Mining Data with Quantum-like Contextuality

Chenchao Ding

Dec. 12, 2024

- Acceptable projects for this course:
  - A project that involves mining of a dataset of decent size (small toy datasets such as the breast cancer data set we used to build a decision tree don't count), in which you apply techniques you have learned in this class for cluster analysis, classification, association rule mining, recommendation systems, etc to address a well-defined problem.
  - It is ok if you use a database available on Kaggle. Considering that many datasets on Kaggle already have notebooks written by fellow Kagglers, your work needs to be sufficiently different from the existing ones.
  - You may also propose a new data mining problem. For this case, little results or no results may be acceptable. However, you need to provide a clear statement of the problem and motivation; you also need to have a detailed plan about how you are going to collect the data (or where to get the data), and your approach to solve the problem.

Motivation: 20 Questions

A famous parlor game – one answerer A vs. one questioner Q.

- A chooses some object a and keeps it in mind;
- Q asks a series of questions p_i to guess the hidden object a;
- A responds Yes or No (boolean type $\mathbf{2}$) to each question p_i;
- The "train-data" is *a collection of predicates* $\mathcal{D}_{\text{train}} = \{(p_i, p_i(a))\}$, where:

$$p_i : \forall(a \in \text{Obj}) \to \mathbf{2}$$

- The "trained model" is a collection of candidate objects $A = \{a_k\}$ satisfying $\mathcal{D}_{\text{train}}$;
- The "test-data" is saved in advance $\mathcal{D}_{\text{test}} = \{(q_j, q_j(a))\}$ satisfied by object a;
- The "loss" is binary: either $\forall j. q_j(a_{\text{guess}}) = q_j(a)$ (win) or $\exists j. q_j(a_{\text{guess}}) \neq q_j(a)$ (lose).

It is a minimal structure that captures:

i. the essential elements of supervised learning;

ii. that "train-data" measured and collected *on-the-fly* (non-i.i.d.).

Variant One: Nothing

If A thinks of *nothing* instead of something a prior to query:

- A gives *random* but consistent answer $p_i(a)$ to each query p_i;
- A accepts whatever a_{guess} is from Q;
- a_{guess} is *manufactured* via the interation between A and Q;

What is crucial here is Q's *misrecognition* of its own subjective position:

- A is *sujet supposé savoir* (*supposed* by Q to know what a is);
- As long as A does not reveal the "truth" that nothing is picked at the begining...
- ...Q can obtain and maintain an "observer's safe distance".

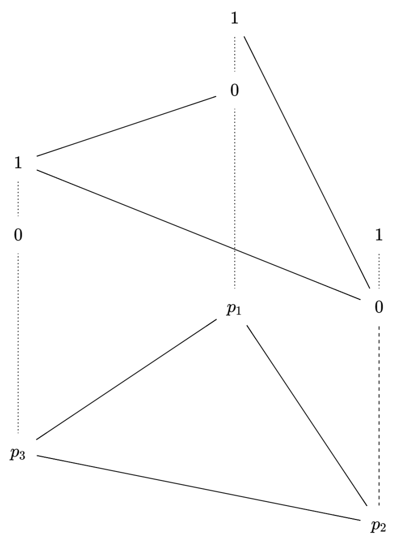
I am always-already in the picture I see in the guise of a blind spot.

Variant Two: Cheating

If A thinks of somthing a but *switches to something else* during the game:

- Q packs 2 questions in a "context" and query simultaneously $C = \{p_i, p_j\}$;
- A is "caught cheating" if $\exists i\,j\,k.\, p_k(a) \in \{p_i(a), p_k(a)\} \neq p_k(a) \in \{p_j(a), p_k(a)\}$;
- Q is forced to conclude the the globally consistent a does not exist at all.

Contextuality arises with a family of data which is **locally consistent**, but **globally inconsistent**.



Recall "pairwise comparisons" in modelling human preference.

