TÉCNICO+
FORMAÇÃO AVANÇADA

Advanced Descriptive Modeling

Learning from Complex Data Structures

DASH: Data Science e Análise Não Supervisionada

Rui Henriques, rmch@tecnico.ulisboa.pt
Instituto Superior Técnico, Universidade de Lisboa

Outline

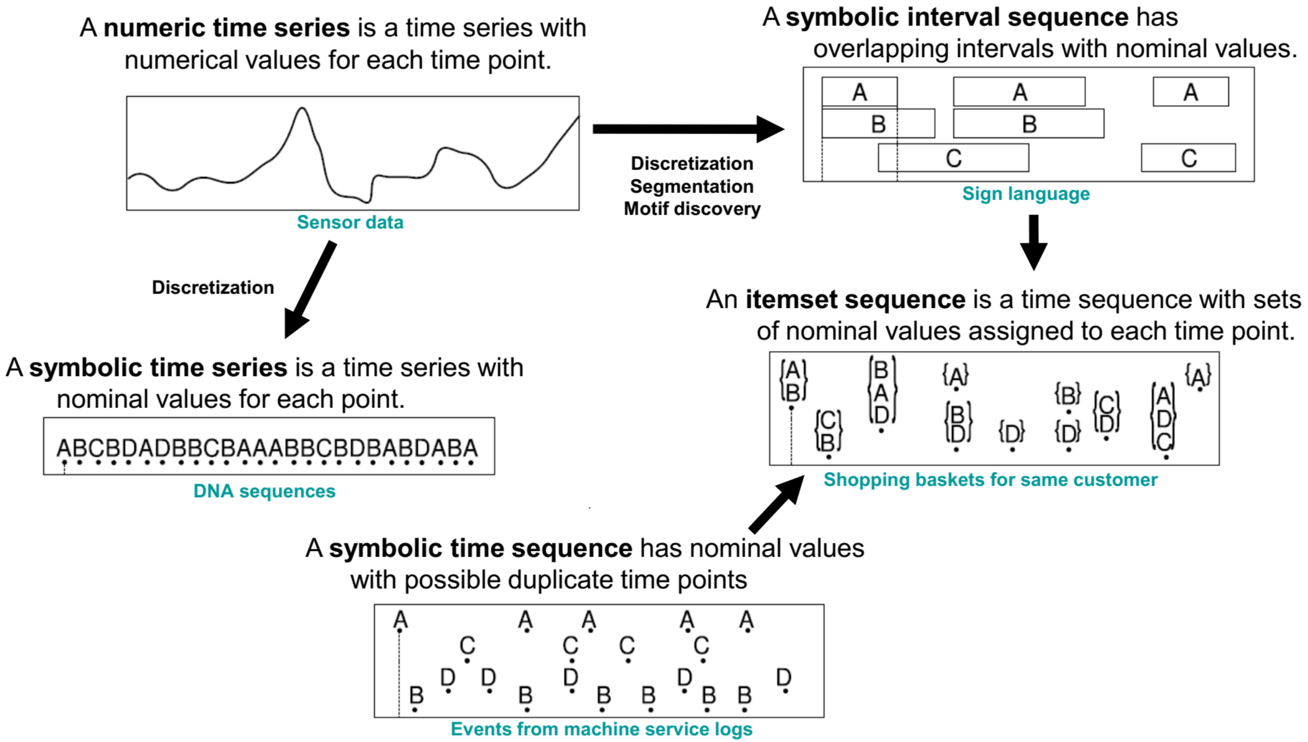
- Learning from temporal data
    - temporal data structures
    - five learning families
    - sequential pattern mining
    - time series pattern mining
    - event pattern mining
- Learning from spatiotemporal data
- Learning from multi-dimensional and relational data

Outline

- **Learning from temporal data**
  - **temporal data structures**
  - five learning families
  - sequential pattern mining
  - time series pattern mining
  - event pattern mining
- Learning from spatiotemporal data
- Learning from multi-dimensional and relational data

Temporal data structures

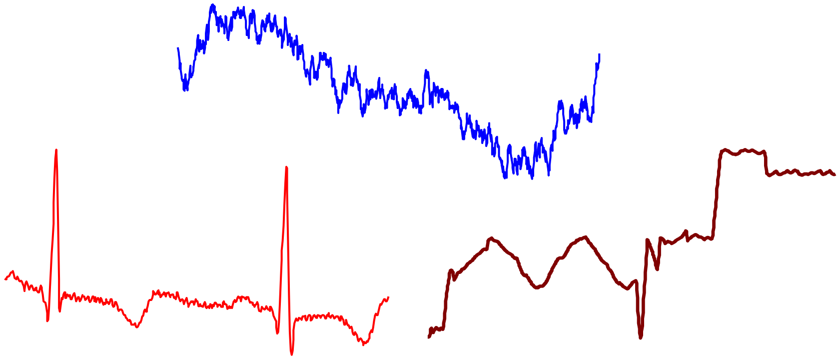
- Some examples…



Time series

- **Time series**: sequence of values or symbols along time $\mathbf{x} = < x_1, .., xT >$
  - *univariate* or *multivariate*, $x_j \in \mathbb{R}^m$ (or $x_j \in \{Y_1..Y_m\}$), where m is the multivariate order
- **Time series data**: $\{\mathbf{x}_1, .., \mathbf{x}_n\}$ where $\mathbf{x}_i$ is a time series
- People measure things...
  - *their blood pressure*
  - *the annual rainfall in New Zealand*
  - *the value of their Yahoo stock*
  - *the number of web hits per second*

  ... and things change over time



time series occur in near every public, scientific and businesses domain

Time series

- Time series are *ubiquotous*: monitoring biological, individual, organizational, geophysical, digital, mechanical, societal systems
- Movement, image and video as time series
- Text data as time series

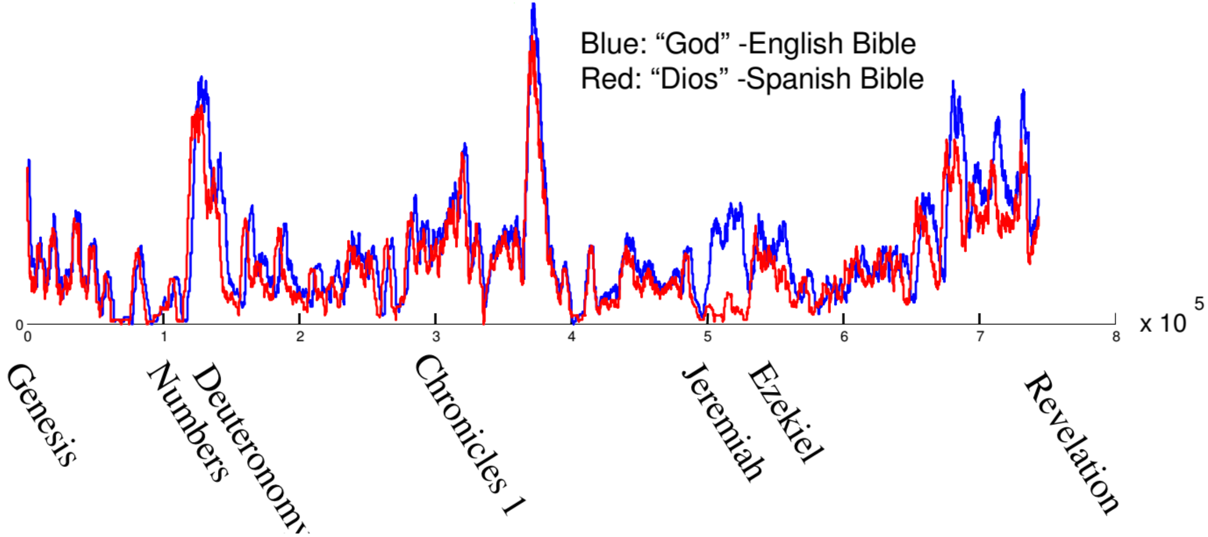
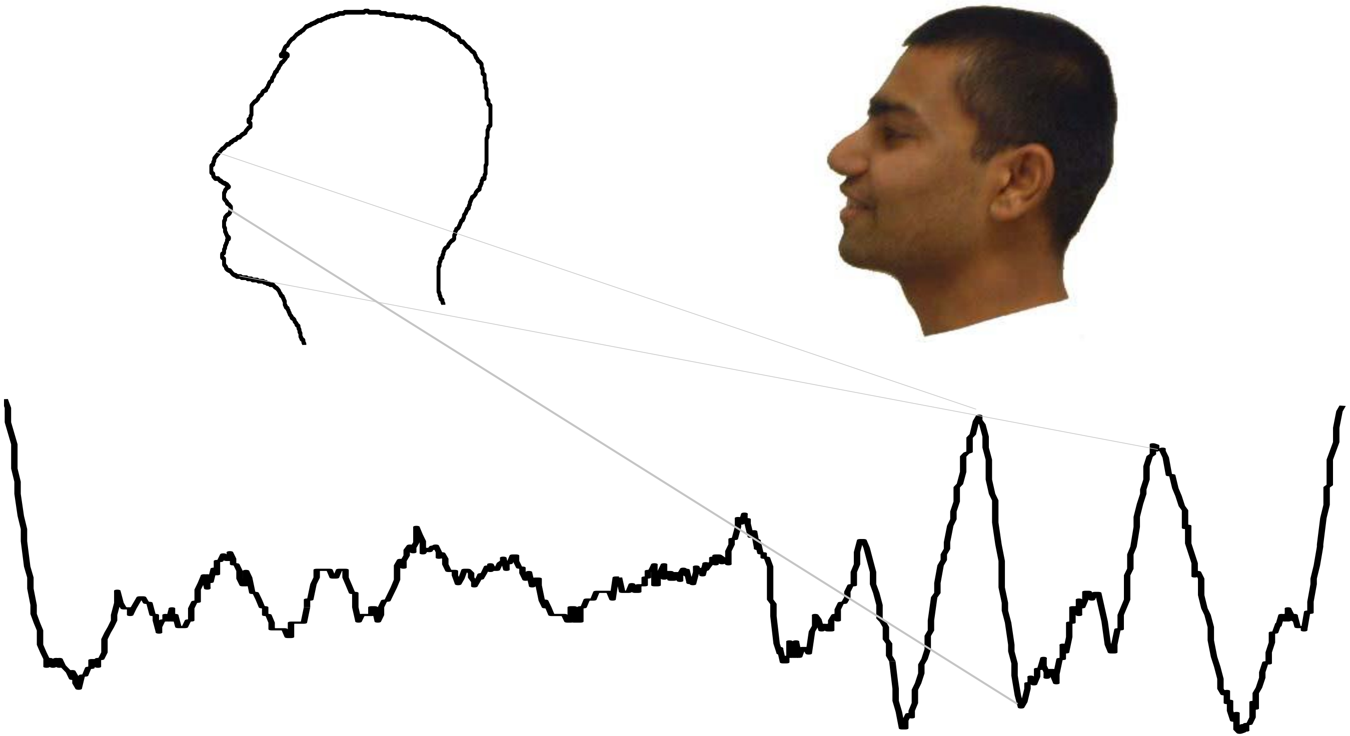
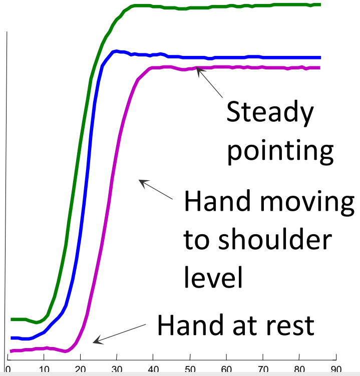


Image data as time series

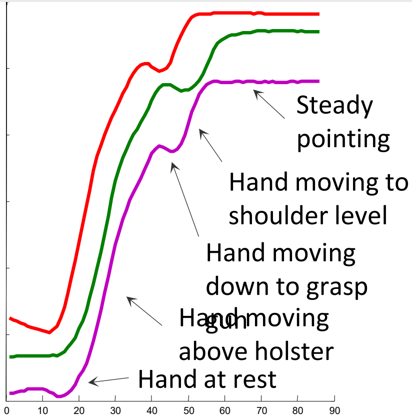


Video data as time series

Point



Gun-Draw



Sequence data

Examples:

- website navigation
- shopping behaviour
- DNA (univariate symbolic sequence)

Focused on ***orders*** instead of *time points*

Recall: transactional data structures

- market basket analysis at the level of the basket
- What if I want to mine sequences of baskets per user?
- *Answer*: **itemset sequence** data, a type of sequence data

Event data

Event: timestamped occurrence

- non-variate, univariate (typed event), multivariate (e.g. high and low blood pressure)

Event sequence: set of events

Event data: set of observations, each being an event sequence

Examples:

- health records
- social interactions
- machine logs
- musical melodies
- headline events

Outline

- **Learning from temporal data**
  - temporal data structures
  - **five learning families**
  - sequential pattern mining
  - time series pattern mining
  - event pattern mining
- Learning from spatiotemporal data
- Learning from multi-dimensional and relational data

Temporal data mining

Describe: understand the past and present

- decomposition into essential elements: trend, seasonalities, spectral components, noise
- causality analysis, temporal rules
- motifs and other patterns of interest

Predict the future and unseen features

- anticipate events of interest
- classify and regress time series for system diagnostics and prognostics
- forecast time series

Exercise: major needs in corporate domains?

