
- δεδομένα ανωτέρας τάξεως, xxxii
- εξόρυξη γνώσεως, xxxii
- πλειογραμμική άλγεβρα, xxxii