"I regret/I changed my mind on x when seeing it put together with y."

noise & malcalibration: "bug" $\Rightarrow$ "feature"

	Object	Classical view	"Quantum" view
20 questions	answerer	vanilla game	nothing/cheating
Physics	system	hidden-variable model	contextuality
Ontology	reality	complete closure	incomplete disclosure
PL theory	expression e	$f_1(e) \otimes f_2(e) = (f_1 \otimes f_2)(e)$	non-compositionality
Logic	predicates	global consistency	global inconsistency
Learning	source	supervised model	?

There are some crucial presuppositions of classical view:

- **Leibniz's Law** (observational equivalence): $x = y \leftrightarrow \forall P[P(x) = P(y)]$. Identity of an object is guaranteed by a collection of predicates (or attributes, observables).
- **Principle of realism** (complete reality): unconditional assertion of an objective reality independent of subjective position and prior to measurement protocols (e.g. contexts).
- **Principle of representationalism** (incomplete knowledge): model does not seek to "outperform" the reality itself, only asymptotic approximation, always has *loss*.

In "quantum" view, data are phenomena produced via the interaction of the observer and the observed *on-the-fly*. The objective source of data (hidden object a) does not (fully) exist.

The matheme of classical view: **[object = (data - noise) = (model + loss)]**.

The matheme of "quantum" view: **[data = (object + noise)]**.

Formalization: Sheaf-theoretic Approach

Data as observables (a.k.a. attributes, predicates, questions) and outcomes:

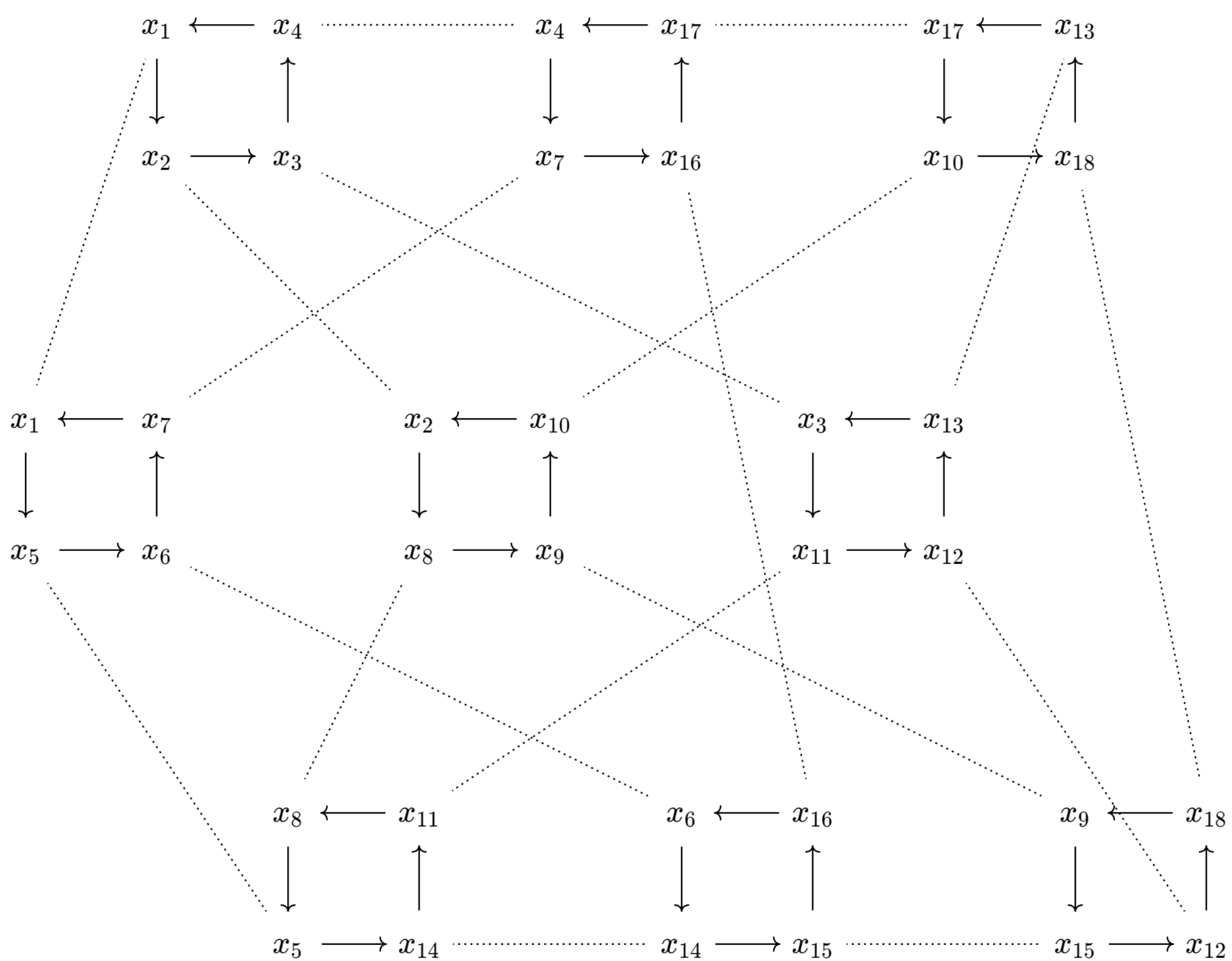
$$
\begin{aligned}
\mathcal{D} &= \{(x_i, y_i)\} \\
&= \{(x_i, x_i(s))\} \\
X &= \{x_i\} \\
x_i &: \forall (s \in \mathcal{S}) \rightarrow \mathcal{O}
\end{aligned}
$$