- mining user behavior, KPIs, stock prices, demand, imports, supply, GDP…

Temporal data mining

Temporal data mining (TDM) tasks include:

- temporal data **clustering**
- temporal data **classification**, **regression**
- temporal data **modelling** (e.g., explanatory equations and mixtures)
- temporal data **transformation** (e.g., frequency domain representations)
- temporal data **reduction**/compression (e.g., variants of PCA for series)
- temporal **pattern mining**, associative analysis
- temporal data outlier/**anomaly detection**
- temporal data **visualization**
- temporal data **forecasting**

TDM can be further characterized by the:

- target temporal data structure
- presence/absence of temporal semantics (domain knowledge)

Learning from temporal data

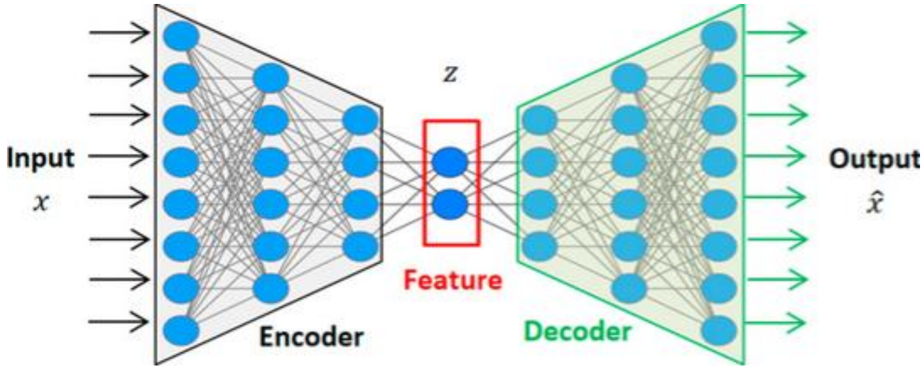
Five major options

1. **traditional** descriptors/predictors
   - *first*: map temporal data into multivariate data
     - retrieve **statistics** along time: centrality, variance, regression coefficients, percentiles…
     - retrieve statistics along **time windows** (rolling mean and rolling variance) and discard temporal dependence between the statistics produced for each window
     - learn **embeddings**: data representations using neural networks
   - *second*: apply multivariate methods (clustering, pattern mining, NBs, DTs, RFs, SVMs, NNs, etc.)
2. **distance-based** descriptors/predictors
3. **associative** descriptors/predictors
4. **deep** descriptors/predictors
5. **prompt-engineered** LLM descriptors/predictors

Learning from temporal data: traditional

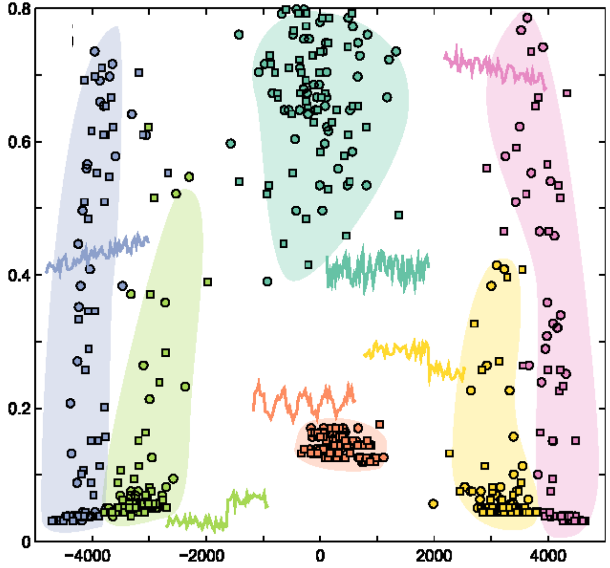
Temporal data can be mapped into numeric vectors (***embeddings***) to subsequently apply **classic ML**

- embeddings are latent feature representations with minimal information loss
- recall the paradigmatic unsupervised case: *autoencoders* (AE)
  - principle: preserve as much information in a compact vector by maximizing reconstruction ability
  - enhanced expressivity when considering multi-task self-supervision: check our early class!
  - the neural architecture should be able to capture temporal dependencies:
    - recurrent layering (e.g., LSTMs), convolutional layering, transformer layering...
- classic ML descriptors and predictors (prepared to learn from tabular data)



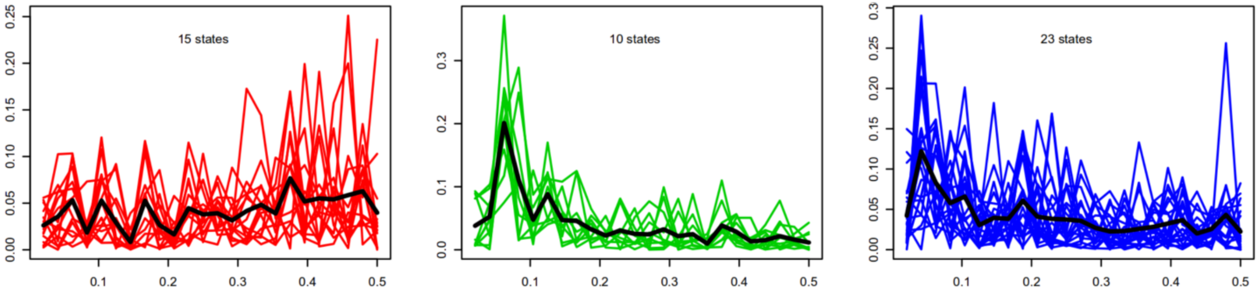
Learning from temporal data: distance-based

- Approaches that rely on distances between two observations
- The simplest regressor/classifier is **lazy learning**
  - *train*: use temporal distances to detect the *nearest neighbors*
  - *test*: use them to estimate targets



Learning from temporal data: distance-based

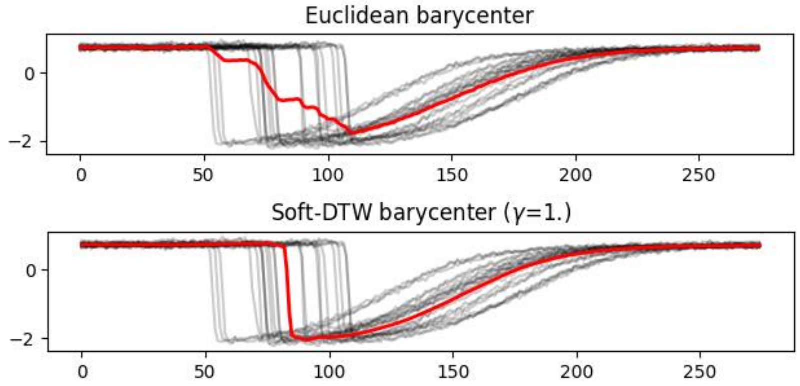
- Temporal data **clustering** based on distances between observations and centroids:
  - partition-based clustering (***k*-medoids**)
  - **agglomerative** clustering
  - **density-based** clustering
- replace simple tabular distances (e.g. Minkowski) for distances able to accommodate temporal misalignments (e.g. elastic distances such as DTW for time series)



TÉCNICO+
FORMAÇÃO AVANÇADA

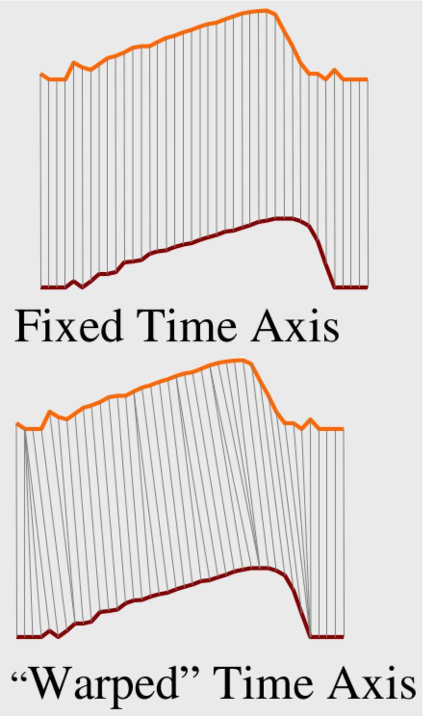
Learning from temporal data: distance-based

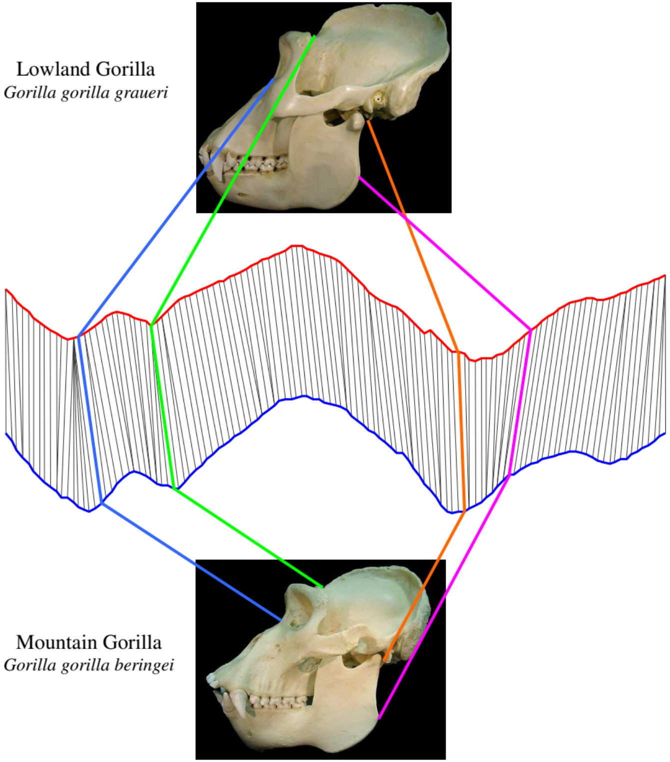
- **Centroid** of a cluster of temporal observations (e.g., time series) referred as **barycenter**



- Partitioning-based methods
  - *means* not adequate as centroid if time series are misaligned
  - solution: *medoids* (prototype time series minimizing DTW)
    or barycenter-driven *k*-means (e.g. tslearn package)

Elastic time series distances





Learning from temporal data: associative

Rule-based predictor

- *TRAIN*: given observations with temporal information
  - find temporal patterns
  - use the found temporal patterns to produce association rules ***pattern* ⇒ *class***
    - rule interestingness criteria reveal the discriminative power of the rule
    - *lift* is a good proxy when data is imbalanced

- *TEST*: given a new observation
  - find the closest patterns from the produced rules
  - label the time series using the rules' consequents (voting stage)
    - mode from either all matched rules or top-*k* closest rules
    - weight mathched rules by their relevance and discriminative power

Learning from temporal data: associative

Pattern-centric predictor

- *TRAIN*: given the time series data
  - discover temporal patterns and create a tabular dataset with a feature per pattern
  - fill the tabular dataset in accordance with one of the following options:

1. boolean dataset: whether a time series x_i has or not a given pattern y_j
2. real-valued dataset: how well a time series x_i contains a given pattern y_j

  - apply classic classifiers to learn a model using this tabular dataset

- *TEST*: given a new time series
  - produce the feature-vector: test if the testing time series has the given patterns (boolean) or how well captures the patterns (real-valued)
  - apply the trained classifier to on the feature-vector to return a label

Learning from temporal data: associative

Euclidean ([8,1,1], [2,7,5])= 9.38

	y1	y2	y3	y4	y5	c
x1	1	2	8	1	1	0
x2	4	8	2	7	4	1
x3	7	4	2	7	5	1
x4	2	6	2	6	5	0
x5	5	1	5	7	8	0
x6	8	7	2	2	1	0

Transformation

	p1	p2	p3	c
x1	9.38	...	...	0
x2	1	...	...	1
x3	0	...	...	1
x4	1	...	...	0
x5	4.24	...	...	0
x6	6.4	...	...	0

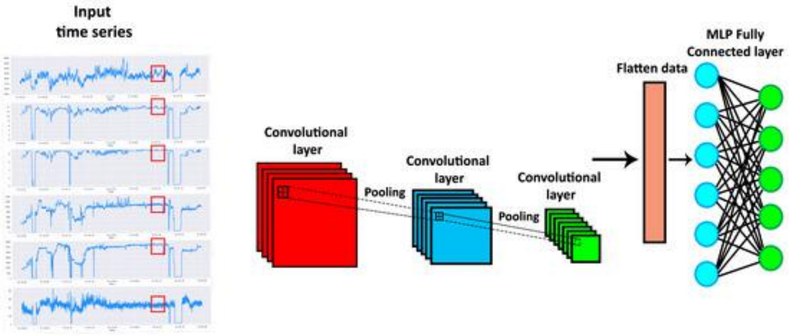
Pattern 1
$y_3=2;\ y_4=7;\ y_5=5$

$$\text{lift} = \frac{\frac{2}{6}}{\frac{3}{6} \times \frac{2}{6}} = 2$$

- Under this mapping: clustering solutions and high-order patterns can be as well pursued

Learning from temporal data: deep

Dedicate *layering*: **recurrent** (e.g. LSTMs), **convolutional** (1D or 2D depending on whether data is univariate or multivariate), temporal convolutions, **transformer**-based…

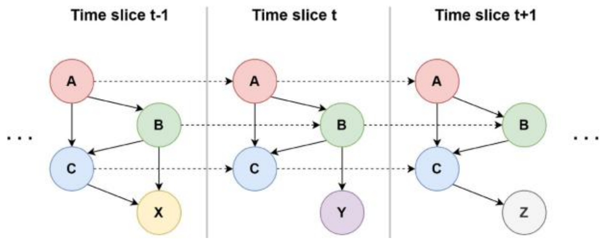


- *Descriptive:* learn NNs to denoise (autoenconding observations), extract features, impute…
- *Precitive:* end-to-end supervised learning
  - TRAIN: learn expressive functions that map time-rich inputs and corresponding targets
  - TEST: apply the function on testing observations and return the estimates

Learning from temporal data: deep

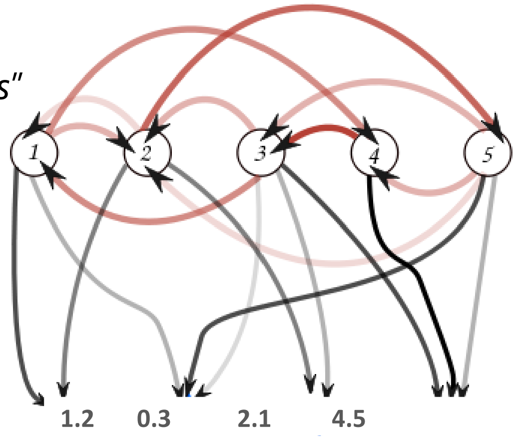
Classic time-aware models can be deep (many parameters): although less used, remain relevant

Dynamic Bayesian networks



Hidden Markov models

- *Descriptive*: describe system dynamics from time series or sequence data
    - check "*generative modeling of repositories of health records*"
- *Predictive*:
    - TRAIN: learn an automaton per class based on the observed time series for each class
    - TEST: given a testing time series, see how each of the automaton better describes it and output the automaton's class as the result



Learning from temporal data: LLM prompting

- Given a **data-rich prompting**…
  can **LLMs** be used to extract features, learn descriptions and place predictions data?
- The answer largely depends on task complexity: use LLMs with great caution
  - poor performance on hard, specialized tasks and even in some simple ones
    (e.g., attention to info in the text prompt may override the attached structured data)
  - **pros**
    - flexible, fast, task-adaptive without retraining
    - works with missing, noisy, or heterogeneous data
  - **cons**
    - limited numerical precision and temporal grounding
    - weak guarantees on predictive accuracy
    - sensitive to prompt design and assumptions

Learning from temporal data: LLM prompting

Yet an option as LLMs hold unique "analytical" capacities from:

- **large-scale multi-task pretraining**
  - learn cross-domain regularities transferable to different tasks (e.g. descriptive, predictive, inferential), enabling generalization beyond task-specific models
  - store latent abstractions of patterns, relationships, and dependencies
- **universal sequence modeling**
  - trained to model sequences, allowing them to process text, code, time series (as serialized inputs) and other (time-rich) data structures
  - inherent capacity to handle variable-length and irregular inputs
- **in-context learning**
  - can infer task structure directly from prompts (zero-shot)
  - can be guided by few-shot examples without the need for parameter updates (fine tuning)

