Topological Interpretation of Interactive Computation

Emanuela Merelli
University of Camerino
Virtual Workshop on Interactive Computation

Joint work with Anita Wasiwleska and Mario Rasetti

Future Tech Week - 27 September 2019

Interactive computation

A *sequential* **interactive computation** is a Turing machine (TM) computation that continuously interacts with its **environment** by alternately accepting an ***input string*** on the *input-tape* and computing on the *work-tape* a corresponding output string to be delivered on the *output-tape*.

An interactive computations can be modelled as a sequence **possibly infinite**, of non-deterministic 3-tape TMs.

Persistent Turing machine

In 2004, Scott A. Smolka worked with Dina Goldin and colleagues on a formal framework for interactive computing.

The **persistent Turing machine (PTM)** is a 3-tape Turing machine dealing with *persistent sequential interactive computations*.

Persistent Turing machine at work

A PTM is a 3-tapes TM that performs an **infinite sequence** of classical computation. It is based on dynamic stream semantics. A computation is called **sequential interactive computation** because it continuously interacts with its environment by alternately accepting an input string on the input-tape and computing on the work-tape a corresponding output string to be delivered on the output-tape. The computation is persistent, meaning that the content of the work-tape persists from one computation step to the next by ensuring a memory function.

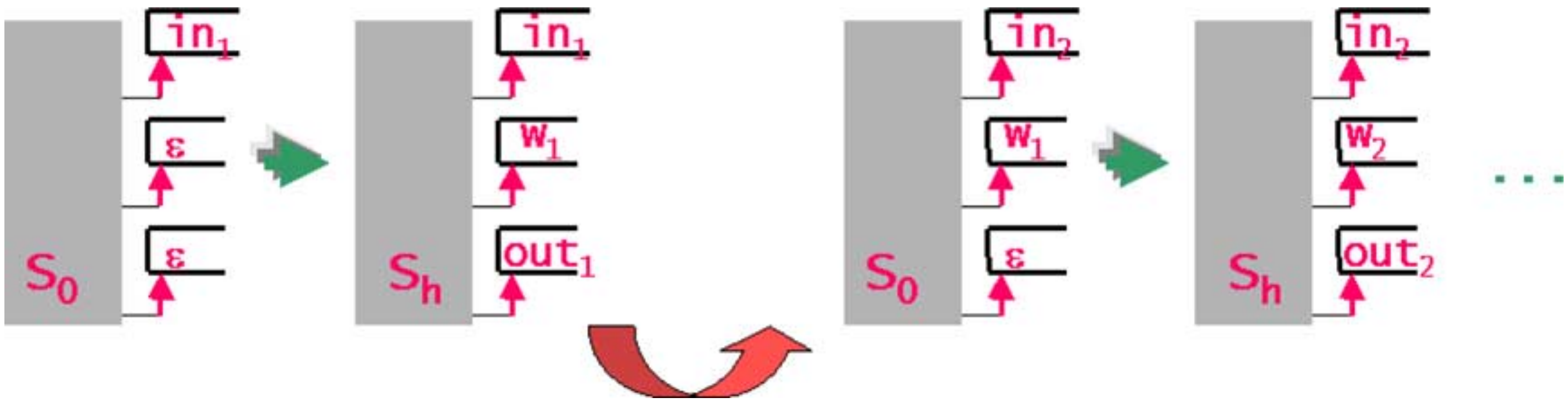


Fig. 1. Illustration of PTM macrosteps.

Preface

Interaction was a common theme in the research of Paris Kanellakis. His doctoral dissertation explored the computational complexity of concurrency control in distributed databases. Later, his research interests included object-oriented and constraint databases, complexity issues in process algebra and other formalisms for concurrent systems, and fault-tolerant parallel algorithms. Given that interaction is a hallmark of each of these areas and that the second author (Smolka) was Paris's first Ph.D. student and the first author (Goldin) was one of his last Ph.D. students, a paper on a formal framework for interactive computing seems appropriate for the special issue of *Information and Computation* commemorating the anniversary of Paris's 50th birthday. A preliminary version of this paper appeared in [1].

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Information and Computation

www.elsevier.com/locate/ic

ELSEVIER

Information and Computation 194 (2004) 101–128

Turing machines, transition systems, and interaction

Dina Q. Goldin,[a,*] Scott A. Smolka,[b] Paul C. Attie,[c] Elaine L. Sonderegger[a]

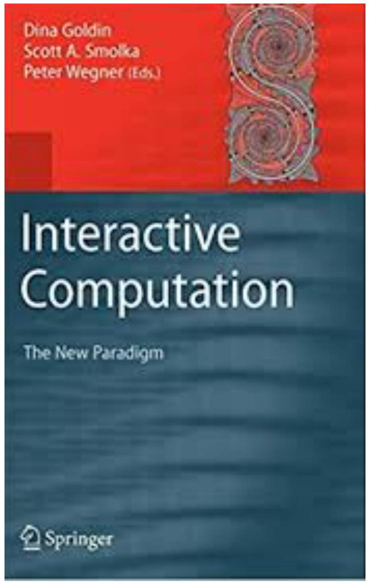
[a] *Computer Science and Engineering Department, University of Connecticut, Storrs, CT 06269, USA*
[b] *Department of Computer Science, SUNY at Stony Brook, Stony Brook, NY 11794, USA*
[c] *College of Computer Science, Northeastern University, Boston, MA 02115, USA*

Received 24 September 2003; revised 24 March 2004
Available online 22 September 2004

Abstract

This paper presents *persistent Turing machines* (PTMs), a new way of interpreting Turing-machine computation, based on dynamic stream semantics. A PTM is a Turing machine that performs an infinite sequence of "normal" Turing machine computations, where each such computation starts when the PTM reads an input from its input tape and ends when the PTM produces an output on its output tape. The PTM has an additional worktape, which *retains* its content from one computation to the next; this is what we mean by *persistence*. A number of results are presented for this model, including a proof that the class of PTMs is isomorphic to a general class of effective transition systems called *interactive transition systems*; and a proof that PTMs without persistence (*amnesic PTMs*) are less expressive than PTMs. As an analogue of the Church-Turing hypothesis which relates Turing machines to algorithmic computation, it is hypothesized that PTMs capture the intuitive notion of *sequential interactive computation*.

Keywords: Models of interactive computation; Persistent Turing machine; Persistent stream language; Interactive transition system; Sequential interactive computation



The definition of *PTM* was based on Peter Wegner's *interaction theory* developed to embody distributed network programming.

Interaction is more powerful than rule-based algorithms for computer problem solving, overturning the prevailing view that all computing is expressible as algorithms [4,5].

4. P. Wegner. Why Intera. is More P Than Algorit. CACM, Vol. 40, No.5, ACM, 1997.
5. P. Wegner. Interactive foundations of computing. TCS, Vol.192, Elsevier, 1998.

Since in this framework interactions are more powerful than rules-based algorithms they are not expressible by an initial state described in a *finite terms*. Therefore, one of the four Robin Gandy's principles (or constraints) for computability is violated, as stated in [6]. The need to relax such constraints allows one to think that interactive systems might have a richer behavior than algorithms, or that algorithms should be seen from a different perspective. Although *PTM* makes the first effort to build a *TM* that accepts *infinite input*, we strongly support the idea that the interaction model should also include the formal characterization of the notion of *environment*.