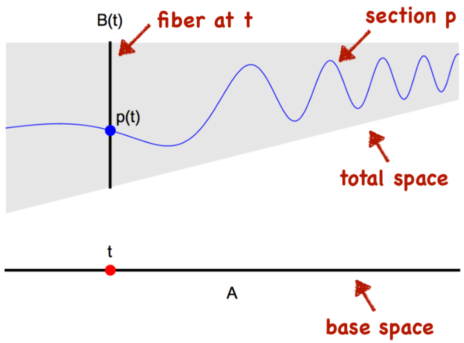
Base space X has topological/functorial structure. Each context C belongs to a measurement cover $\mathcal{M}$ of base space X:

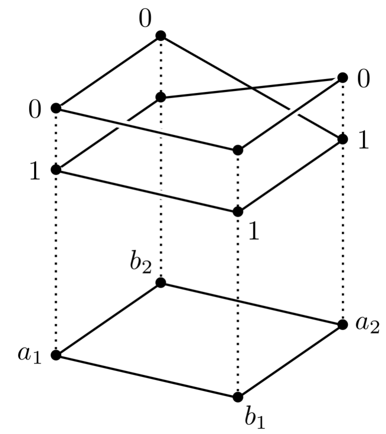
$$
\begin{aligned}
\mathcal{M} &\subset \mathcal{P}(X) \\
\bigcup_{C \in \mathcal{M}} C &= X
\end{aligned}
$$

- Measurement protocol: query is performed (therefore data are collected) "context by context".
- It can be visualized as a hypergraph, or a database schema with overlapping attributes.

Example: Kochen-Specker Configuration







$$p \;:\; \forall(t \in A) \to B(t)$$

Global consistency (global section): a closed path traversing all the fibers *exactly once*, assigning a unique value to each observable.

	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
(a_1, b_1)	1	0	0	1
(a_1, b_2)	1	0	0	1
(a_2, b_1)	1	0	0	1
(a_2, b_2)	0	1	1	0

$$\begin{aligned} X &= \{a_1, a_2, b_1, b_2\} \\ \mathcal{M} &= \{\{a_1, b_1\}, \{a_1, b_2\}, \{a_2, b_1\}, \{a_2, b_2\}\} \\ \mathcal{O} &= \{0, 1\} \end{aligned}$$

For a more detailed formal definition see the final report.