Learning from temporal data: LLM prompting

LLM “analytical” capacities:

- *Compositional Reasoning* for chain descriptive-predictive subtasks in complex data analysis pipelines
- *Flexible Input–Output Mapping* with prompting allowing for custom input and output formats
- *Prior Knowledge Integration* from learnt domain heuristics and background associative patterns
- *Robustness to Imperfect Data* (tolerance to missing, noisy, inconsistent inputs) from contextual inference
- *Task Unification via Language* by reformulating data-driven tasks as a “*predict the next token*” task
- *Prompt-Controlled Inductive Bias* by imposing temporal-causality assumptions and constraints on reasoning and outputs, effectively steering the model toward task-specific behavior

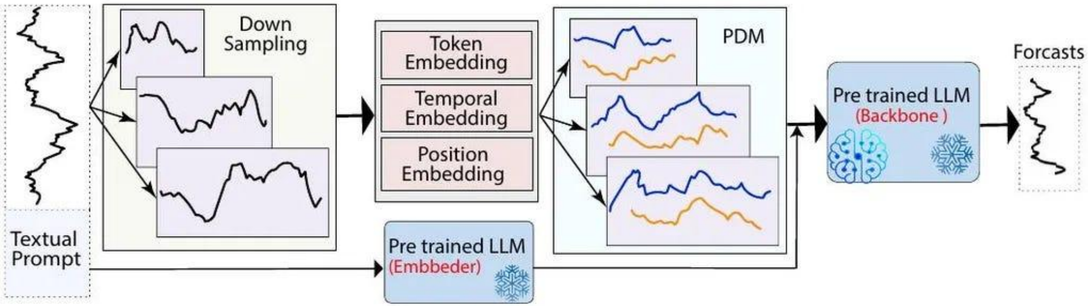
The promise: the increasing *LLMs size* and *context from multi-modal and multi-stage analyses* can push LLMs to answer complex, specialized descriptive and predictive tasks

Learning from temporal data: LLM prompting

- Prompting principles for *descriptive* tasks
    - explicitly define **temporal scope** (time window, granularity)
    - ask for **trend-aware descriptions** (seasonality, peaks, regime shifts)
    - use **structured outputs** (tables, bullet summaries, timelines)
    - on inferential tasks, encourage **step-by-step temporal reasoning**
    - request **explicit feature identification** (lags, growth rates, volatility)
    - compare **multiple time windows or regimes**
- **Best suited for**
    - exploratory analysis, hypothesis generation, feature engineering support
    - reporting, data quality assessment, model interpretability

Learning from temporal data: LLM prompting

- Some multimodal LLMs indeed trained on temporal data…



- Prompting principles for *predictive tasks*
  - clearly specify the prediction **horizon** and **assumptions**
  - constrain outputs to **probabilistic** or **scenario-based estimates**
  - explicitly ask to separate **data-driven inference** from speculation
- Example: *Given the historical data, forecast the next 7 time steps, disclosing uncertainty drivers.*
- **Best suited** for short-term forecast assistance, scenario analysis, decisions under uncertainty

Outline

- **Learning from temporal data**
  - temporal data structures
  - five learning families
  - **sequential pattern mining**
  - time series pattern mining
  - event pattern mining
- Learning from spatiotemporal data
- Learning from multi-dimensional and relational data

Sequential pattern mining

- A **sequence** is an ordered list of events, denoted $< e_1\, e_2\, \dots\, e_l >$
  - an event can be univariate, multivariate or an itemset
- Given two sequences $\alpha =< a_1\, a_2\, \dots\, an >$ and $\beta =< b_1\, b_2\, \dots\, bm >$
  - α is called a **subsequence** of β, denoted as $\alpha \subseteq \beta$, if there exist integers $1 \leq j_1 < j_2 < \dots < j_n \leq m$ such that $a_1 \subseteq bj_1, a_2 \subseteq bj_2, \dots, an \subseteq b_{jn}$
    - β is a **supersequence** of α
  - e.g. <a(bc)dc> is a *subsequence* of <a(abc)(ac)d(cf)>
- A **sequence database** is a set of (itemset) sequences
- A **sequential pattern** is an association capturing relevant **precedences** and **co-occurrences** in a sequence database

Sequential pattern mining (SPM)

- Given a set of sequences and support threshold, **SPM** tasks aims at finding all *frequent* subsequences
  - *example*: considering a min support of 2: <(ab)c> is a **sequential pattern**
  - *challenges*?

SID	sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

- Extend the definition to further guarantee:
  - **statistical significance** of sequential patterns (unexpectedly high frequency)
  - ability to incorporate various kinds of user-specific constraints to focus on **novel**, **actionable** and **non-trivial** patterns

Sequential pattern mining (SPM)

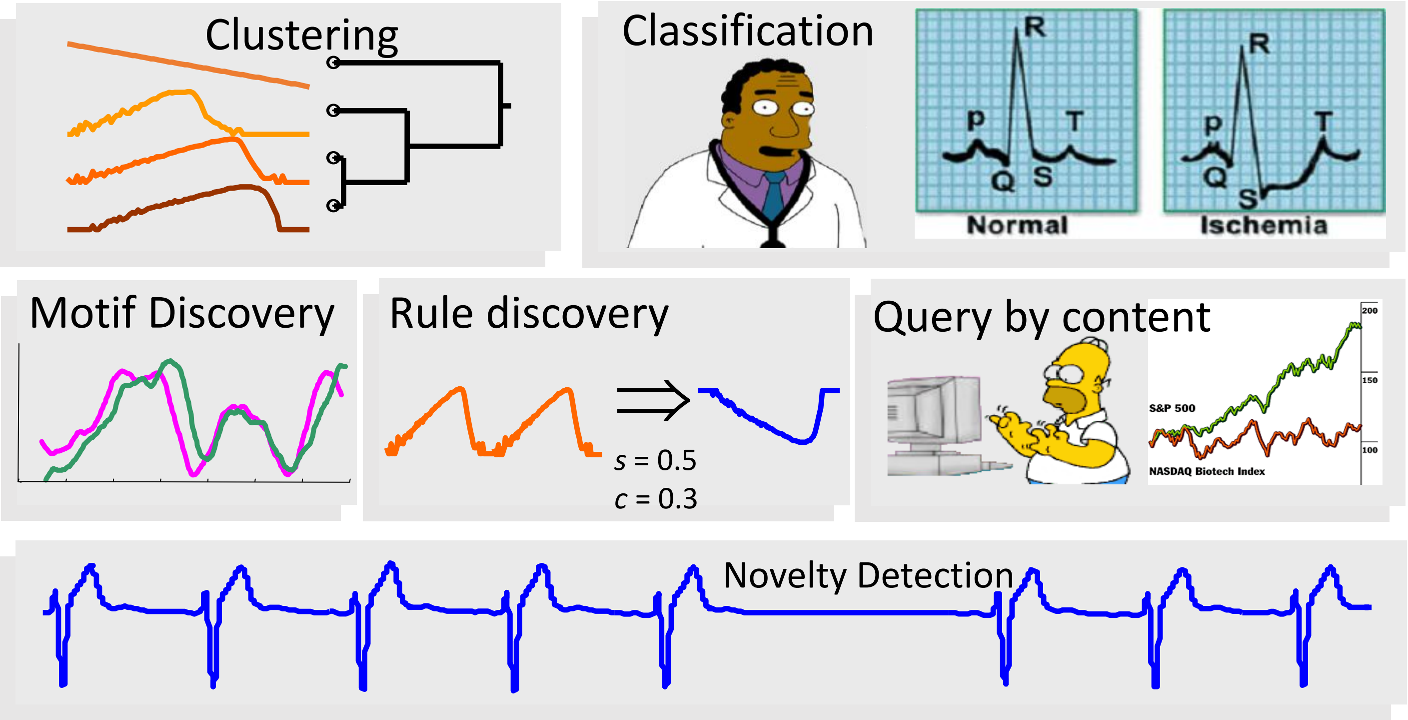
- SPM is computationally complex! **Approaches**:
  - candidate generation: Apriori-like (e.g. **GSP** method)
  - pattern growth using suffix trees (e.g. **PrefixSpan**)
- **Voluminous** solutions?
  - exact same principles as in classic pattern mining:
    filtering, condensed pattern representations, dissimilarity
- **Statistical significance**
  - exact same binomial statistical test as in simple patterns on the pattern support
  - null model based on Markov assumption
    - e.g. $p_{null}(< adc >) = p(a)p(< ad > |a)p(< dc > |d)$ where the probabilities are based on the frequentist view (ratio of observations)
      - using previous table $p_{null}(< adc >) = 1 \times \frac{2}{4} / 1 \times \frac{3}{4} / \frac{3}{4} = \frac{2}{4}$

TÉCNICO+
FORMAÇÃO AVANÇADA

Outline

- **Learning from temporal data**
  - temporal data structures
  - five learning families
  - sequential pattern mining
  - **time series pattern mining**
  - event pattern mining
- Learning from spatiotemporal data
- Learning from multi-dimensional and relational data

Recall: learning from time series data



Pattern mining in time series

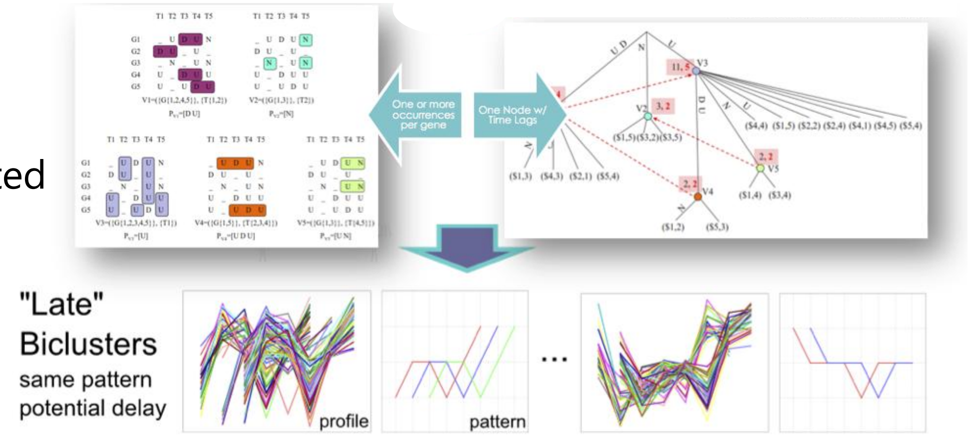
Frequency…

- *across* time series observations
  - **sequential pattern mining** (frequent orders and precedences in symbolic series data)
  - **biclustering** (univariate time series data)
  - **triclustering** (multivariate time series data)
  - **temporal association rules**
- *within* single time series
  - **motif discovery**
  - **predictive rule mining** $A \Rightarrow^{\Delta t} B$

    once antecedent is observed, consequent expected within interval Δt

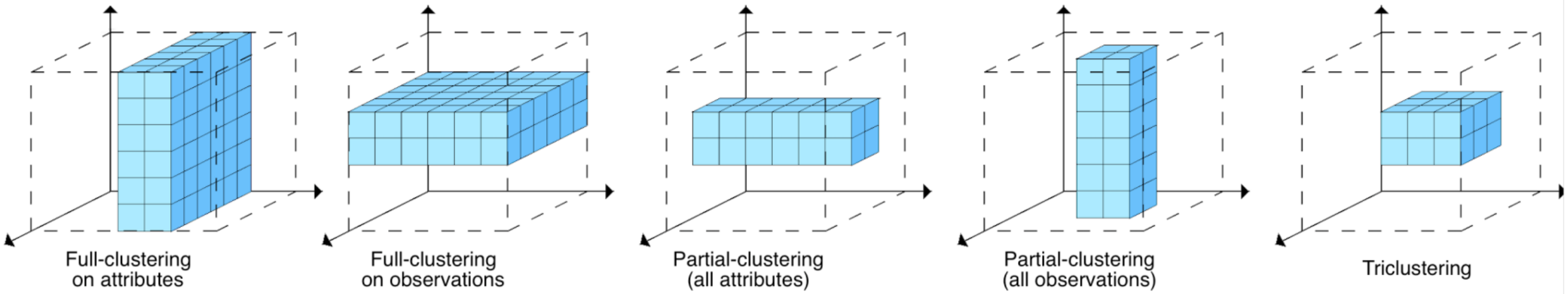
Time series biclustering

- Recall: **biclustering** aims at discovering patterns in simple multivariate data such that each pattern satisfies specific criteria of *homogeneity* and *statistical significance*
  - can further include *dissimilarity* and, given variables of interest, *predictive power*
- Biclustering is also used to retrieve patterns from *univariate time series*
  - a *bicluster* is a subset of **observations** with coherent values on a subset of **time points**
  - **contiguity** is generally assumed across time points (convex temporal pattern)
  - **temporal misalignments** between observations can be further accommodated (e.g. patients at different disease stages)

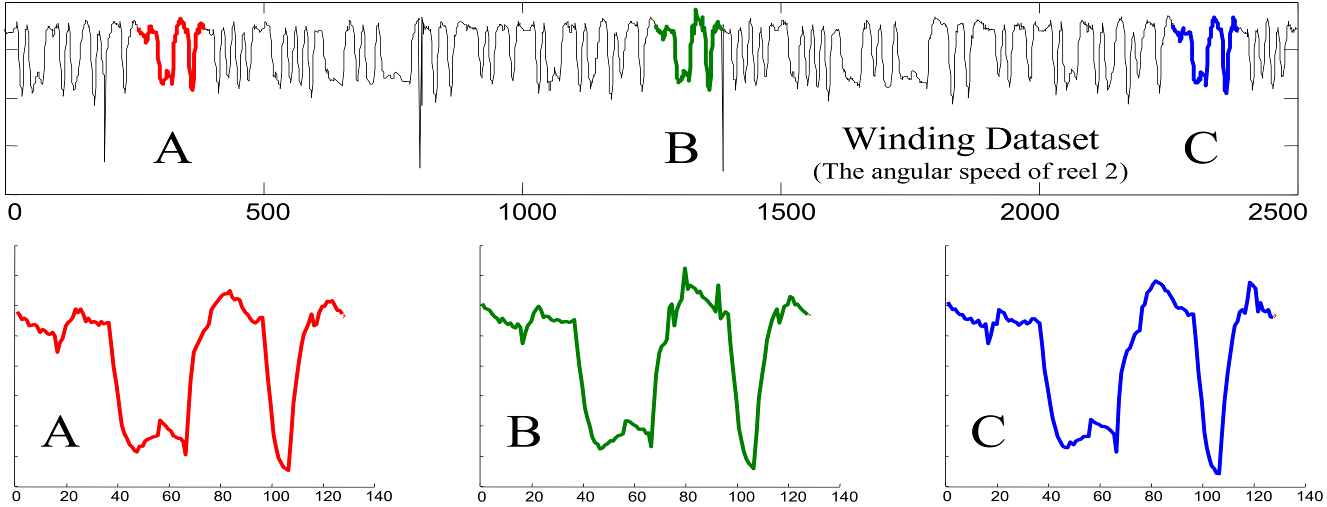


Triclustering

- How to discover patterns in multivariate time series (MTS) data?
  - MTS data: each observation is described by a set of variables measured along time
- Option: **triclustering**
  - a **tricluster** is a subset of *observations*, *variables* and *time points* with good:
    - homogeneity, e.g. well established temporal pattern on a subset of variables
    - statistical significance, e.g. unexpected high #observations supporting the pattern



Motif discovery

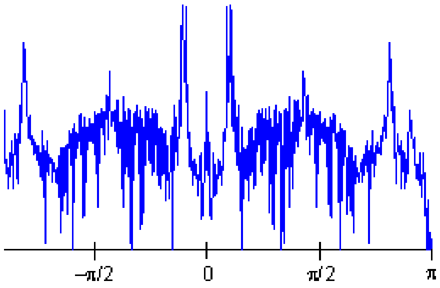


- To exhaustively find all motifs: combinatorially explosive number of distances to compute
    - the obvious brute force search algorithm is just too slow!
    - one solution: symbolize the time series and apply efficient intersections on a sliding basis

Signals… any care?