6. R.O Gandy. Churchs Thesis and Principles for Mechanisms. J. Barwise, H. J. Keisler and K. Kunen, eds, The Kleene Symposium, North-Holland Publishing Company, 1980.

The definition of *PTM* was based on Peter Wegner's *interaction theory* developed to embody distributed network programming.

Interaction is more powerful than rule-based algorithms for computer problem solving, overturning the prevailing view that all computing is expressible as algorithms [4,5].

4. P. Wegner. Why Intera. is More P Than Algorit. CACM, Vol. 40, No.5, ACM, 1997.
5. P. Wegner. Interactive foundations of computing. TCS, Vol.192, Elsevier, 1998.

Smolka Thesis 1 *Any sequential interactive computation can be performed by a PTM.*

Since in this framework interactions are more powerful than rules-based algorithms they are not expressible by an initial state described in a *finite terms*. Therefore, one of the four Robin Gandy's principles (or constraints) for computability is violated, as stated in [6]. The need to relax such constraints allows one to think that interactive systems might have a richer behavior than algorithms, or that algorithms should be seen from a different perspective. Although *PTM* makes the first effort to build a *TM* that accepts *infinite input*, we strongly support the idea that the interaction model should also include the formal characterization of the notion of *environment*.

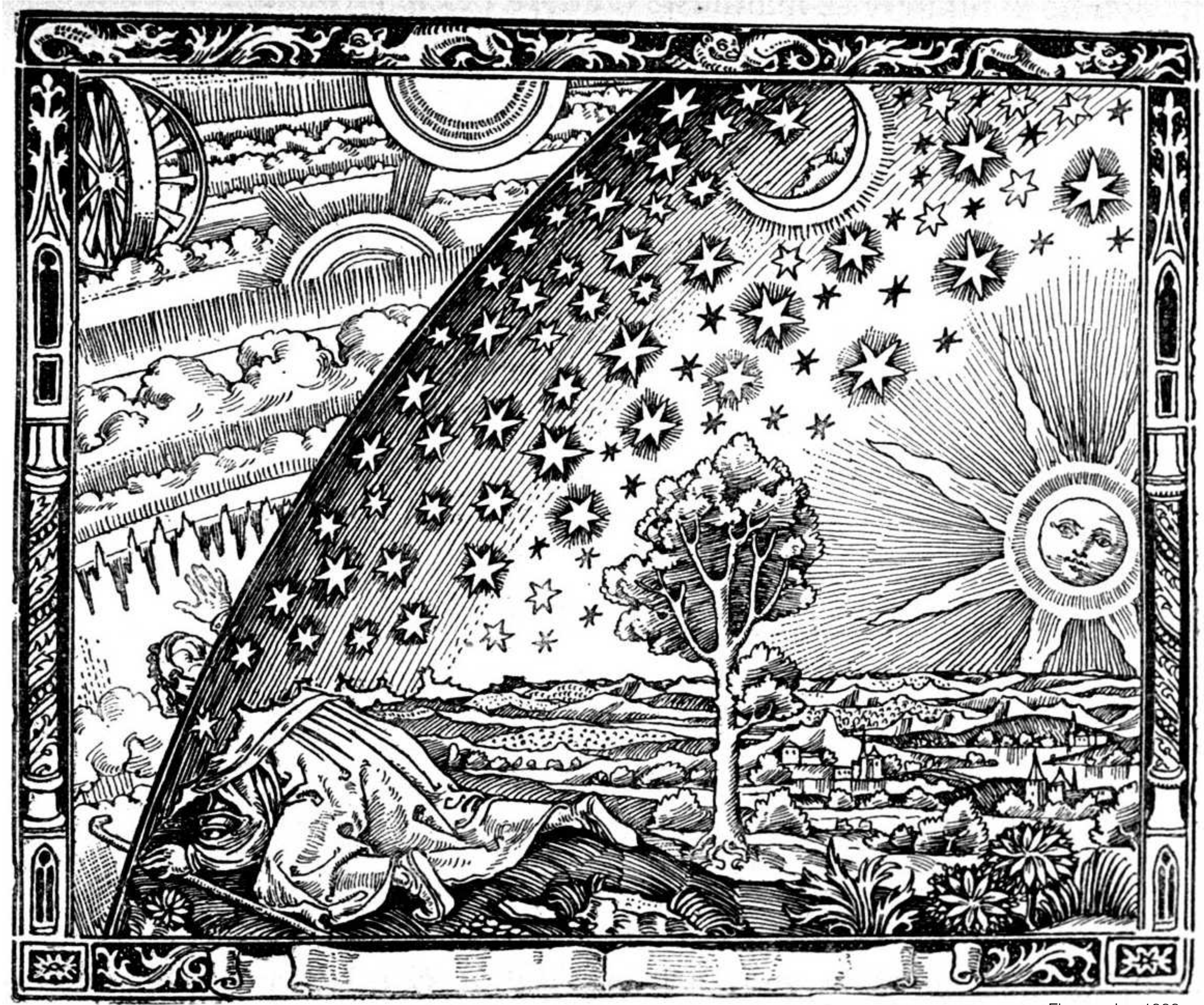
6. R.O Gandy. Churchs Thesis and Principles for Mechanisms. J. Barwise, H. J. Keisler and K. Kunen, eds, The Kleene Symposium, North-Holland Publishing Company, 1980.

We are living in a not flat world

that for some aspects it is unknown

...

and perhaps some of its symmetries have been missed in some theory



Flammarion 1888

Topological *Persistent* Turing Machine

Topological Turing machine (TTM) extends the PTM with the notion of topological environment a global space constructed over a set of possible TM's configurations.

Is TTM a potential universal model for interactive computation and possibly a new model for concurrent computation?

Topology and Persistent Homology

Topology is the geometry of a shape, it deals with with **qualitative geometric information of a space**, such as connectivity, classification of loops and higher dimensional manifolds, invariants.

Algebraic topology is a branch of mathematics that uses algebraic tools to study **topological spaces**, a set of points and for each point a set **neighbourhoods**, both satisfying a set of axioms.
Its goals is to find algebraic **invariants** that classify topological spaces up to some **homeomorphism** or **homotopy equivalences**.

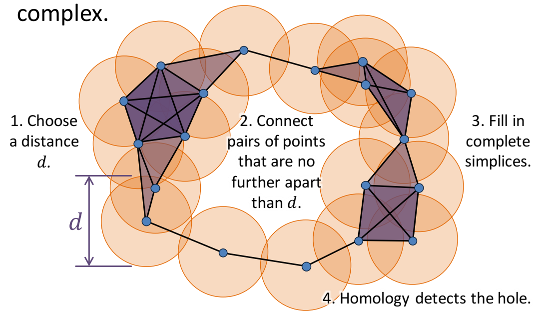
In a discrete setting a full information about topological spaces is inherent in their **simplicial** representation, a piece-wise linear, combinatorially complete, discrete realization of functoriality.

Persistent homology is an algebraic method for discerning topological features of space of data

e.g. components, graph structure holes

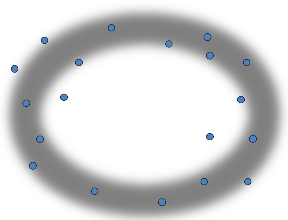
set of discreet points

Idea: Connect nearby points, build a simplicial complex.