Compatibility with Reinforcement Learning

Contextuality	Reinforcement Learning
inexistence of globally consistent reality	unknown ground truth
primacy of data over object	reward instead of loss
observer-observed interaction	agent-environment interaction
online (non-i.i.d.)	online + offline (non-i.i.d.)
casuality + retrocasuality	casuality

Contextuality is a feature of empirical data, not of model! (as a special "noise" honestly)

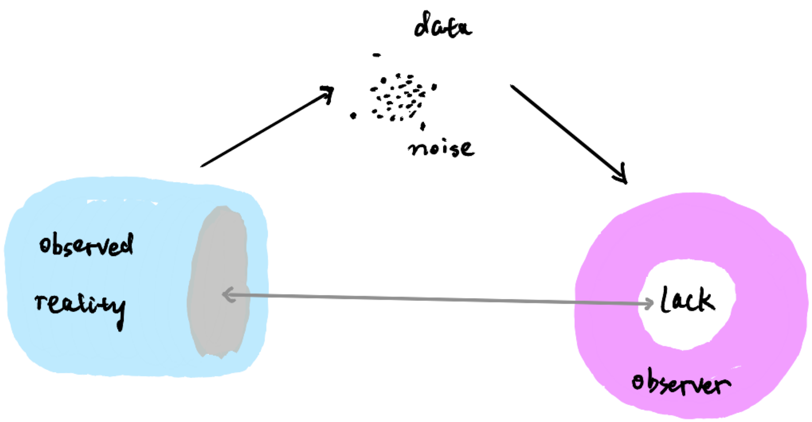
In general:

- state: topological space (bundle diagram) witnessing and maintaining contextuality.
- action: $C_t \in \mathcal{M}$ at each step (decide which context to measure next).
- reward: depends on the learning goal.

Contextuality data can be "noisey/lossy environment feedbacks" in RL, with a radical turn:

- such "noise" is an indication of agent's inclusion in the environment.
- …therefore reducing "noise" restores observer's safe distance and naive realism.

More Meta-description

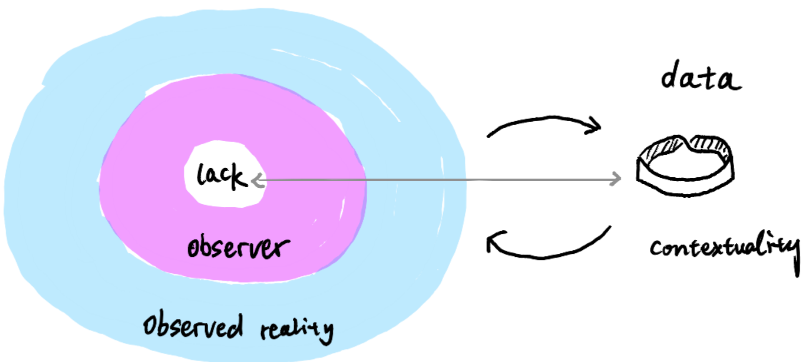


observer's safe distance

lack (loss) indicates an impenetrable

excessive part in observed reality

⇒ epistemological limit (irreducible loss)



contextuality indicates a "shared lack"

inclusion of observer in observed reality

⇒ ontological incompleteness

Thesis: shared lack is pervasive but elusive (recall variant one of 20 questions), while contextuality data "exposes"/"reifies" it and renders visible its computational potential.

So how to utilize contextuality data? It seems to be quite an complex and open question...

20 Questions, *Encore* (or 20,000 Questions)

The user plays as the answerer, the recommendation system plays as the questioner!

User who know "less" (Nothing & Cheating)

User has fuzzy preference, or no preference at all. There is no preference prior to recommendation – preference is *manufactured and refined* via the cooperation of the user and the system.

The system is becoming a "prosthesis" of the user not only to show but also to develop his preference. The system knows more than the user about his own preference.

Potentially interesting problems involving contextuality data

- identify users with fuzzy preference (witnessing more inconsistency in contextuality data);
- identify "high/low score items in most context", "context sensitive items"...;
- identify "perfect/poor context where most items got high/low score";
- identify causal structure among different items ("I regret" and so on);
- detect broken of compositionality: items get higher/lower score when put in larger/smaller context.