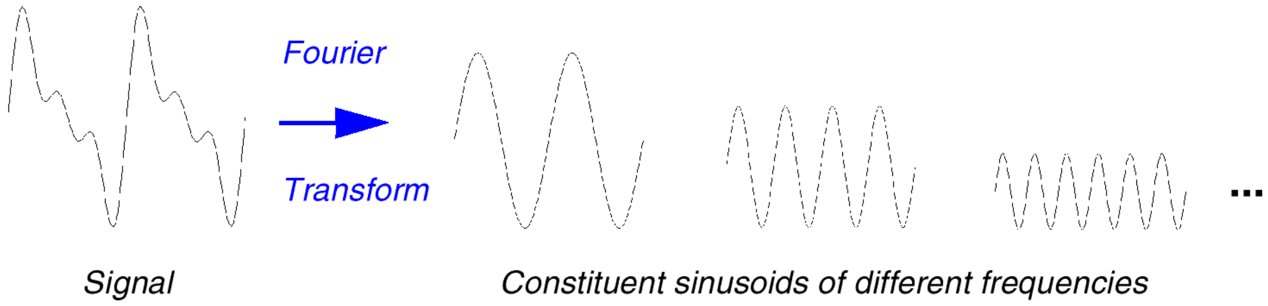
- Signals collected from sensors show unique complexities

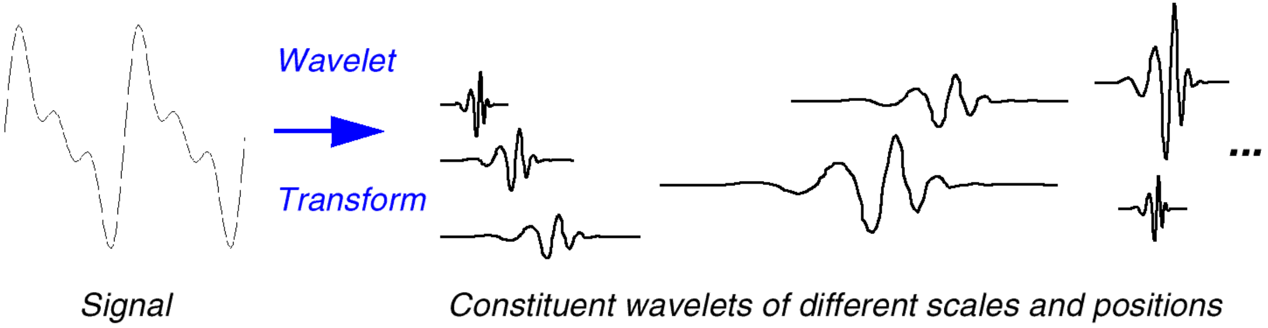
- Time series representations essential to:
  - Reveal the **internal structure** of the signal
  - decompose signal into a set of meaningful components



- Alternatives to deep learning? Yes…
  - describe *raw signal* as a finite <u>*composition of well-known abstractions*</u>
  - spectral analysis
    - **Fourier** transform for biosensors (e.g. brain waves from EEG)
    - **Wavelet** transform in telemetry (e.g. appliances on from utility consumption sensors)

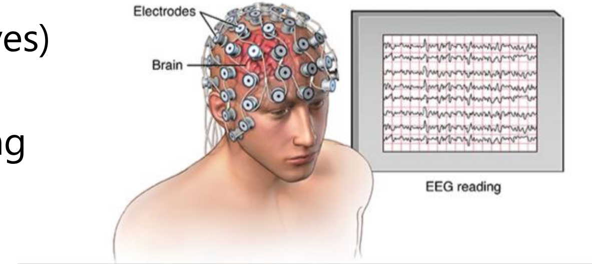
Fourier and Wavelet analysis





Case: EEG analysis

- Given a raw electroencephalographic signal:
  - multivariate time series with *p* order (corresponding to #electrodes)
  - nearly impossible to interpret in raw form
  - spectral analysis to decompose signal into activity levels (waves)
    - *gamma* (40-100Hz): cognition, info processing, learning
    - *beta* (12-40Hz): conscious focus, memory, problem solving
    - *alpha* (8-12Hz): transition between focus and relaxation
    - *theta* (4-8Hz): emotional connection, relaxation
    - *delta* (0-4Hz): healing, restorative/deep sleep



- Use **triclustering** or (multivariate) **motif discovery** to find patterns on the frequency representation of the raw signals
  - e.g. decreased alpha-to-gamma activity in the frontal lobe and increased high activity in the occipital lobe is a statistically significant pattern and discriminative of schizophrenia

Time series representations

- **Goal**: identify the structural elements of a time series
  - automatically extract explanatory components of a time series using **abstractions**
    - e.g. embeddings, spectral components…
  - useful to tackle inherent idiosyncrasies of time series data analysis
    - high **dimensionality**, noise and **variability**
    - promote ability to **measure distances** between time series

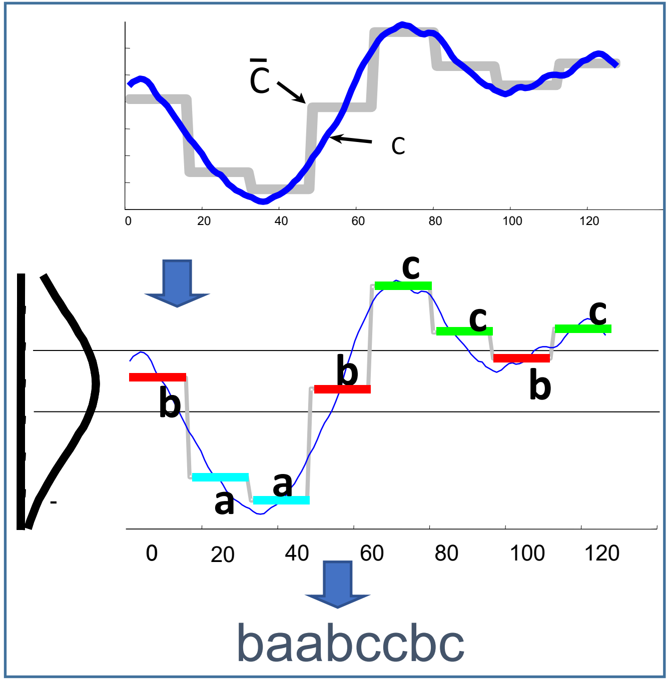
- Two large familities
  - **numeric** representations: $\mathbf{x}_i \in \mathbb{R}^n \rightarrow \mathbf{x}_i \in \mathbb{R}^p$
  - **symbolic** representations: $\mathbf{x}_i \in \mathbb{R}^n \rightarrow \mathbf{x}_i \in \Sigma^p$

Symbolic time series data

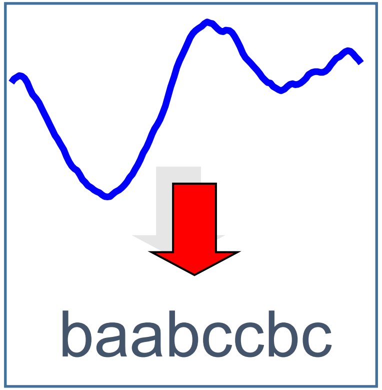
- Time series can be mapped into symbolic sequences (SAX, codebooks)
- Many research contributions from:
  - *bioinformatics* (efficient analysis of DNA and protein sequences)
  - *information retrieval* (efficient analysis of text data)

- Mining symbolic time series data is indeed similar to real-valued data:
  - **distance-based** clustering, classification and regression
    - distance metrics on strings, e.g. edit distance, Jaro-Winkler, Levenshtein
  - **associative** clustering, classification and regression
    - symbolic temporal patterns (instead of continuous motifs, patterns)

Symbolic Aggregate approXimation (SAX)



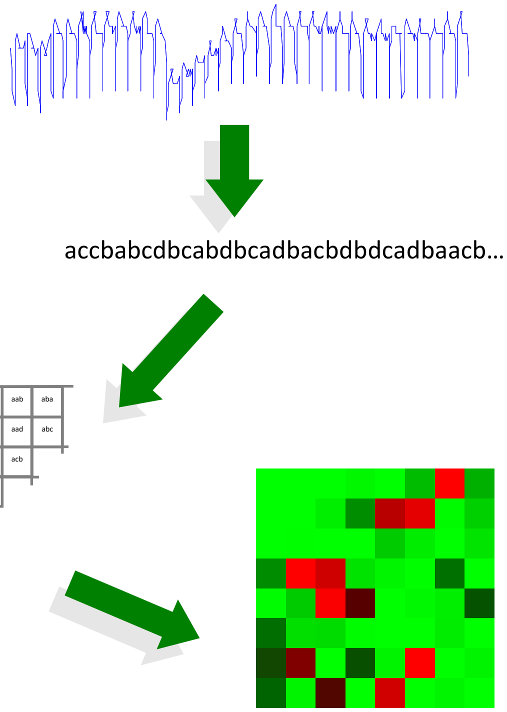
paradigmatic approach to convert/symbolize time series using piecewise aggregation followed by discretization

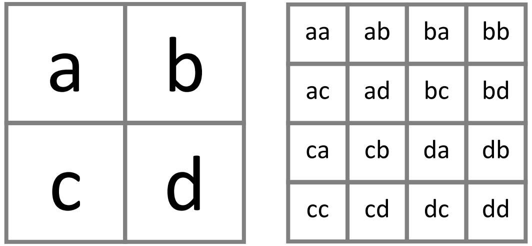


TÉCNICO+
FORMAÇÃO AVANÇADA

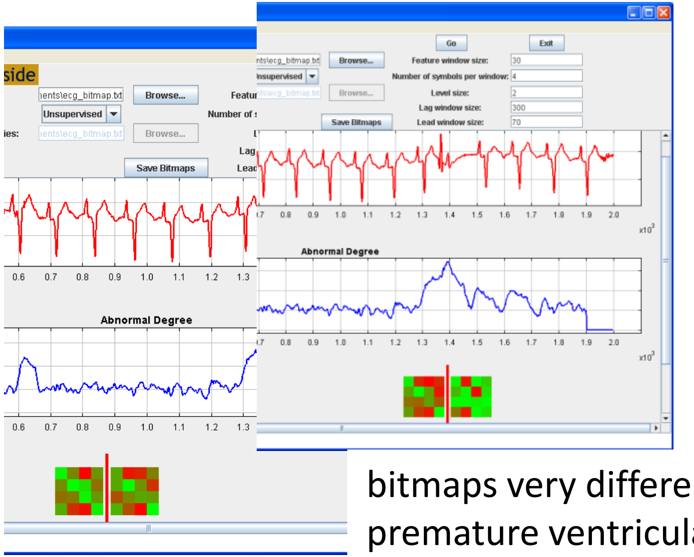
Time series bitmaps

- **SAX** benefits…
  - finding motifs
  - visualizing massive time series
  - clustering streaming time series
  - kolmogorov complexity data mining
  - classification and indexing
- These ends are grounded bitmap analysis:



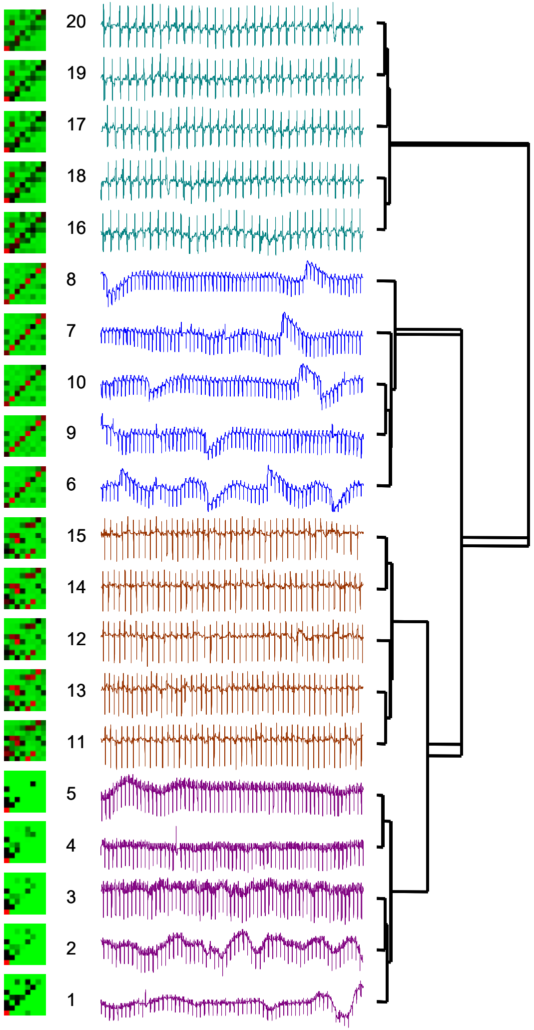


Clustering and anomalies

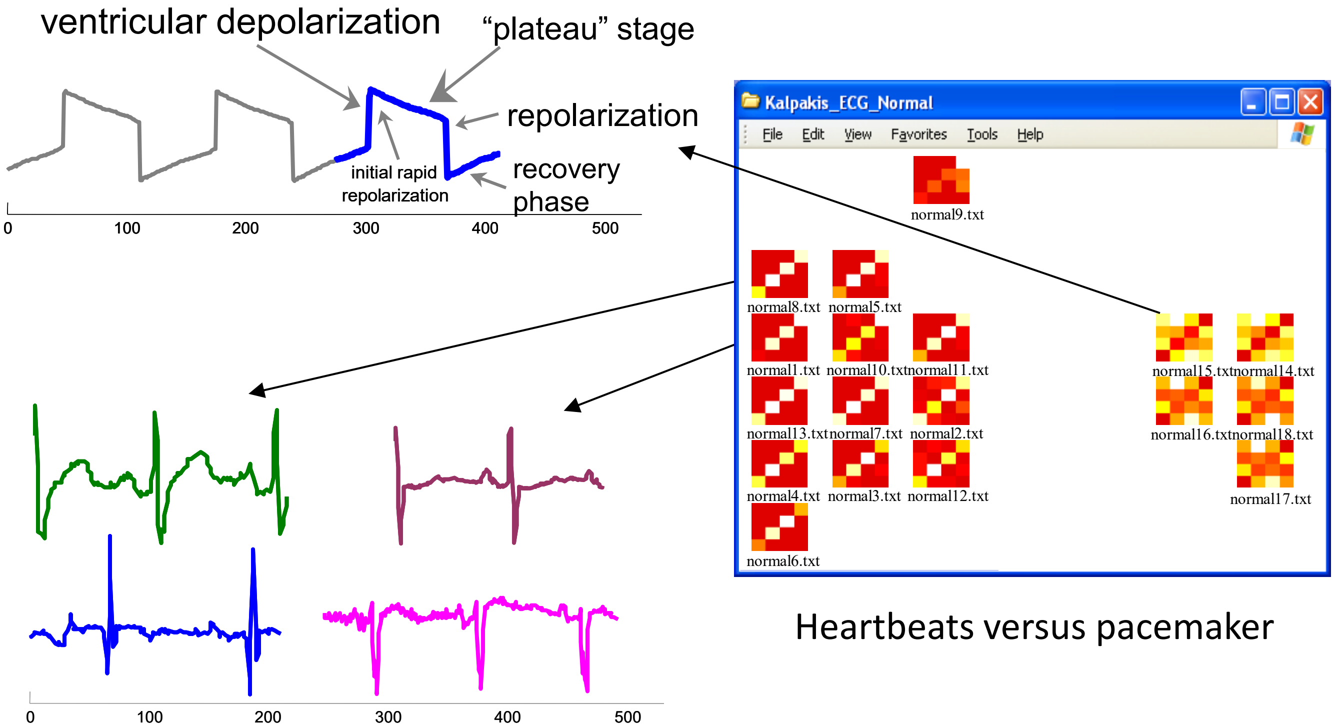


bitmaps similar
(normal behavior)

bitmaps very different in
premature ventricular
contraction (PVC)



Classification using bitmaps

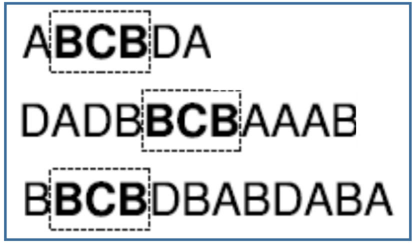


Heartbeats versus pacemaker

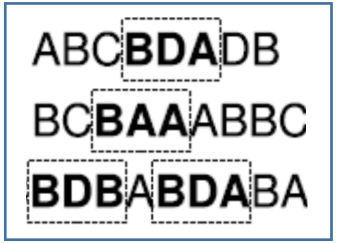
Patterns in symbolic time series data

Examples

- symbolic motifs in a single series
- **substring patterns** (**B → C → B)**
  - sequence of symbols
  - extensions to allow gaps



- **regular expression patterns** (**B → ¬C → A | B**)
  - extension to allow gaps (via wildcards), negations, repetitions, etc.

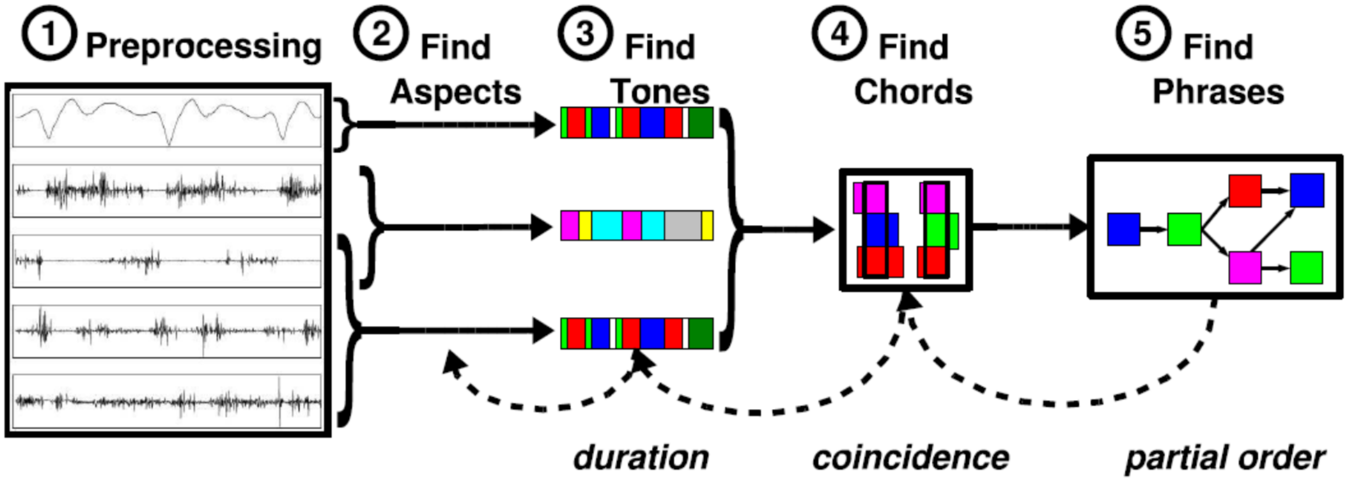


Exercise: are the given patterns sufficient to describe web usage behavior? Any addition?

Patterns in multivariate time series

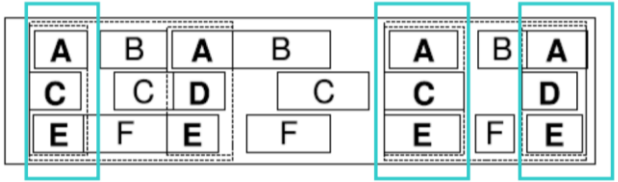
Example: pattern mining on multivariate time series

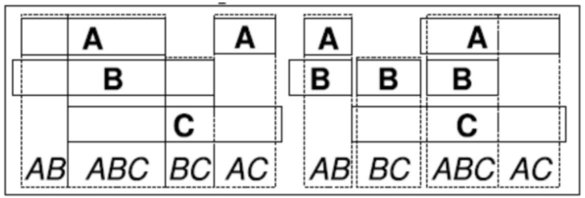
- **tone mining**: discretization, segmentation
- **chord mining**: variation of itemset mining
- **phrase mining**: variation of partial order mining



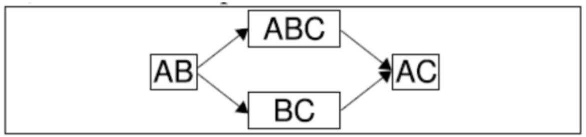
Patterns in multivariate time series

- Tones represent **duration** with intervals
- Chords represent **coincidence** of tones
- Phrases represent **partial order** of chords





Chords



Phrase

Outline

- **Learning from temporal data**
  - temporal data structures
  - five learning families
  - sequential pattern mining
  - time series pattern mining
  - **event pattern mining**
- Learning from spatiotemporal data
- Learning from multi-dimensional and relational data

Learning from event data

- **Clustering** with dedicated distances between events sets
- **Clustering**, **classification** and **regression**
  - traditional learners using neural embeddings or statistics from event sets
  - distances between event sets
  - discriminative patterns in event data (*next slides*): associative models

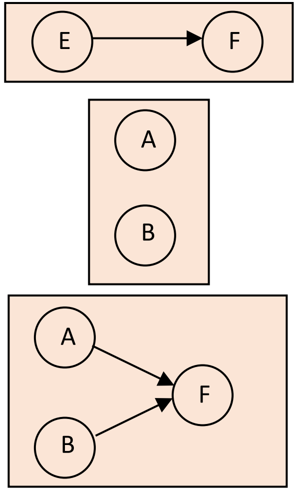
- **Distances** between event sets:
  - map events into *sequence* data by focusing on orders
  - map events into *sparse time series*
  - map events into *itemset sequence* data using time windows

  … and use distance functions for these temporal data structures

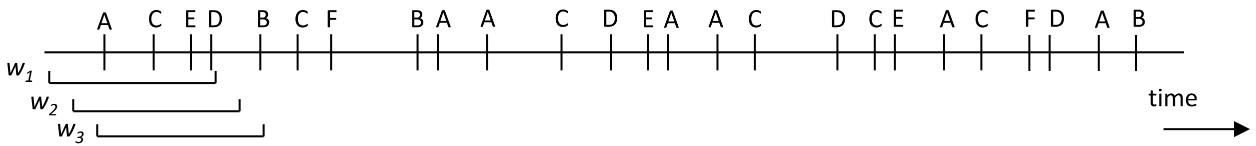
Frequent episodes

- **Episode** is an **arrangement of events**
  - **serial** episode: F appears after E
  - **parallel** episode: A appears with B in any order
  - **hybrid** serial-parallel episode: no total order

- Two formulations:
  - frequent episodes along a single sequence
    - similarities with *motif discovery*?
  - frequent episodes on a dataset of sequences
    - similarities with *sequential pattern mining*?



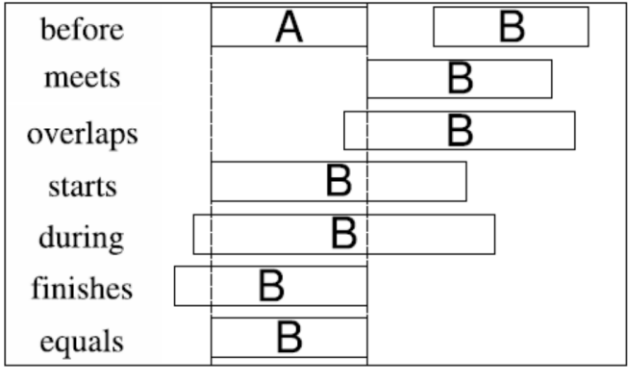
Frequent episodes



- Given a sliding windows, the frequency of an episode P: fraction of windows where P appears
- Apriori-style search, given maximum window length:
  - find frequent events (e.g., A, B, C)
  - generate candidate episodes (e.g., AB, AC, BC), counting frequencies
  - find next-level episodes
- Efficient counts:
  - no need to count all arrangements when sliding (updates)
  - WINEPI search further uses automata and hierarchies

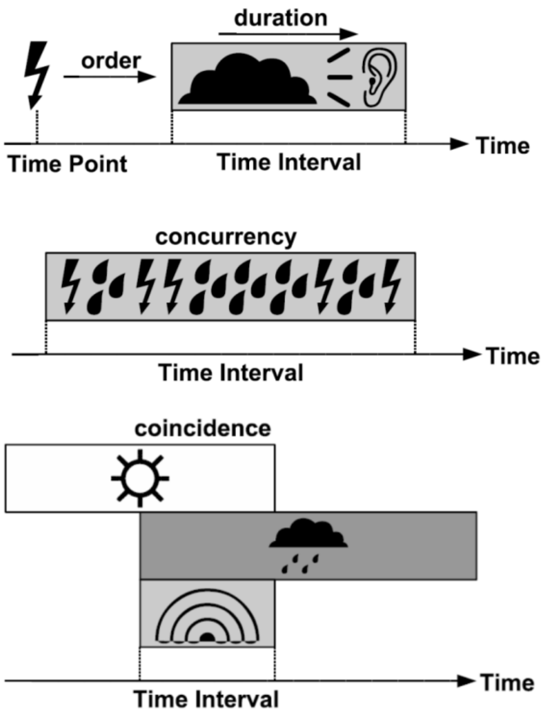
Interval series data

- A **time sequence** is a multi set of time points
- A pair of time points defines a **time interval**
- Two intervals **overlap** if there is at least one time point that lies within both intervals
- An **interval series** is a set of non overlapping time intervals
- Representation of *event data* when events have duration
- Interval series different than **feature intervals** [min *v*, max *v*]

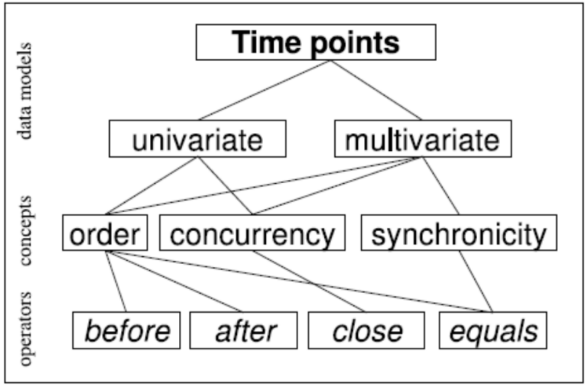


Interval series data

- **Duration**: persistence of an event along time points
- **Order**: sequential occurrence of time points or intervals
- **Concurrency**: closeness of two or more events in time
- **Coincidence**: intersection of intervals
- **Synchronicity**: synchronous occurrence of two events
- **Periodicity:** repetition of the same event with a nearly constant period

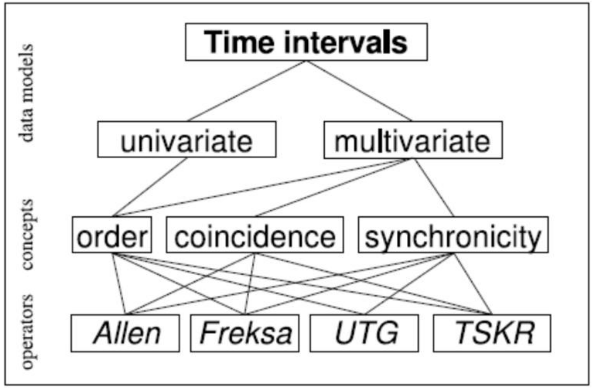


Mining interval series data



In *time series data*

- focus on *orders* or *concurrency*



In *interval series data*:

- focus on *order, concurrency, synchronicity*

Mining interval series data

- **Clustering** using dedicated distances for interval series
- **Classification** and **regression**
  - traditional learners using statistics from interval series
  - distance-based learners with distances on interval series
  - associative learners using interval patterns
- **Deep representations** using dedicated neural architectures
- Prompting engineering

- ***Note***: there are variants not only prepared for *time intervals* but **feature intervals**
  - e.g. daily variation of temperature (min-max) along a year
  - correctly interpret dependencies between pairs of features defining an interval

Patterns in interval series data

Apriori-style [Hoeppner 2001]