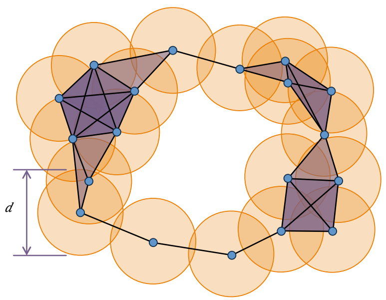


movie by Matthew L. Wright

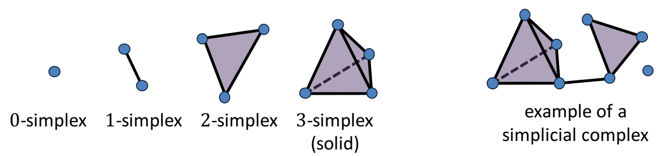
Example: What is the shape of the data?



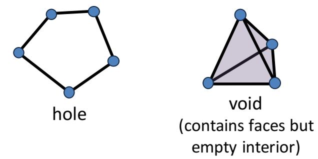
Problem: Discrete points have trivial topology.



A simplicial complex built from points, edges, triangular faces, etc.



Homology counts components, holds, voids, etc.



Homology of a simplicial complex is computable via linear algebra.

Emanuela Merelli, University of Camerino

Topology driven Methods for Complex Systems

topdrim

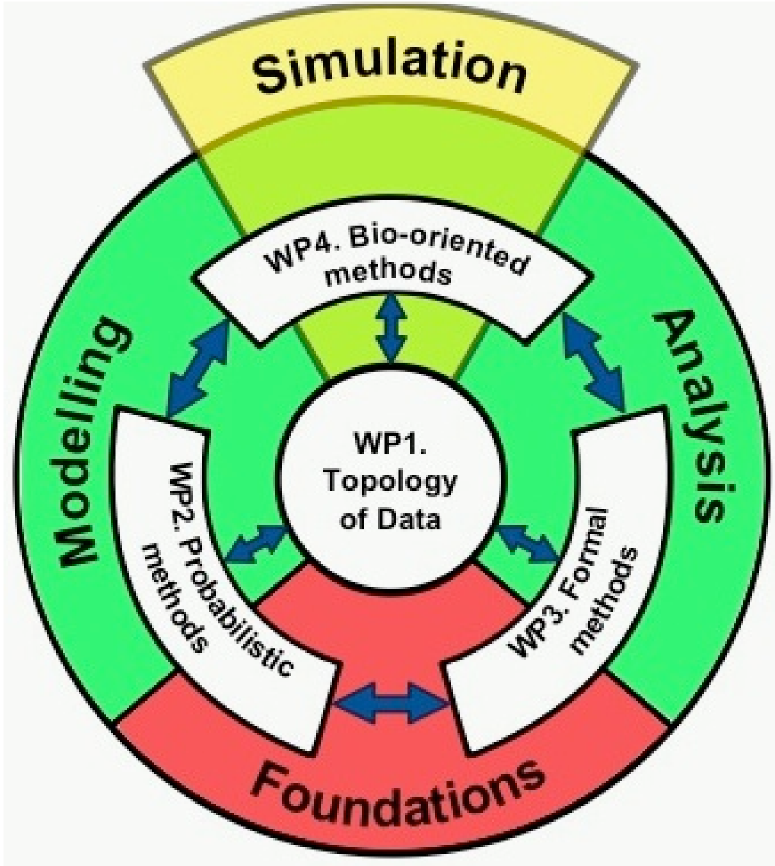
2012-2015

~3ml euros

www.topdrim.eu

The Dream

Taming BIG DATA with Topology
the geometry of 'shapes'



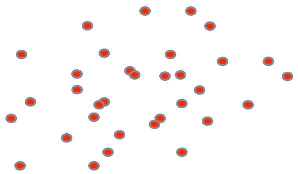
University of Camerino Emanuela Merelli & Stefano Mancini	Camerino, IT	coordinator
University of Southern Denmark Christian Reidys	Odense, DK	
University of Amsterdam Peter Sloot	Amsterdam, NL	
Open University Jeffrey Johnson	Milton Keynes , UK	
ISI Foundation Mario Rasetti & Francesco Vaccarino	Torino, IT	
Aix-Marseille University Marco Pettini	Marseille, F	

The Consortium

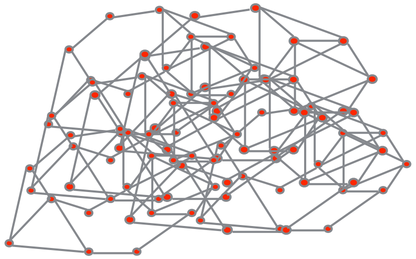
The TOPDRIM objective

The study of new maths for understanding
the dynamics of complex systems

Dataset

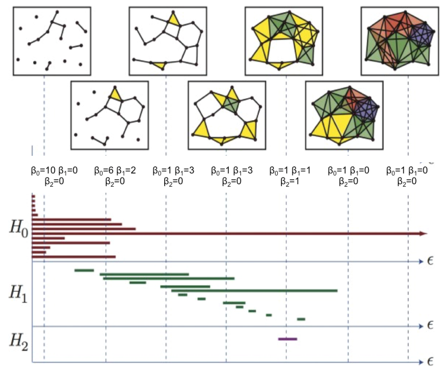


Data Point Cloud



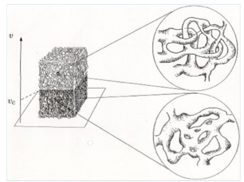
Complex Networks

Time series

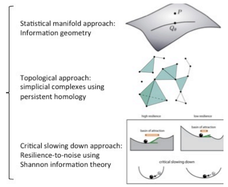


Persistent homology

Probabilistic Methods

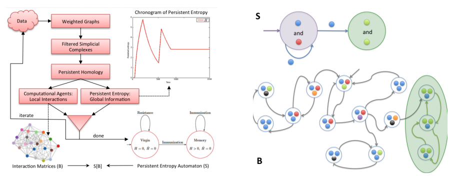


Geometric Entropy



Persistent Entropy

Formal Methods

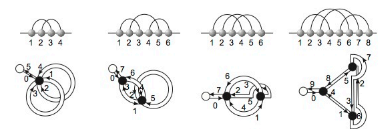


Persistent Entropy Automaton

Bioinspired Methods

Lemma

There are only four irreducible shapes of genus 1.



Topological language for RNA Shapes

Results

Innovative methodology for data mining

Topological Algorithmic Field Theory of Data

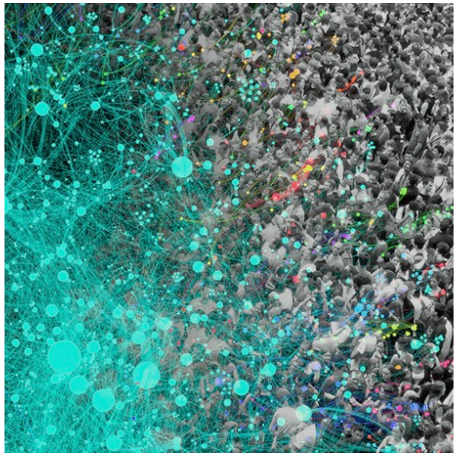
- Topological data analysis (TDA)
- Topological fields theory
- Formal languages