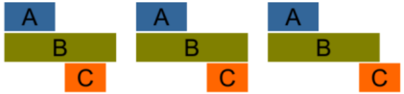
- combine two k-patterns with common $k-1$ prefix

	A	B
A	Equals	Starts
B	Started by	Equals

+

	A	C
A	Equals	before
C	after	Equals

=



	A	B	C
A	Equals	Starts	before
B	Started by	Equals	?
C	after	?	Equals

- use transitivity of interval relations to prune candidates
  - *B {contains, ended by, overlaps, meets, before} C*
  - pruned relations: {*after, met by, overlapped by, started by*}

Outline

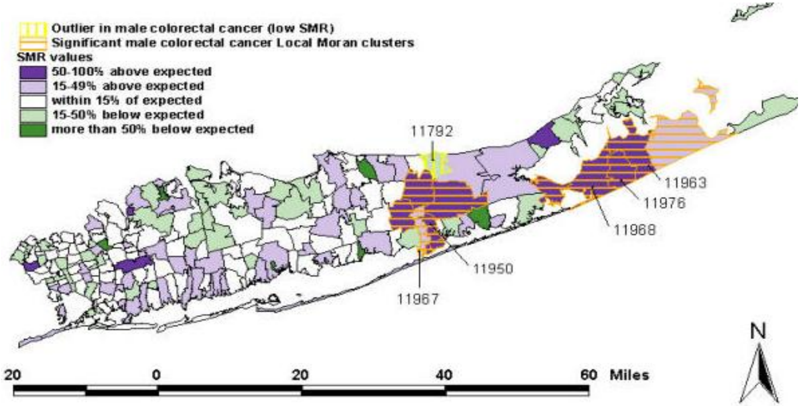
- Learning from temporal data
  - temporal data structures
  - five learning families
  - sequential pattern mining
  - time series pattern mining
  - event pattern mining
- **Learning from spatiotemporal data**
- Learning from multi-dimensional and relational data

Spatiotemporal data

- **Urban** data: mobility, emergency services, utility supply systems
- **Social** data: georeferenced activity (messaging, photos)
- **Deep space** data
- **Location-based search** data, **navigation** data (trajectories)
- **Brain** activity: functional connectivity, synchronization
- **Biological** data: organ and tissue properties
- **Image** and **video** data: moving objects
- **Geophysical** data (Earth science): atmospheric, ecosystem and seismic activity
- **Forensic** data (crime mapping)
- **Economic** (national or world-wide) data

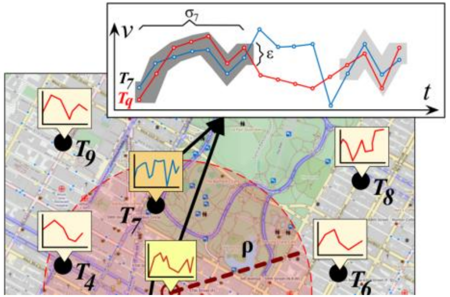
Spatiotemporal data

- **Epidemiology** data (geography of health)
  - Example: patterning of viral spread?
    Changes in transmissibility over space and time?
- **E-commerce** and marketing data
  - *What happens if a new store is added?*
    - predicting consumer spatial behaviors
    - delineating trade areas
    - analyzing market performance
  - *How sales divert geographically? Trends?*
    - changes in population, ethic-mix, and transportation network impact choices and communication with customers

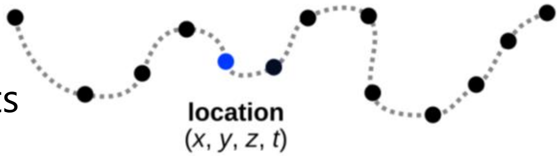


Spatiotemporal data structures

- *georeferenced observations*
  - **tabular** data with spatial attributes
    - e.g. location of individuals, geographies, stores, etc.
  - geolocalized **time series** data
    - e.g. telemetry data: measurements at a specific location



- *timestamped georeferences*
  - **trajectory** / moving object data
    - e.g. city mobility, vehicle monitoring, deep space objects
  - geolocalized **event** data
    - e.g. location of payments, health records, shopping baskets, activities of interest



- *continuous/interpolated spatial data* (e.g. geophysical maps)

Spatiotemporal data mining

The process of discovering useful, non-trivial patterns from spatiotemporal data to support decisions

- spatiotemporal data prediction and clustering
- spatiotemporal outlier discovery (discontinuities, unexpected events/trends)
- spatiotemporal pattern mining

What's **NOT** spatial data mining

- **querying** (storing and indexing) spatial data
    - ex. retrieve current traffic on the shortest path from Boston to Houston
- testing **simple hypotheses** (also referred as primary data analysis)
    - ex. female chimpanzee territories are smaller than male territories
    - SDM ≡ secondary data analysis: generate multiple plausible hypotheses
- uninteresting or **obvious patterns** in spatial data
    - ex. rainfall in Minneapolis correlated with rainfall in St. Paul (10 miles apart)

Spatiotemporal data mining

- **input**: spatiotemporal data structures and their properties
- **output**: patterns, predictors, clusters, outliers…
- **learning** (input→output): statistical foundations and algorithms
  - *descriptive learning*: pattern mining, clustering and outlier analysis
  - *predictive learning*: classification, regression and trend analysis

- *Location* helps bring rich contexts to learning
  - physical: e.g., rainfall, temperature, and wind
  - demographical: e.g., age group, gender, and income type
  - problem-specific: e.g. distance to highway or water

Input: spatiotemporal data relationships

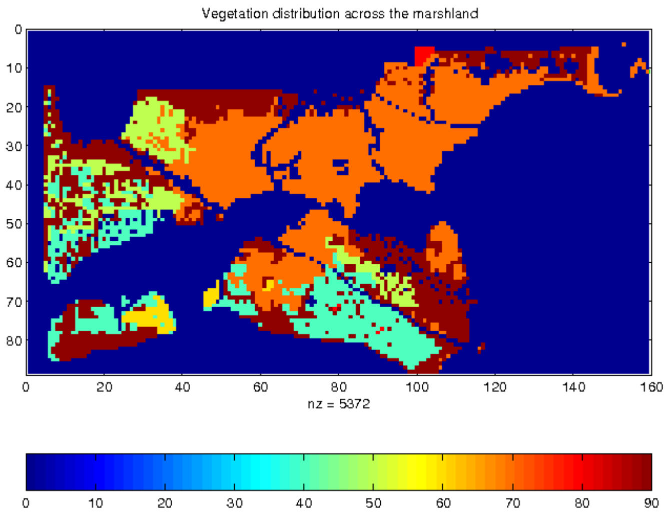
- relationships on ***non-spatial data***
  - *arithmetic*: order, correlation, etc.
  - *temporal*: duration, order, concurrency, coincidence, periodicity
- relationships on ***spatial data***
  - set-oriented: union, intersection, membership, etc.
  - topological: meet, within, overlap, etc.
  - directional: North, NE, left, behind, etc.
  - metric: distance, area, perimeter, etc.
  - dynamic: update, create, destroy, etc.
  - granularity-based →
  - shape-based and visibility

Granularity	*Elevation example*	*Road example*
Local	Elevation	On road?
Focal	Slope	Adjacent to road?
Zonal	Highest elevation in a zone	Distance to nearest road

Learning: spatial autocorrelation

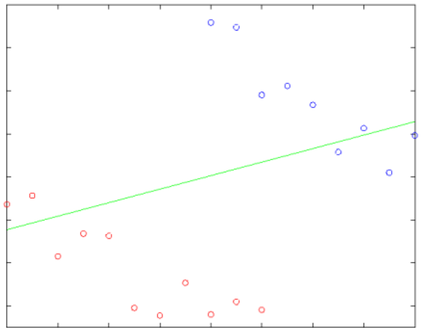
- *Classical data mining*
  - observations assumed to be independent
  - cross-correlation measures

- *Spatiotemporal data mining*
  - observations are **not independent**
    - nearby observations tend to be more similar than distant observations
  - **spatial autocorrelation**
  - spatial heterogeneity

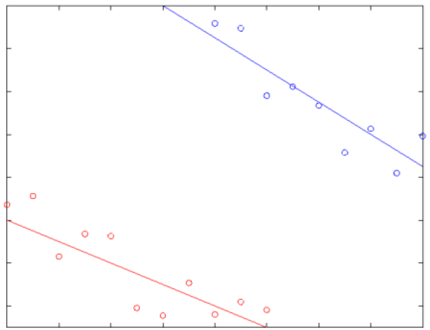


Learning: spatial splicing

- **Spatial heterogeneity:** global model might be inconsistent with regional models



global model



regional models

- *Learning different models for different spatial regions* (and time periods)
  - **slicing input** data can improve the effectiveness of SDM
  - **slicing output** models: e.g. association rule with support map

Spatiotemporal data mining: approaches

- *Option 1*: **map spatial data into multivariate data**
  - observations with **spatial variables**
    - consider latitude, longitude, elevation, radius as numeric variables
    - whenever possible adapt distances to guarantee sensitivity to such variables
      - e.g. d(location(customer), location(store))
    - cluster geographies (nominal/ordinal variables)
    - when an observation is a **time series**: combine time series statistics with spatial features
  - **trajectory data**: statistics on trajectory
  - **georeferenced event data**: statistics (spatiotemporal distribution)
  - **continuous spatiotemporal data**: e.g. Wavelet and spectral features after spatial slicing

- Do not forget: ***neural embeddings*** and *LLM* ***prompt-based features*** as alternative representations

Spatiotemporal data mining: approaches

- *Option 2:* **use spatial-sensitive distances**

  Distanced-based learners – e.g. kNN, clustering and biclustering – over:
  - observations with **spatial attributes**
    - Euclidean, walking, driving, contiguity distance between two locations
    - weighted attributes: relevance of spatial component (versus remaining)
    - when an observation is a **time series**: combine time series distance with spatial distance
  - **trajectory data**
    - dedicated distances between two trajectories
  - **georeferenced event data**
    - distance between sets of spatial events (e.g. differences between spatiotemporal distributions)
  - **continuous spatiotemporal data**:
    - matrix distances (local or global) able to accommodate misalignments

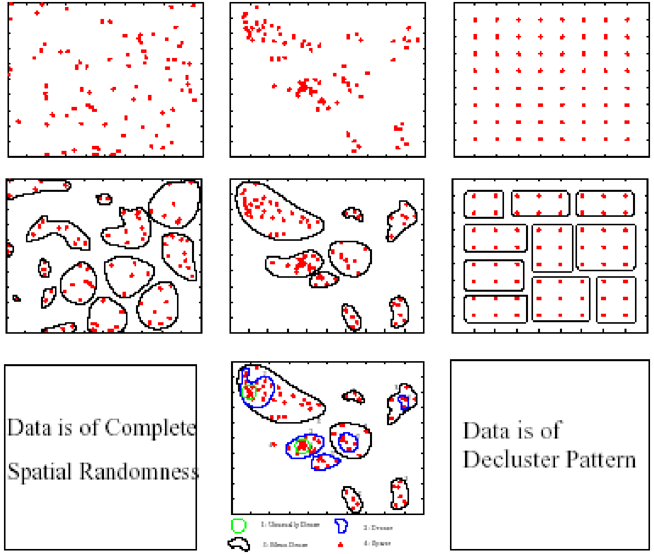
Spatial clustering

- **Different types**
  - clustering observations with spatial information (e.g., georeferenced vectors and series)
  - clustering trajectories…
- **Similarity measures**
  - distances combining non-spatial and spatial attributes
  - topological: neighborhood EM (NEM) for joint partitioning feature space and space
- **Interest measures**
  - spatial proximity
  - cartographic generalization
  - unusual density
  - consistent feature-space (nearest neighbors in same cluster)
- **Challenges**
  - temporal data (changing locations of an observation)
  - spatial constraints in algorithmic design

Spatial clustering

Variants:

- clustering geographies (maps) ensuring spatial contiguity
- weighted clustering on space and feature-space



Inputs:
Complete Spatial Random (CSR), Cluster, Decluster

Classical spatial clustering

Effective spatial clustering

Spatial outlier analysis

Types of outliers

- *global* outlier versus *spatial* outlier (contextual/local)
- georeferenced data: outliers are observations with deviant behavior and location (referred as multi-attribute spatial outliers)
- continuous spatial data: outliers are regions with unexpected values/events

Approaches

- quantitative outlier detection: scatter plot, and z-score
- graph-based outlier detection: variogram, Moran scatter plot

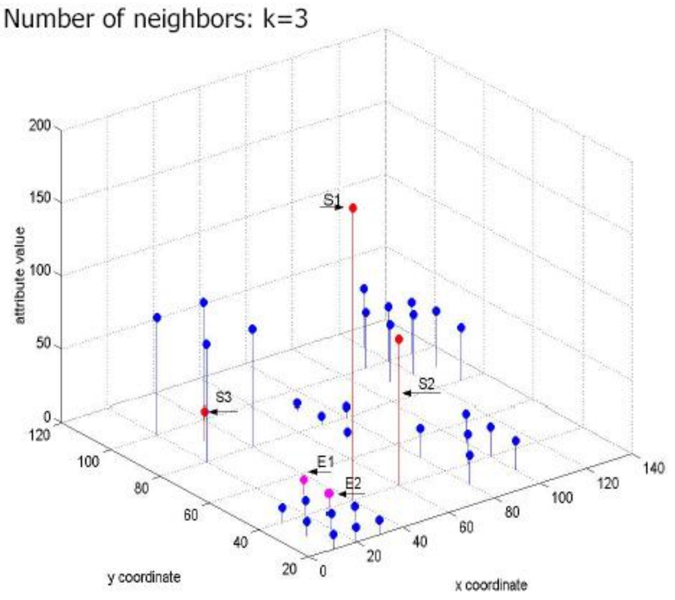
Challenges

- adequate spatial statistical tests
- collective spatial outlier detection

Spatial outlier analysis

- Traditional
  - quantitative tests (scatter spatial plots)
  - graphical tests (e.g. variogram)
- Deficiency of traditional tests
  - outliers can negatively impact nearby points
  - outliers may be ignored

- Solution
  - replace the features of the detected outlier with the median of its neighbors' values



Expected Outliers: S1, S2, S3

Outliers by traditional approaches: E1, E2, S1

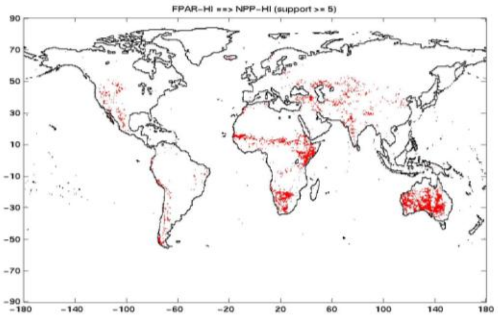
TÉCNICO+
FORMAÇÃO AVANÇADA

Spatiotemporal data mining: approaches

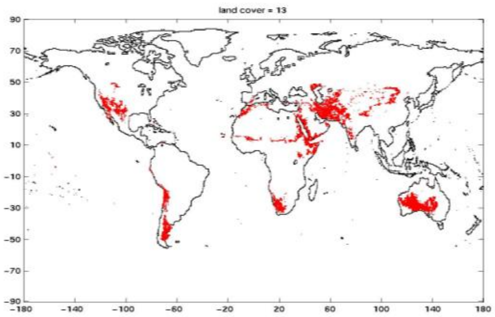
Option 3: **rely on spatiotemporal patterns**