Applications

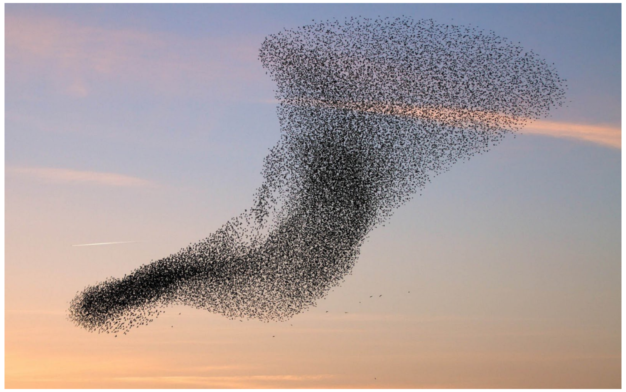
- neuroscience
- epidemiology
- life science
- financial
- robotics

Emanuela Merelli, University of Camerino

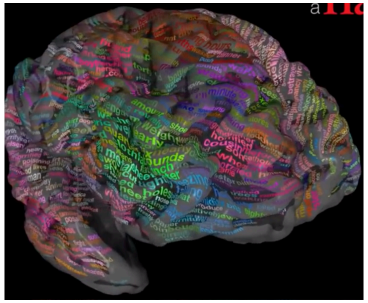
Topology and complex systems



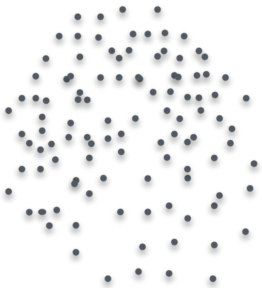
1. to go ***beyond*** Complex Networks and graph theory, beyond structural and functional connectivity of networks



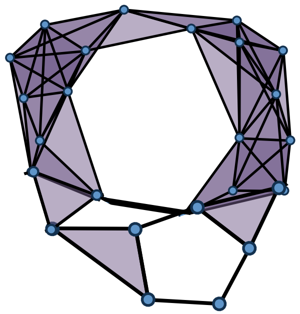
2. to exploit the potential of ***Simplicial Complexes*** and topology theory (Persistent Homology) for studying n-dimensional objects (n >= 2) and many-body interactions that characterize the emerging behaviour of complex systems



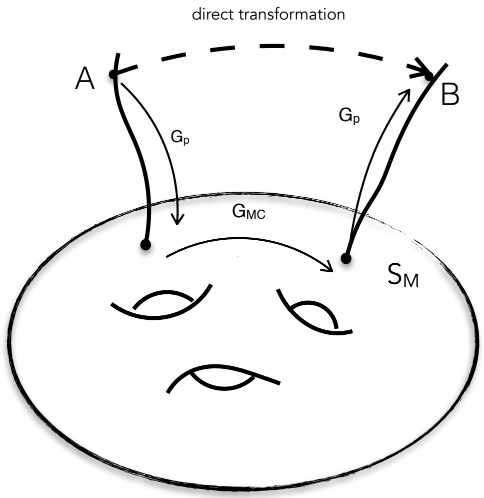
3. to find ***correlation patterns*** hidden in a data space whose dimension justifies the use of Topological Data Analysis



data space



simplicial complex
base space



fiber bundle + field action
relation patterns
topological data field

COMPLEX SYSTEMS

Complex systems are composed of many non-identical elements, **entangled in loops of nonlinear interactions**, and characterized by the typical 'emergent' behaviour, a non-trivial superstructure that cannot be reconstructed by a reductionist approach.

M. Rasetti, E. Merelli Topological field theory of data: mining data beyond complex networks, Cambridge University Press, 2016

Topological algorithmic Field Theory of Data

The TFTD consists of four main steps:

1. embedding data space into a combinatorial topological object, a simplicial complex;

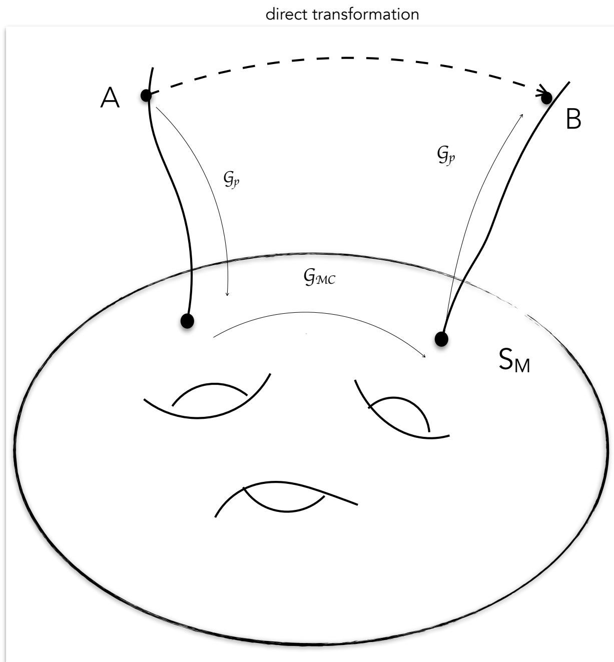
2. considering the **complex as base space** of a **(block)** fiber bundle;

3. assuming a field action**,** which has a free part, the combinatorial Laplacian over the simplicial complex, and an interaction part depending on the process algebra;

4. constructing the gauge group as semi-direct product of the group generated by the algebra of processes (the fibers) and the group of (simplecio-morphisms modulo isotopy) of the data space.

Emergent features of data-represented complex systems were shown to be expressed by the correlation functions of the field theory.

Direct Transformation



fiber bundle + field action

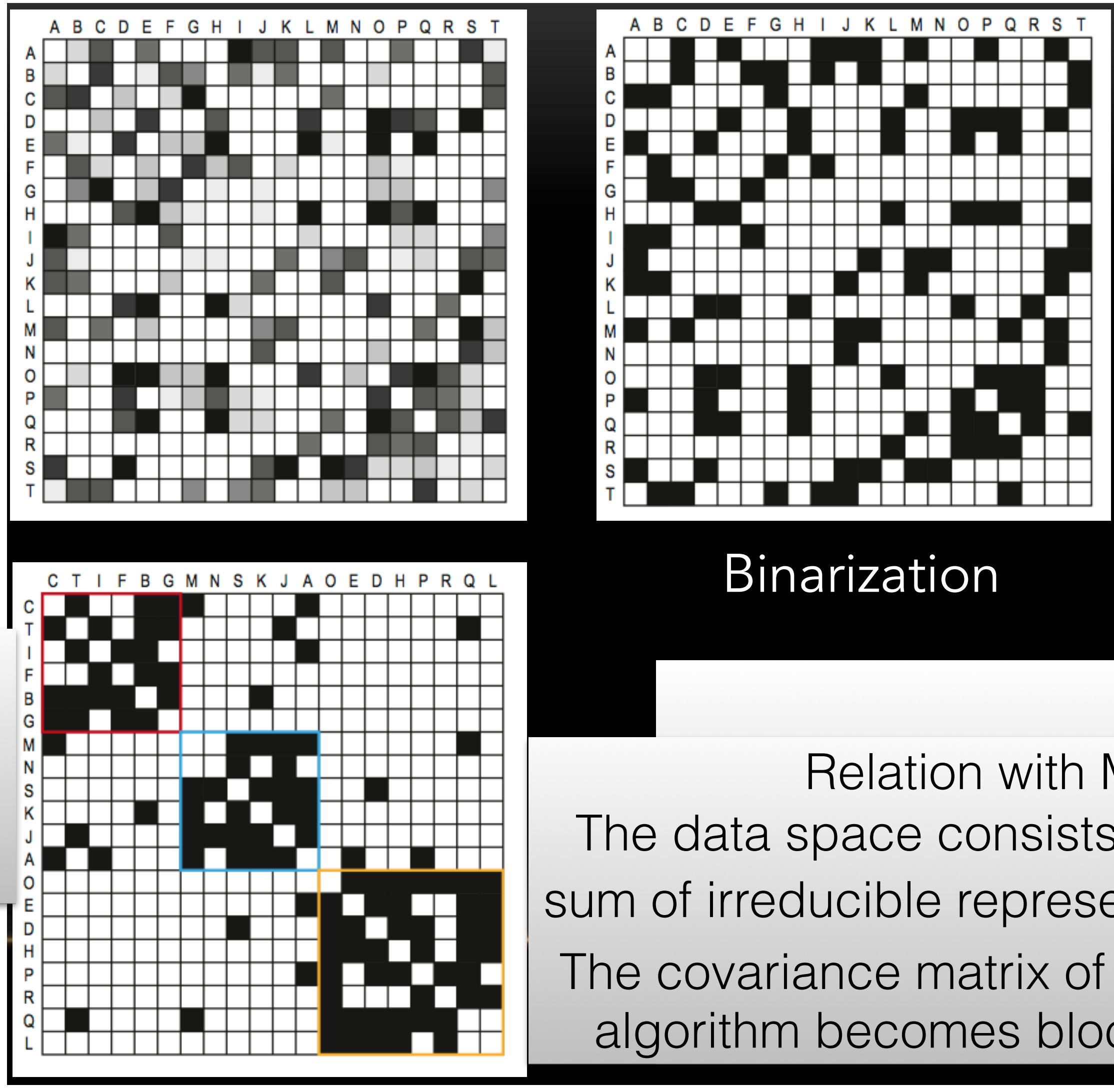
relation patterns

topological data field

M. Rasetti and E. Merelli. *Topological field theory of data: mining data beyond complex networks*, Cambridge University Press, 2016

Topological algorithmic field theory of data at work

Connectivity intensity (grey shadows)



Toy Model

Reordering and modularisation by harmonic analysis over $\mathcal{G}$

Relation with ML
The data space consists of the direct sum of irreducible representations of $\mathcal{G}$.
The covariance matrix of a generic ML algorithm becomes block diagonal.

By courtesy of the ISI foundation

Topological algorithmic field theory of data: an interdisciplinary research program

7th International Workshop DICE2014 Spacetime – Matter – Quantum Mechanics IOP Publishing
Journal of Physics: Conference Series **626** (2015) 012005 doi:10.1088/1742-6596/626/1/012005

The Topological Field Theory of Data: a program towards a novel strategy for data mining through data language

M Rasetti [1] and E Merelli [2]
[1] ISI Foundation, Via Alassio 11-C, 10126 Torino (Italy)
[2] School of Science and Technology, University of Camerino, Via del Bastione 1, 62032 Camerino (Italy)
E-mail: rasetti@isi.it ; emanuela.merelli@unicam.it

Abstract. This paper aims to challenge the current thinking in IT for the 'Big Data' question, proposing – almost verbatim, with no formulas – a program aiming to construct an innovative methodology to perform data analytics in a way that returns an automaton as a recognizer of the data language: a *Field Theory of Data*. We suggest to build, directly out of probing data space, a theoretical framework enabling us to extract the manifold hidden relations (patterns) that exist among data, as correlations depending on the semantics generated by the mining context. The program, that is grounded in the recent innovative ways of integrating data into a topological setting, proposes the realization of a Topological Field Theory of Data, transferring and generalizing to the space of data notions inspired by physical (topological) field theories and harnesses the theory of formal languages to define the potential semantics necessary to understand the emerging patterns.

Nat Comput (2015) 14:421–430
DOI 10.1007/s11047-014-9436-7

CrossMark

Topology driven modeling: the IS metaphor

Emanuela Merelli · Marco Pettini · Mario Rasetti

Published online: 24 June 2014

Abstract In order to define a new method for analyzing the immune system within the realm of Big Data, we bear on the metaphor provided by an extension of Parisi's model, based on a mean field approach. The novelty is the multilinearity of the couplings in the configurational variables. This peculiarity allows us to compare the partition function Z with a particular functor of topological field theory—the generating function of the Betti numbers of the state manifold of the system—which contains the same global information of the system configurations and of the data set representing them. The comparison between the Betti numbers of the model and the real Betti numbers obtained from the topological analysis of phenomenological data, is expected to discover hidden *n-ary relations* among idiotypes and anti-idiotypes. The data topological analysis will select global features, reducible neither to a mere subgraph nor to a metric or vector space. How the immune system reacts, how it evolves, how it responds to stimuli is the result of an interaction that took place among many entities constrained in specific configurations which are relational. Within this metaphor, the proposed method turns out to be a global topological application of the S[B] paradigm for modeling complex systems.

Keywords Immune system · Multilinear mean field · Pattern discovery · Complex systems · Adaptive models · S[B] paradigm · Topology of data · Betti numbers · Big Data

1 Introduction

The objective pursued in this note is to frame the research on the immune system as part of *data science*. Such research is naturally complex and articulated and our contribution intends to be here along the lines of seeing it as a viable candidate for topological data analytics and an example of the S[B] paradigm for modeling complex systems. We recall that data science is the practice to deriving valuable insights from data by challenging all the issues related to the processing of very large data sets, while Big Data is jargon to indicate such a large collection of data (for example, exabytes) characterized by high-dimensionality, redundancy, and noise. The analysis of Big Data requires handling high-dimensional vectors capable of weaning out the unimportant, redundant coordinates. The notion of data space, its geometry and topology are the

Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Computer Science 13 (2014) 90–99
Procedia Computer Science
www.elsevier.com/locate/procedia
ELSEVIER

International Conference on Computational Science, ICCS 2013

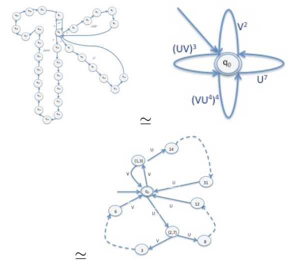
Non locality, topology, formal languages: new global tools to handle large data sets

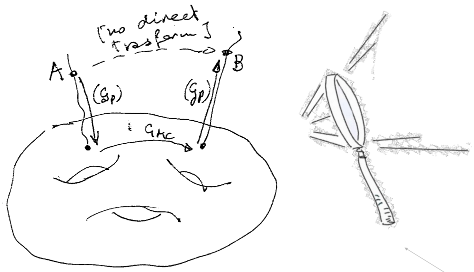
Emanuela Merelli[a,*], Mario Rasetti[b]
[a] School of Science and Technology, University of Camerino, Via del Bastione,1, Camerino 62032, italy
[b] ISI Foundation, Via Alassio 11-C, Torino 10126, Italy

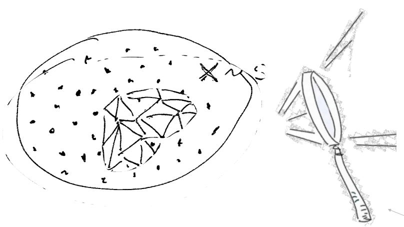
Abstract
The basic idea that stems out of this work is that large sets of data can be handled through an organized set of mathematical and computational tools rooted in a global geometric vision of data space allowing to explore the structure and hidden information patterns thereof. Based on this perspective, the objective is naturally that of discovering and letting emerge, directly from probing the data space, the manifold hidden relations (patterns), e.g. correlations among facts, interactions among entities, relations among concepts and formally describing, in a semantic mining context, the discovered information. In this note, we propose an approach that exploits topological methods for classifying global information into equivalence classes and regular languages for describing by the corresponding automata the hidden features of complex systems.