- **discover spatiotemporal patterns**
- *option 3.1*: learn **predictive rules**
  - discriminative rules spatiotemporal pattern ⇒ label/value
  - temporal rule spatiotemporal pattern ⇒[*t*] spatioremporal
- *option 3.2*: infer a **multivariate dataset** *observations x patterns*
  - binary or real-valued defining the likelihood of pattern belonging to observation
  - use the dataset to:
    - to group observations (clustering)
    - to learn classification and regression models
    - to find high-order patterns

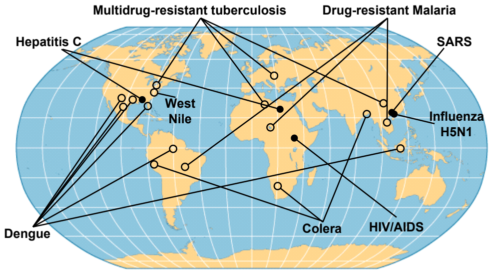
Spatiotemporal patterns



FPAR-Hi ⇒ NPP-Hi
(sup=5.9%, conf=55.7%)



grassland/shrubland areas



• Newly emerging diseases o Re-emerging diseases

Emerging patterns and spatiotemporal associations

- urban dynamics
- consumer/user habits
- public health (e.g. infectious diseases)
- homeland defense

emerging road traffic
congestions in the city

TÉCNICO+
FORMAÇÃO AVANÇADA

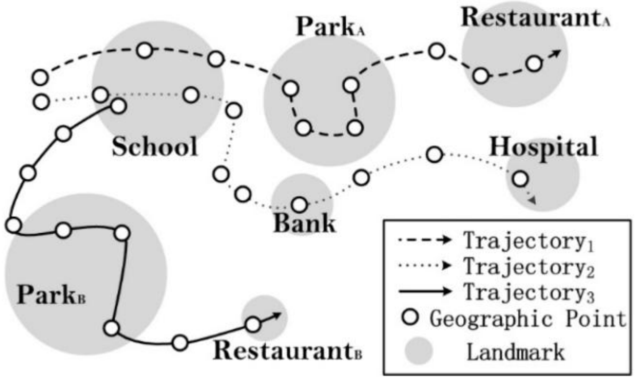
Spatiotemporal patterns

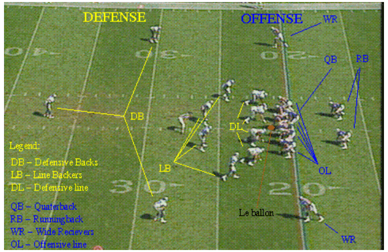
Open field with thousands of possibilities

- **flock patterns** on trajectory data
- **moving patterns** from evolving spatial clusters
- **colocation patterns** from event data
- patterns whose cause and consequence do not happen colocated or at the same time
    - spatial distance or temporal delay for the consequence to show up

… less-trivial (yet relevant) applications

- *ecology* (e.g. migration, relocation patterns)
- *games (e.g.* game tactics)





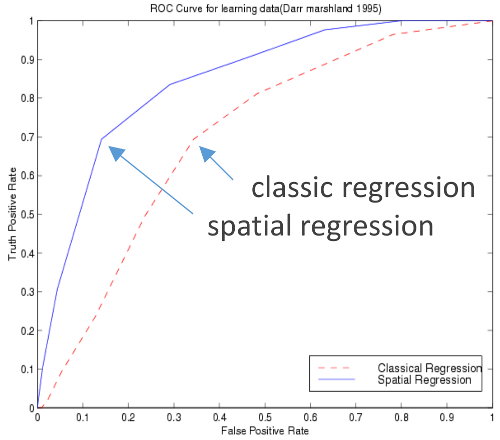
Spatiotemporal data mining: approaches

Option 4: **dedicated approaches**

- spatial slicing over classical approaches to turn learning sensitive to space
    - guarantee sensitivity to spatiotemporal dynamics
- annotate events or group observations in accordance with spatial partitions
    - *examples*: *Bayesian classifiers* with probabilities conditional to class and location, *tree-based classifiers* with info gain also dependent on location
    - grouping criteria:
        - circles centered at reference features
        - gridded cells
        - min-cut partitions
        - Voronoi diagrams
- dedicated learning algorithms to solve new tasks (e.g., location prediction)

Location-aware prediction

- **Location-aware regression**
  - spatial autoregressive model (SAR)
    - linear Regression $z = A\beta + \varepsilon$
    - spatial Regression $z = \rho W y + A\beta + \varepsilon$
      - models spatial autocorrelation using W (continuity matrix)
  - geographically weighted regression (GWR)
- **Location-aware classification**
  - logistic SAR and GWR
    (similarly as classic logistic regression)
  - Markov random fields (Bayesian view of the neighborhood region)

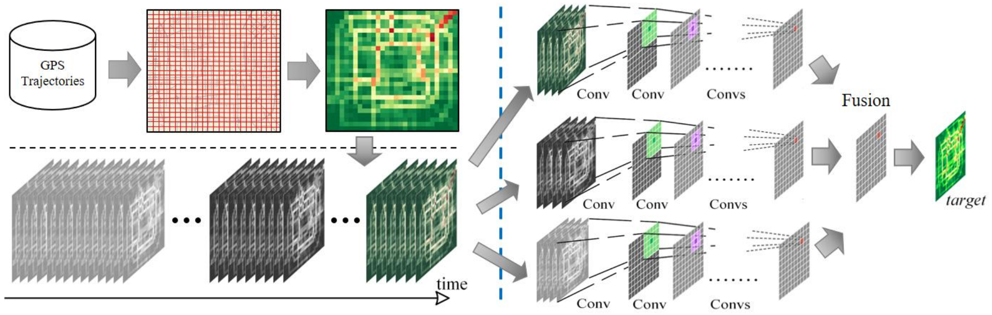


ROC Curve for testing data comparing linear and spatial regression (spatial is better)

Spatiotemporal data mining: approaches

Option 5: **deep learning**

- **spatial locality** via *convolutional* layers
- **time dependency** (short/long-range) using *recurrency*, *convolutions* or *attention*
- *hierarchical layering* for **multi-scale spatial** and **temporal** representations



- **graph** neural networks to represent non-Euclidean spatial structures
- modular combination of spatial and temporal neural components into a unified architecture
- **parameter sharing** in space and time to improve generalization
- *spatiotemporal factorization* to reduce complexity

Option 6: **LLM prompting**

Outline

- Learning from temporal data
    - temporal data structures
    - five learning families
    - sequential pattern mining
    - time series pattern mining
    - event pattern mining
- Learning from spatiotemporal data
- **Learning from multi-dimensional and relational data**

Motivation

- Disperse/distributed data in private and public organizational settings often consolidated using **multi-dimensional data structures** (also known as data warehouses)
- Most real-world organizational data stored follows a **relational data structure**

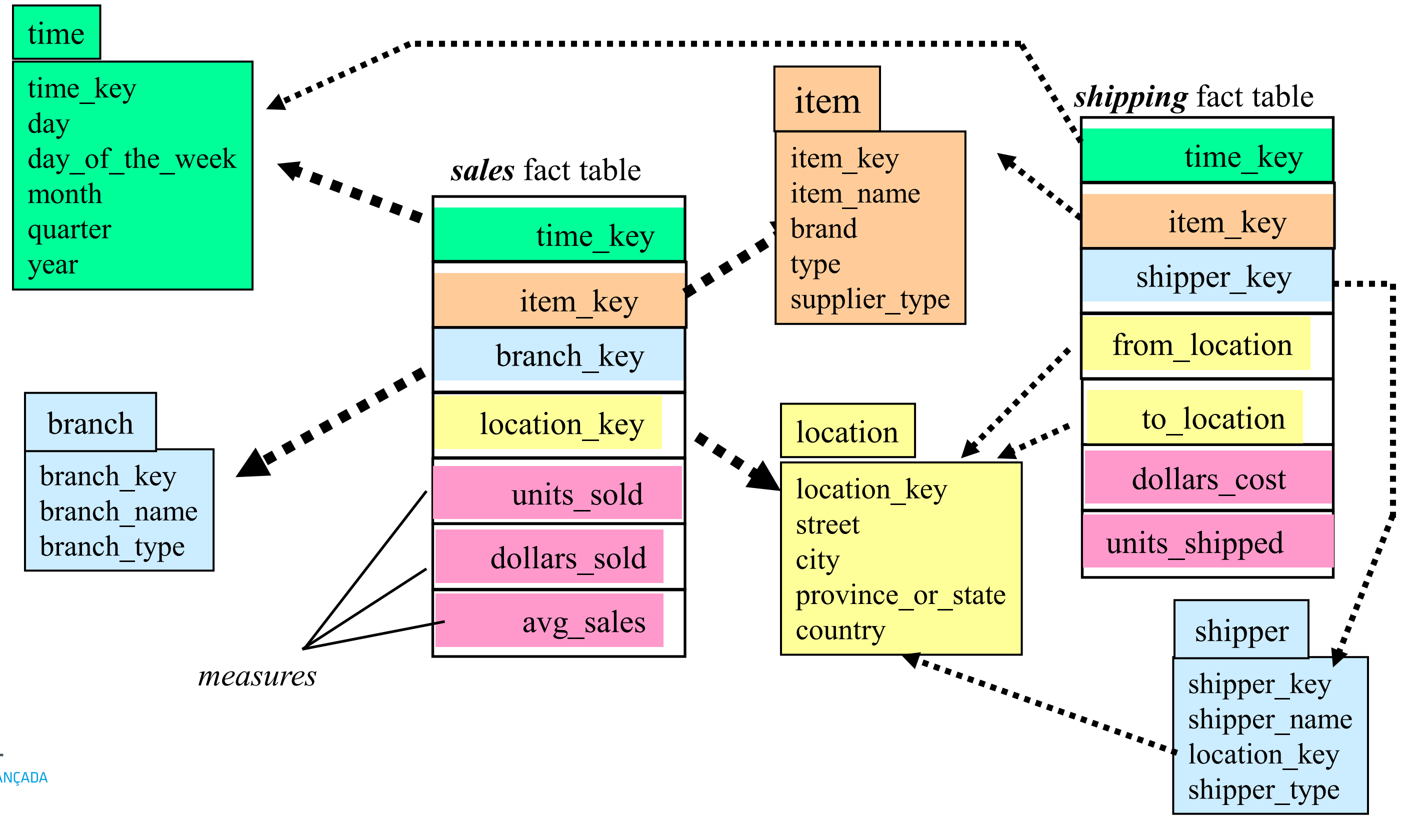
  - Business
  - Commerce
  - Education
  - Healthcare
  - Turism
  - Banking
  - Transport
  - Public administration

- This poses a **key question**: how to mine multi-dimensional and relational data?

What is a data warehouse?

- Data warehouse
  - database maintained separately from the organization's operational database(s) for consolidated, historical data analysis and decision making
- Data warehouse composition:
  - **dimension tables**: such as item, supplier, location or time
  - **fact table**: contains measures and keys to each related dimension table
- *Why a separate data warehouse?*
  - databases: tuned for OLTP (access methods, indexing, concurrency, recovery)
  - warehouses: tuned for OLAP (complex queries, consolidation)
  - missing data: operational DBs do not typically maintain all historical data
  - data consolidation: aggregation, summarization of heterogeneous data
  - data quality: reconciliation of sources with inconsistent data representations, codes, formats

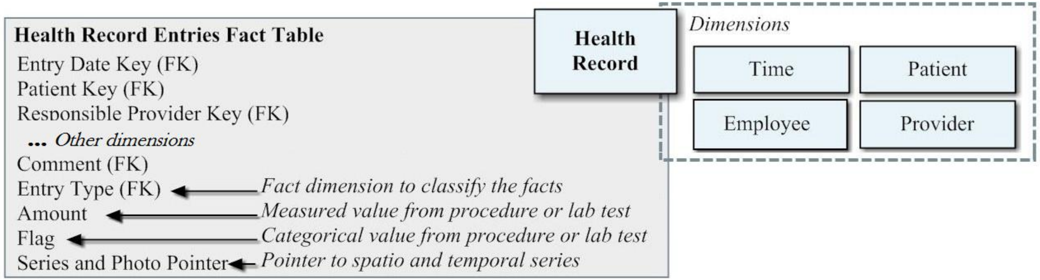
Multi-dimension: facts and dimensions



Mining multi-dimensional data

Example: electronic health records

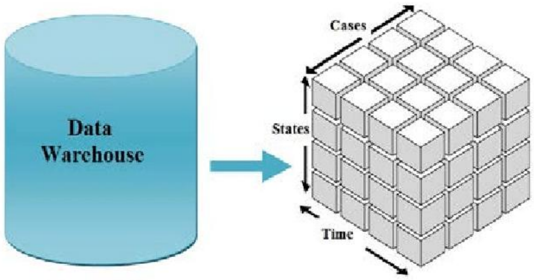
- Health-record as a central fact table (high multiplicity of measures) linked to multiple dimensions (*date*, *patient*, payer, provider, prescription, location)



- **Mapping *multi-dimensional data* ⟹ *event sequences***
  - *aggregation* dimension (*patient*) and *date* dimension to compose *repository of events (patient, fact-measure, value, timestamp)*
- **Learning from event sequences**: recall the multiple options! For instance:
  1. Discover event patterns (e.g. frequent arrangements)
  2. Learn predictive models (e.g. associative classification)

Learning from multi-dimensional data

- Equals learning from ***tensors extracted from a fact table***
  - If the tensor is three-dimensional (cube)
    - triclustering
    - generative modeling (HMMs, DBNs)
    - multivariate time series data analysis
      - pattern and motif discovery
      - classification and regression
    - temporal network data analysis
  - If the tensor has $k>3$ dimensions (hypercube)
    - k-way subspace clustering
    - tensor decomposition
    - mapping to more adequate (temporal) data structures
      - multi-dimensional space transformations
      - denormalization + dimensionality reduction



TÉCNICO+
FORMAÇÃO AVANÇADA

Example: MD sequential patterns

- Multi-dimensional (MD) sequential pattern mining as an illustrative case: integrates multi-dimensional analysis and sequential pattern mining
- Recap: sequential pattern mining to find frequent subsequences