Keywords: Topology of data; Mapping Class Group; Formal Language; Complex systems.

$$L \sim G_{168} = \langle U, V | V^2, (UV)^3, U^7, (VU^4)^4 \rangle.$$







automaton

$$\mathcal{G} = G_{\mathcal{MC}} \wedge G_{\mathcal{P}}$$

data field

behavioural semantics

big data collections

contextual semantics

$\mathcal{G}$: Semantics —> Syntax

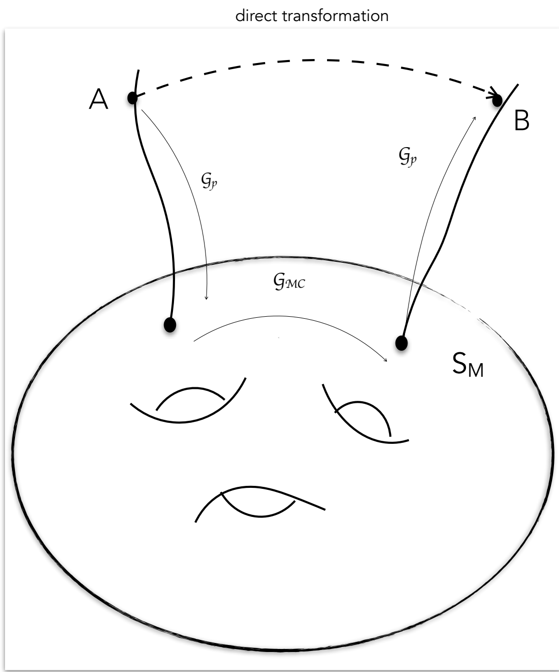
TFTD for Automata-based Learning

The learning process is articulated in three pillars:

i) persistent **homology methods,** to extract, from a simplicial complex representation of the manifolds embedded in data-space, those correlation patterns that encode deep level information;

ii) topological **field theory** of data, able to extract all characteristic information about such patterns, non-linearly tangled in the data set, in a way that – just in view of nonlinearity – is expected also to feedback the re-organization of the data set itself (the gauge group);

iii) theory of **formal languages,** enabling us to express the semantics of the transformations implied by the field dynamics into automata-based learning processes.



The learning framework

Classical Notion of Process

- process is the **behavior** of system
- a process domain is the **context** to interpret a process
- a behaviour can be described as **action relations**
- a process performs actions and **interacts** with other processes through its **environment**

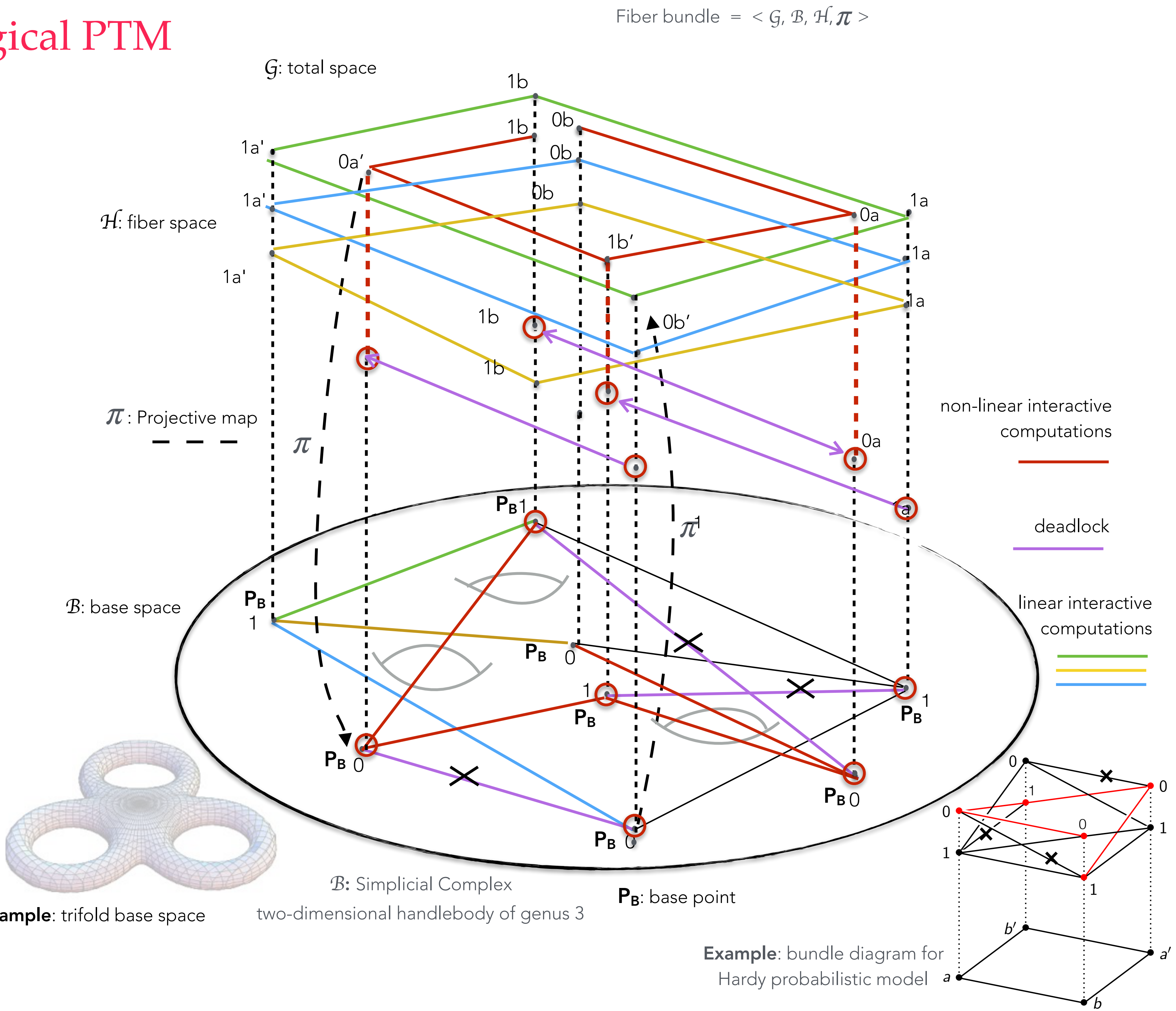
Topological Persistent Turing Machine:
the algorithmic aspect of TFTD
the algorithm as a process

A **topological Turing machine (TTM)** is a model of computation able to describe both the **behaviour** of interactive machines, *its processes*, and the computational environment, the **context** to interpret a process, *its process domain*.

Topological *Persistent* Turing Machine

Topological Turing machine (TTM) extends the PTM with the notion of topological environment, a global space to interpret its processes.

Topological PTM

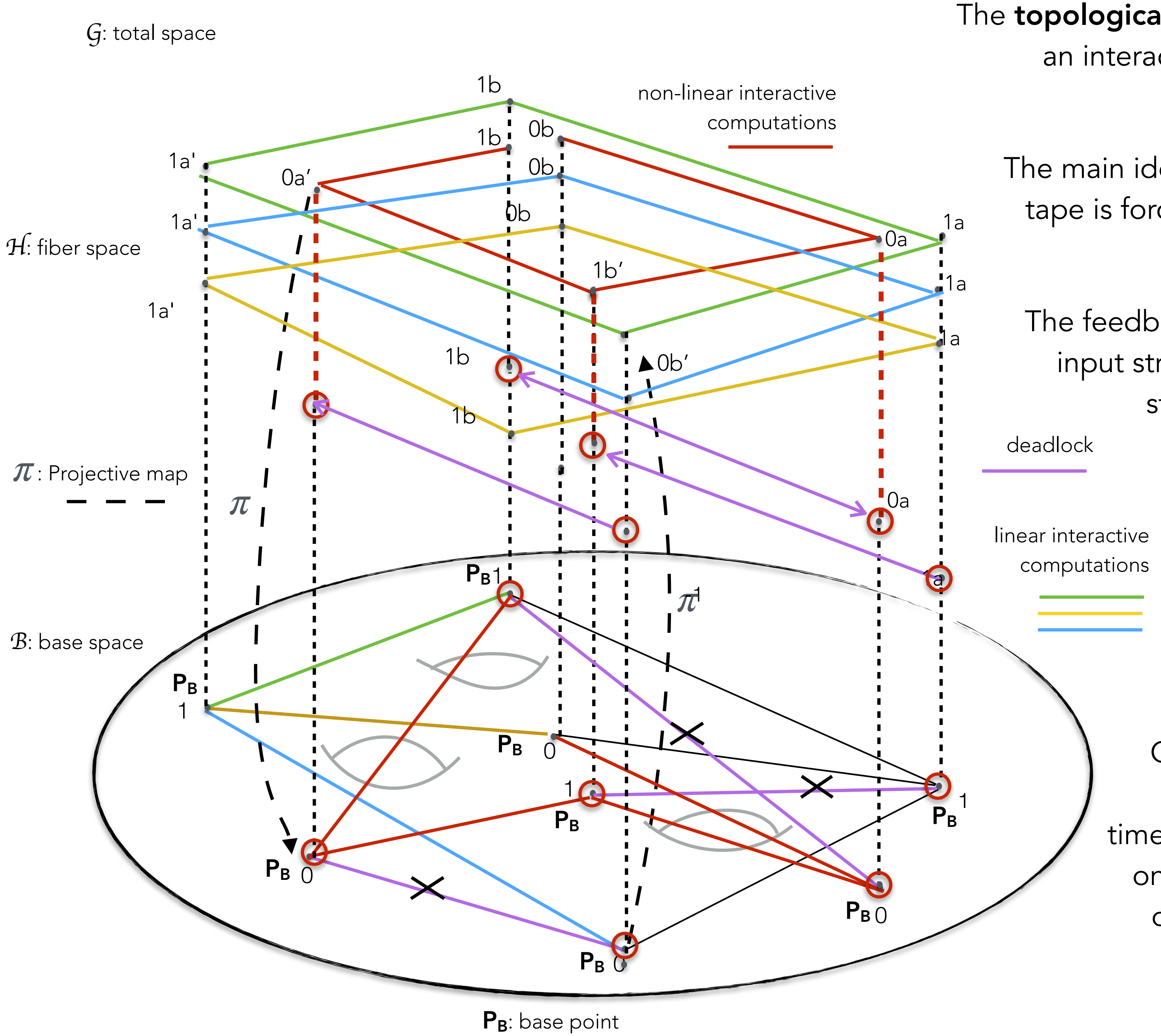


Example: trifold base space

Example: bundle diagram for Hardy probabilistic model

Topological PTM

Fiber bundle = < $\mathcal{G}, \mathcal{B}, \mathcal{H}, \pi$ >



The **ambient space** and **PTM** can be thought as mathematical representation of **complex systems**, merely defined as systems composed of many non-identical elements, constituent agents living in an environment and entangled in loops of non-linear interactions.

The **topological PTM** models both the behaviour of an interactive machine and its **computational environment.**

The main idea of the generalization is that output-tape is forced to be connected to the input-tape through a **feedback loop**

The feedback loop is modelled in a way that the input string can be affected by the last output strings, and by the current **state of the computational environment.**

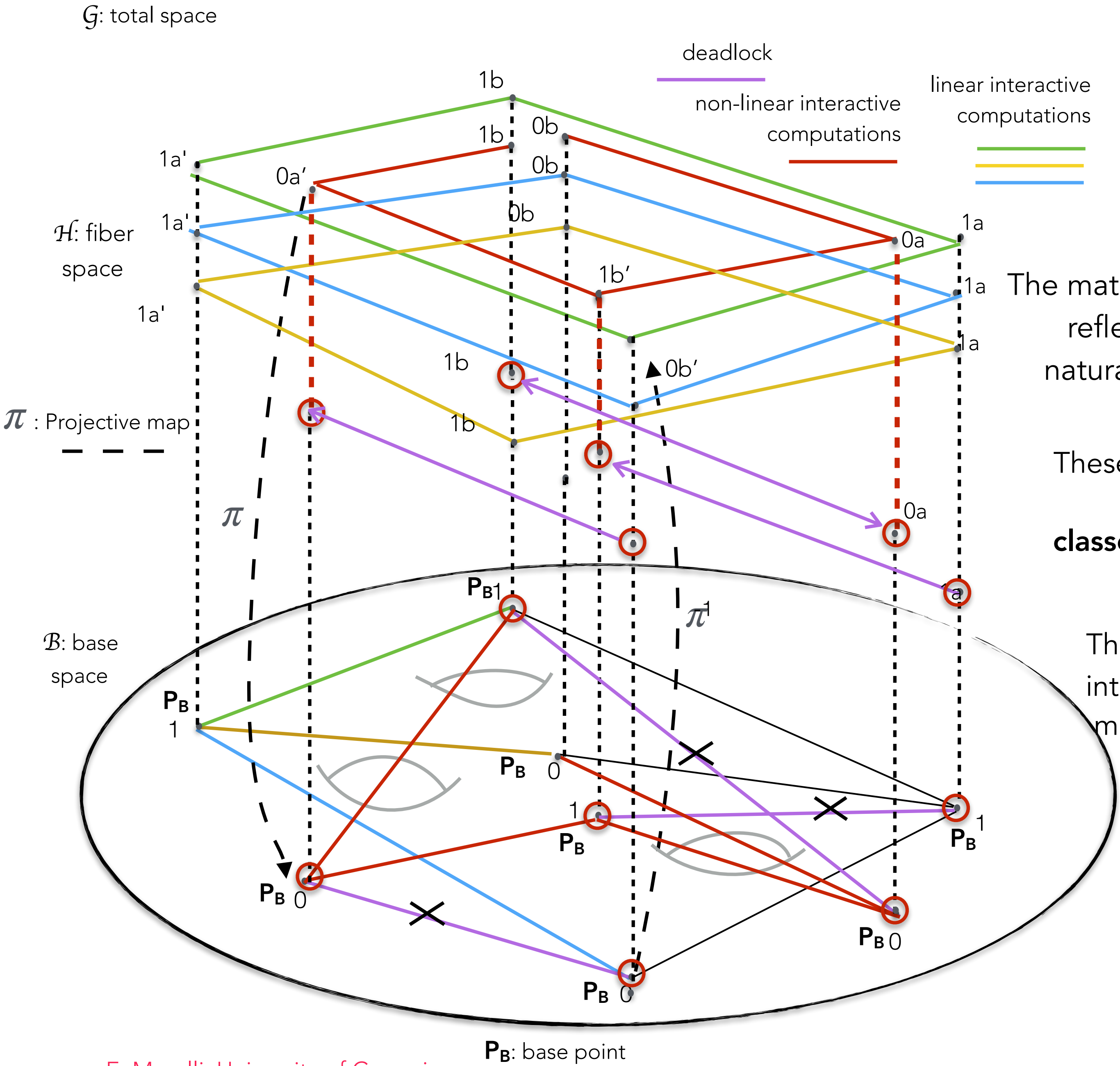
The computational environment depends on **time and space**. The time is represented by **collection of steps.**