10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

<a(bc)dc> is a *subsequence* of <a(abc)(ac)d(cf)>

Given *support threshold* θ=2,
<(ab)c> is a *sequential pattern*

- MD sequence database: combine dimensional information to the itemset sequence produced from the fact entries of a given object (split dimension)
  - *example*: if θ=2, P=(group=*,city=Chicago,age=*,qeq=<bf>) is a MD pattern

cid	Cust_grp	City	Age_grp	sequence
10	Business	Boston	Middle	<(bd)cba>
20	Professional	Chicago	Young	<(bf)(ce)(fg)>
30	Business	Chicago	Middle	<(ah)abf>
40	Education	New York	Retired	<(be)(ce)>

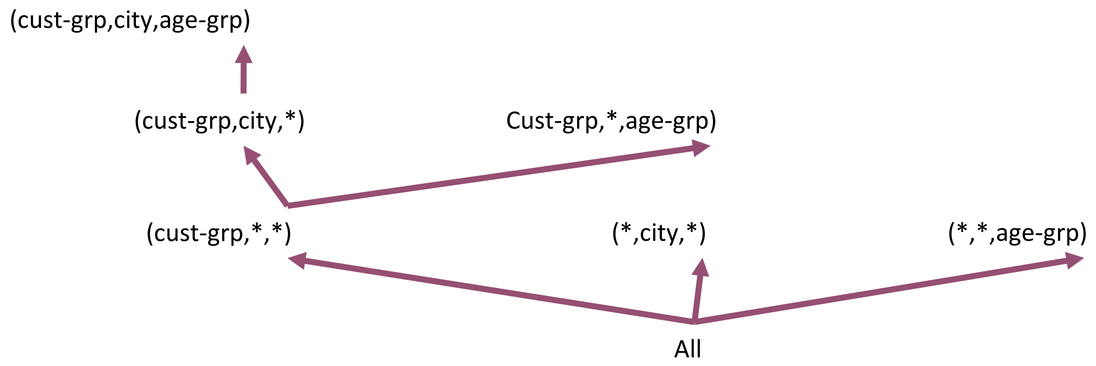
Example: MD sequential patterns

Mining MD-patterns – e.g. (*,Chicago,*)

- first project seq. databases – e.g. <(bf)(ce)(fg)> and <(ah)abf> for (*,Chicago,*)
- find seq. patterns in projected database – e.g. P=(*,Chicago,*,<bf>)

cid	Cust_grp	City	Age_grp	sequence
10	Business	Boston	Middle	<(bd)cba>
20	Professional	Chicago	Young	<(bf)(ce)(fg)>
30	Business	Chicago	Middle	<(ah)abf>
40	Education	New York	Retired	<(be)(ce)>

cid	MD-extension of sequences
10	<(Business,Boston,Middle)(bd)cba>
20	<(Professional,Chicago,Young)(bf)(ce)(fg)>
30	<(Business,Chicago,Middle)(ah)abf>
40	<(Education,New York,Retired)(be)(ce)>



Relational data structures

How to mine relational databases?

- **naïve solution**: bringing all information to a single table
  - e.g. *customer* table where we combine as much info as possible
- **problems**:
  - redundancies
  - feature dependence
  - how to deal with the multiplicity of orders per customer?
    - one line per '*order*' → analysis results will be about orders, not customers!

ID	*Name*	*First Name*	…	*Response*	*Delivery mode*	*Payment mode*	*Store size*	*Store type*	*Location*
3478	Smith	John	…	Y	regular	cash	small	franchis	city
3478	Smith	John	…	Y	express	check	small	franchis	city
…	…	…	…	…	…	…	…	…	…

Relational data mining

Relational data mining (RDM)
- analysis of data distributed in multiple relations

Multiple paradigms: classic, deep learning, prompt-based…
… yet many RDM principles come from **inductive logic programming** (ILP)
- ILP concerned with *finding patterns expressed as logic programs*
    - initially: data modelling/description/synthesis
    - in recent years: whole spectrum of data mining tasks
- ILP successes in commercial settings and scientific fields such as:
    - chemistry/biology (toxicology, nuclear magnetic resonance spectra)
    - traffic accident data
    - survey data in medicine
    - ecological biodegradation rates

Example: relational patterns

- *Relational patterns* involve multiple relations from a relational database
  - Typically stated in a more expressive language
    - relational classification rules
    - relational regression trees
    - relational association rules

```
IF Customer(C1,N1,FN1,Str1,City1,Zip1,Sex1,SoSt1, In1,Age1,Resp1)
  AND order(C1,O1,S1,Deliv1, Pay1)
  AND Pay1 = credit_card AND In1 ≥ 108000
THEN Resp1 = Yes
```

- Relation in a relational database: predicate in predicate logic
- Relational pattern can be expressed in a subset of first-order logic

```
good_customer(C1) ←
  customer(C1, N1,FN1,Str1,City1,Zip1,Sex1,SoSt1,In1,Age1,Resp1)∧
  order(C1,O1,S1,Deliv1,credit_card)∧ In1 ≥ 108000
```

Example: relational association rules

LIKES

KID	OBJECT
Joni	ice-cream
Joni	dolphin
Elliot	piglet
Elliot	gnu
Elliot	lion

HAS

KID	OBJECT
Joni	ice-cream
Joni	piglet
Elliot	ice-cream

PREFERS

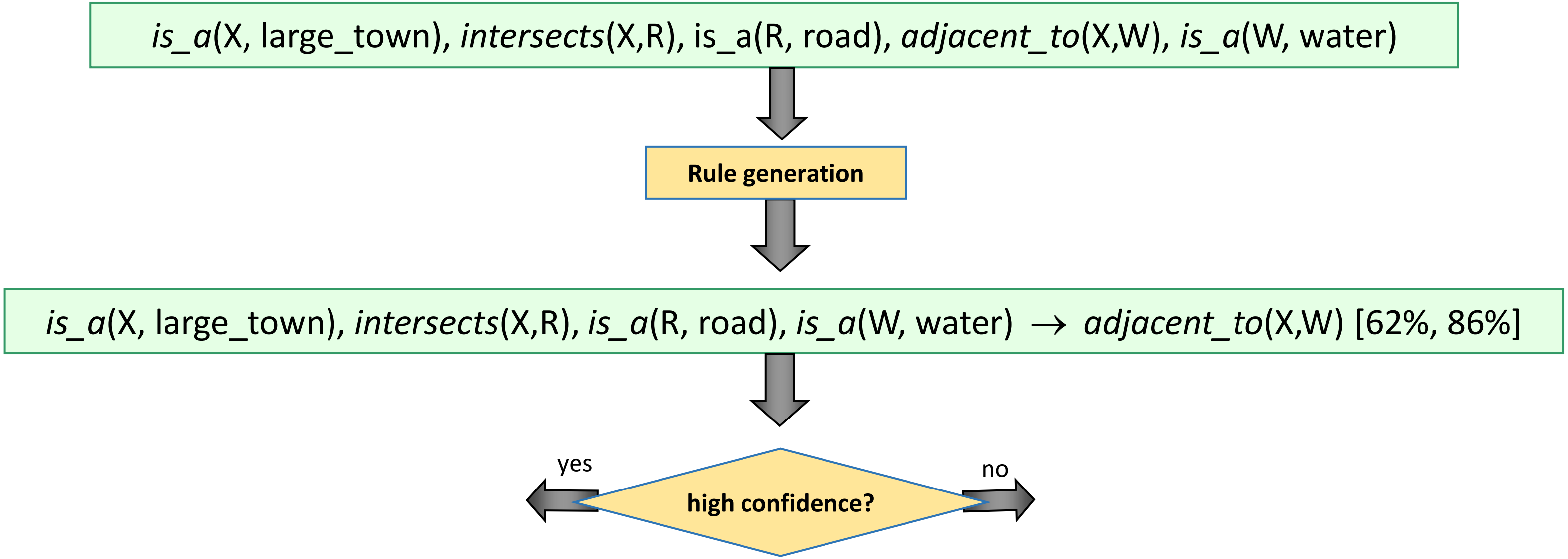
KID	OBJECT	TO
Joni	ice-cream	pudding
Joni	pudding	raisins
Joni	giraffe	gnu
Elliot	lion	ice-cream
Elliot	piglet	dolphin

likes(KID, A), **has**(KID, B) → **prefers**(KID, A, B) (70%, 98%)

WARMR: iteratively generate candidate k-atomsets from (k-1)-atomsets *until no more large atomsets are found*

likes(KID, piglet), likes(KID, ice-cream) atomset

Example: relational association rules (ILP)

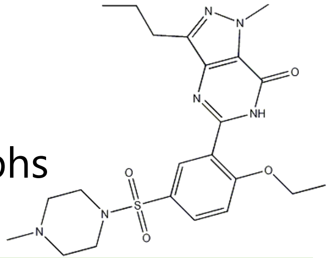


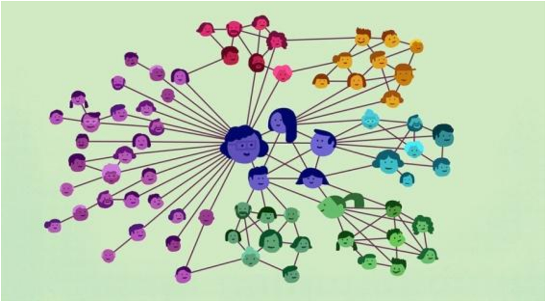
Outline

- Learning from temporal data
    - temporal data structures
    - five learning families
    - sequential pattern mining
    - time series pattern mining
    - event pattern mining
- Learning from spatiotemporal data
- Learning from multi-dimensional and relational data

Many other data structures…

- **image**, **text**, **multimodal**… to be explored in DL and LLM DASH modules
- yet we overlook a pervasive one… **network data** (graphs)
  - most real-world systems can be represented as static or dynamic graphs
    - chemical compound structures
    - biological networks
    - cities (multimodal transportation networks)
    - social networks (messaging, content interaction)
    - ecosystems
    - …

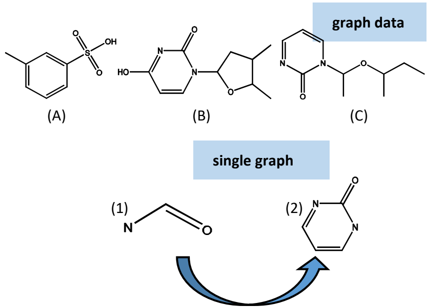




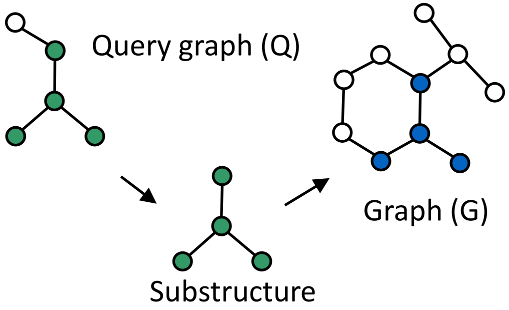
  - the large *network science* community is dedicated to study systems as graphs
    - and you guessed it right: strong intersection with data science

Example: graph pattern mining

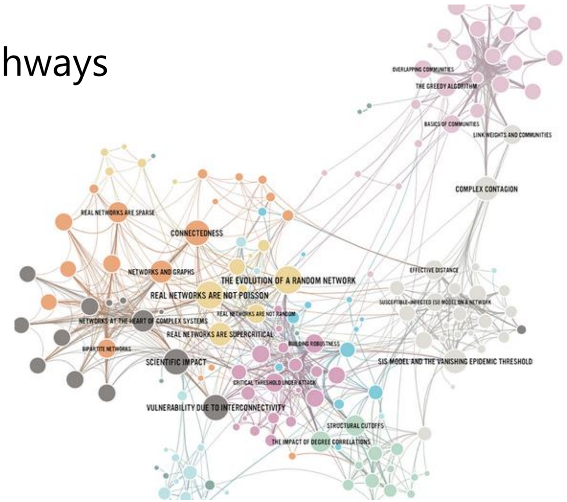
- The five learning approaches can be as well considered when learning from graphs
- Yet for now… a sneak peek to a pattern-centric stance
- Pattern discovery in graphs follows similar principles as we saw for time series
    - graph dataset: find frequent subgraphs
    - single large graph observation: find frequent components



- We can as well query graphs:
  find all graphs containing a given query
- All the studied pattern metrics (lift, support, significance) are key here



Example: graph pattern mining



- Bioinformatics: gene networks, protein interactions, metabolic pathways
- Software engineering: program execution flow analysis
- Web graphs, XML structures, semantic web, information networks
- Social networks, web communities, tweets, computer networks
- Chem-informatics: mining chemical compound structures
- Across domains, for different ends:
  - knowledge acquisition
  - at the core of graph indexing and graph similarity search
  - building blocks for graph prediction, clustering, compression, comparison, correlation

Summary

- **Temporal data are ubiquitous**: measuring bio, individual, organizational and societal systems
- **Application** domains include the analysis of log data, medical records, user behavior, shapes, musical sheets, sensor data, text, Twitter, speech, user actions
- Mining temporal data commonly rely on one of **four major paradigms**:
  - **feature extraction** followed by classic algorithms
  - **distance-based** approaches with **elastic distances** (e.g. DTW)
  - pattern mining following by **associative analysis**
  - **generative models** (e.g. temporal neural networks)
- **Symbolic** and **spectral representations** (e.g. SAX, Fourier and Wavelet transforms) offer relevant abstractions to handle the idiosyncrasies of signal data
- **Pattern mining** in temporal data
  - **sequential pattern mining** in sequence data
  - **biclustering**, **triclustering** and **motif discovery** in (multivariate) time series data
  - discovery of **episodes**, **partial orders** and event arragments over event sequences

Summary

- Spatiotemporal data is pervasive: urban, social, healthcare, geophysical, deep space
- Learning from **spatiotemporal structures:** observation is a feature-vector, time series, trajectory
  - observations are not independent (dependent): **spatial auto-correlation** and **heterogeneity**
  - tasks: spatiotemporal pattern mining, outlier analysis, prediction, clustering
- Principles can be placed to learn from spatiotemporal data
  - **spatial slicing** to learn spatial-aware descriptive and predictive models
  - **spatial-aware distances** to weight non-spatial and spatial attributes
  - discovery of **spatiotemporal patterns** (co-locations, associations, emerging patterns, flocks)
- Real data structures in public and private orgs are relational or multi-dimensional
- **Mining relational** and **multi-dimensional data** focused on a table (fact) of interest
  - apply subspace clustering, generative modelling, multivariate time series analysis and tensor
  - map into more adequate (**temporal**) **data structures** for the sequent application of classic TDM
- Relational data mining (RDM) dedicated to analyse data in multiple relations using **ILP principles**

Thank you!

Rui Henriques

rmch@tecnico.ulisboa.pt