Given a PTM, let X be a set of its input and output strings. For **each step i** in time, we define an equivalence relation $\sim_i$ on X such that $input_i$ in X there exists an operator $\mathbf{f_i}$ such that $\mathbf{f_i}(input_i) = output_i$

E. Merelli, University of Camerino

Topological PTM

Fiber bundle = < $\mathcal{G}$, $\mathcal{B}$, $\mathcal{H}$, π >



In classical Turing machine the set of **operators f_i** is called rules or transformations. Our goal is to build an **environment** where this set of functions **f_i can be discovered.**

Each element of X represents a transition from one state of the machine to a next guided by the **operator f_i (unknown for the model)** constrained over the computational environment.

The mathematical objects we are looking for should reflect the collective properties of the set X in a natural way to support the discovery of the set of operators f_i.

These operators allow us to represent X as a union of quotient spaces of the set of **equivalence classes X/ ~i** of all the **feasible relations** hidden in X.

The resulting **functional matrix of fi**, also called interaction matrix, represents the computational model or what we called the learnt algorithm in

9. E. Merelli, M. Pettini, M. Rasetti. Topology driven modeling: the IS metaphor. Natural Computing Vol.14, No.3, 2015.

In order to **characterize** the set of operators f_i, we decided to analyze the set X of environmental data by a **persistent homology**.

E. Merelli, University of Camerino

We revisit and formalize a concept of *computational environment* for *PTM* following Avi Wigderson's machine learning paradigm in [7].

Many new algorithms simply 'create themselves' with relatively little intervention from humans, mainly through interaction with massive data[4].

[4] https://www.ias.edu/ideas/mathematics-and-computation

A. Wigderson. Mathematics and Computation. IAS, Draft: March 2018.

We use the notion of computational environment to define class of abstract computable functions as **sets of relations between inputs and outputs of PTM.**

The computational environment depends on **time and space**. It can evolve and so the effectiveness of these functions depends on a given moment and a given context.

The infinite computation can be reduced to a **set of relations, constrained within its ambient space by loops of non-linear interactions.** The ambient space is **not necessarily a vector space.**

The non-linearity originated from the **shape** that can be associated to the ambient space, which can be obtained by the topological analysis of the set of data provided by the real environment.

Merelli, Wasiwleska - *Topological Interpretation of Interactive Computation* (LNCS 11500, 2019)

The Church-Turing thesis states that

every effective computation can be carried out by a Turing machine or equivalently *a certain informal concept (algorithm) corresponds to a certain mathematical object (Turing machine)* [16].

H. Lewis, C.H. Papadimitriou. Elements of the Theory of Computation. 2nd Ed. Prentice Hall, 1998.

Definition 6 (Topological environment) *Given the set of PTM configurations $\mathcal{P}_i$ available at a given time i, the* topological environment *is the simplicial complex $\mathcal{S}_{\mathcal{P}_i}$ constructed over $\mathcal{P}_i$.*

Definition 7 (Topological Turing machine) *A Topological Turing machine (TTM) is a group $\mathcal{G}$ consisting of all interaction streams generated by the group of PTMs entangled with the group of all transformations of the topological space $\mathcal{S}_{\mathcal{P}}$ preserving the topology. Formally $\mathcal{G} = \mathcal{G}_{AP} \wedge \mathcal{G}_{MC}$, where $\mathcal{G}_{AP}$ is the group of PTMs and $\mathcal{G}_{MC}$ the simplicial analog of the mapping class group.*

Proposition 1 *If $\mathcal{G}$ is automatic, the associated language $\mathcal{L}$ is regular. Since the representations of $\mathcal{G}$ can then be constructed in terms of quivers $\mathcal{Q}$ with relations induced by the corresponding path algebra induced by PTMs, the syntax of $\mathcal{L}$ is fully contained in $\mathbb{T}$ and its semantics in $\mathbb{M}$.*

Definition 8 (Constrained interactive computation) *An interactive computation is constrained if it is defined over a topological space $\mathcal{S}_{\mathcal{P}}$ and it is an element of the language of paths of $\mathcal{S}_{\mathcal{P}}$.*

Theorem 4 *Any* constrained interactive computation *is an effective computation for a TTM.*

Thesis 2 *Any concurrent computation can be performed by a TTM.*

Merelli, Wasiwleska - *Topological Interpretation of Interactive Computation* (LNCS 11500, 2019)

4 Final remarks

In 2013, Terry Tao in his blog [20] posted this question: if there is any computable group G which is "Turing complete" in the sense that the halting problem for

any Turing machine can be converted into a question of the above form. In other words, there would be **an algorithm** which, when given a Turing machine T, would return (in a finite time) a pair x_T, y_T of elements of G with the property that x_T, y_T generate a free group in G if and only if T does not halt in finite time. Or more informally: *can a 'group' be a universal Turing machine?*

Mathoverflow. https://mathoverflow.net/questions/88368/can-a-group-be-a-universal-turing-machine

Is our gauge group $\mathcal{G}$ a new algebraic models of computation for interactive computation?

Can the $\mathcal{G}$ group be a Universal Turing machine?

Which process semantics is associated to $\mathcal{G}$?

For which process domain?

Is our gauge group $\mathcal{G}$ a new algebraic models of computation for interactive computation?

Can the $\mathcal{G}$ group be a Universal Turing machine?

Which process semantics is associated to $\mathcal{G}$?
For which process domain?

References

On Topological Algorithmic Field Theory of Data

- Merelli, Wasiwleska - *Topological Interpretation of Interactive Computation* (LNCS 11500, 2019)

On Interactions of RNA and Proteins

- Maestri, Merelli - *Process calculi may reveal the equivalence lying at the heart of RNA and proteins (*Nature Scientific Reports 9,1, 559, 2019)
- Quadrini, Tesei, Merelli - *An algebraic language for RNA pseudoknots comparison* (BMC Bioinformatics 20(161), 2019)

On Persistent Entropy Automata

- Piangerelli, Tesei, Merelli - *A Persistent Entropy Automaton for the Dow Jones Stock Market (*LNCS 11761, 2019)
- Rucco, Rocio-Gonzalez-Diaz, Merelli - *A new topological entropy-based approach for measuring similarities among piecewise linear functions* (Signal Processing 134, 2017)

On Topological Classifiers

- Piangerelli, Rucco, Tesi, Merelli - *Topological Classifier for Detecting the Emergence Epiletic Searures* (BMC Research Notes 11(1),392, 2018)

Thanks

Oltre il confine, Salone Int. del libro, Torino 2017