Is this correct? Let’s Check!

Omri Ben-Eliezer  Dan Mikulincer  Elchanan Mossel  Madhu Sudan
Is societal knowledge robust?

• Why ask this question?
  • Builds on error-prone processes
    • Collecting Data
    • Analyzing it
    • Combining results
  • Last is especially problematic/interesting: Knowledge is cumulative!!
    • Accumulation can be very bad for errors!!!!!
• There must exist error-correcting processes
  • What are they? How do they work? How well do they work?
Lebesgue’s Mistake

• In 1904 Lebesgue proved the following theorem:
  “A projection of a measurable set is measurable”
Lebesgue’s Mistake

- According to Google Scholar the paper has 303 citations.
Lebesgue’s Mistake

- According to Google Scholar the paper has 303 citations.
- Some citations prior to 1917.
Lebesgue’s Mistake

- According to Google Scholar the paper has 303 citations.
- Some citations are from before 1917.
- In 1917 Suslin discovered a counterexample:
  
  “There exists a projection of measurable set which is not measurable”

- Happy ending: The field of “Descriptive set theory” was born.
- Did the mistake propagate?
Today

Question
Can we guarantee that the effects caused by a single error do not propagate?
Cumulative Knowledge Process

- In the paper we model the process of accumulating knowledge.
- Main properties:
  - New “units of knowledge” build upon previous units.
  - Errors are sometimes introduced and may propagate forward.
  - Errors can be checked and removed from the process.
- We study structural properties of the process.
The Model
Representation of Knowledge

• **Ideally**: Knowledge is stored as **Directed Acyclic Graph (DAG)**.
  - Vertices represent units of knowledge
  - Edges represent dependence or “inherited knowledge”.
  - E.g. a paper cites several papers.

• Would require a proper model of knowledge clustering.
• **Simplified notion**: Knowledge is **represented as a tree**.
The Model

- **The Knowledge DAG Tree**: In the Cumulative Knowledge Process (CKP) knowledge units are modeled as a tree.
- A node represents a single "unit of knowledge" and edges represent the relation of "building upon existing knowledge"
- Each node has:
  - A hidden state conditionally true(CT)/conditionally false(CF)
  - A public state proclaimed true(PT)/proclaimed false(PF)
  - A node is considered to hold true knowledge if all ancestors are CT
Accumulating Knowledge

- At each time $t \geq 0$, we have a knowledge tree $T_t$ with associated labels.
- At time $t + 1$ a new node is added to the tree, by choosing a random proclaimed true (PT) parent.
- Parents are chosen according to the preferential attachment model.
  - The more PT children a node has the more likely it is to generate new knowledge.
- A new node is always proclaimed true.
Injection of Errors

- Recall: nodes also have hidden states.
- The hidden label of a new node is determined randomly:
  - Parameter $\varepsilon$: with probability $\varepsilon$ the new node is CF and otherwise CT.
Injection of Errors

- Recall: nodes also have hidden states.
- The hidden label of a new node is determined randomly:
  - Parameter $\varepsilon$: with probability $\varepsilon$ the new node is CF and otherwise CT.
Aside: Probabilistic Models

- Quote from unknown source*
  - “All models are wrong. Some are useful”
Checking for Errors

- Checks may be performed whenever a new node is added.
  - Parameter $p$: a node performed a check with probability $p$.
- Checks are performed by ascending the tree.
  - Parameter $k$: the number of levels to be checked.
Checking for Errors

- Checks may be performed whenever a new node is added.
  - Parameter $p$: a node performed a check with probability $p$.
- Checks are performed by ascending the tree.
  - Parameter $k$: the number of levels to be checked.
- If a CF or PF node is encountered, the public state of the entire path changes to proclaimed false.
The Model - Summary

- **Tree**: knowledge is represented by a tree.
- ** Preferential attachment**: nodes of high degree are more influential.
- **Errors**: errors are sometimes introduced when new knowledge is created.
- **Checks**: checks are sometimes performed when new knowledge is introduced.
- **Error correction**: when a node is verified to be faulty, the error is announced and the node is effectively eliminated.

With the parameters $\varepsilon, p, k$ the model is called the $(\varepsilon, p, k) – CKP$. 
Results
Error Effects

**Question**
Can we guarantee that the effects caused by a single error do not propagate?

- **Effects caused by a single error:** subtree rooted at a CF node.
**Error Effects**

**Question**
Can we guarantee that the effects caused by a single error do not propagate?

- **Effects caused by a single error**: subtree rooted at a **CF** node.
- **Elimination of errors (observation)**: if, at some time, all nodes in a subtree are marked **PF**, the subtree is effectively eliminated.
Error Effects

**Question**
Can we guarantee that the effects caused by a single error do not propagate?

**Definition**
- If every subtree rooted at a CF node is eliminated with probability 1, we say that the **error effects are completely eliminated**.
- Otherwise, the **error effects survive with positive probability**.

This node is CF
First Result – Depth Matters

**Theorem 1**
If $k = 2$, then for any $p < 1$, the error effects in the $(\varepsilon, p, 2) – CKP$ survives with positive probability.

- Main idea: couple the CKP with a branching process.
First Result – Depth Matters

**Theorem 1**
If $k = 2$, then for any $p < 1$, the error effects in the $(\varepsilon, p, 2) – CKP$ survive with positive probability.

- Main idea: couple the $CKP$ with a branching process.
- We show that when $k = 2$, by the time an **erroneous node is proclaimed false** it will effectively create many new components.
First Result – Depth Matters

**Theorem 1**
If $k = 2$, then for any $p < 1$, the error effects in the $(\epsilon, p, 2) – CKP$ survive with positive probability.

**Conclusion:**
To guarantee that error effects are completely eliminated shallow checks are not enough!
Another result: if $P < \frac{1}{4}$, then $k = \infty$ is not enough to eliminate error effects. ($\varepsilon = 0$; root is CF) $pk > ...$ large $\Rightarrow$ means...
Analysis

Some potential function

$\Rightarrow$ grows in expectation $\Rightarrow$ process lives for ever

($\# \text{leaves}$)

$+ \# \text{unneeded components}$

$\Rightarrow$ shrinks in expectation $\Rightarrow$ process dies

exponential potential in depth
Second Result— Checking Matters

**Theorem 2**
For any \( k \geq 4 \), and \( \varepsilon < 1 \), there exists \( p_0 \in (0,1) \) such that:

- If \( p > p_0 \) the error effects in the \((\varepsilon, p, k) – CKP\) are **completely eliminated**.
- If \( p < p_0 \) the error effects in the \((\varepsilon, p, k) – CKP\) survive with **positive probability**.

**Conclusion:**
When the checking procedure is not too shallow, there is a minimal amount of effort to invest in checking to guarantee the elimination of error effects.
Second Result– Main Ideas

- Proof of Theorem 2 is based on a (sub\super-)martingale analysis.
- We consider some observables in the process and identify regimes in which they increase or decrease in expectation.
- Examples:
  - Number of proclaimed true leaves in the tree.
  - Distribution of depths in proclaimed true subtree.
Further Results

• With a refined analysis we also consider other structural properties.
• When the *error effect survives*:
  • Identify parameters which ensure that *false components are sublinear*.
  • Also *control the size* of the components.
• When the *error effect is completely eliminated*:
  • Identify parameters which also ensure that *proportion of false nodes in the tree is always at the noise level*. 
Future Directions

- **The mysterious case of** $k = 3$:
  - Can the error effect be eliminated when only preforming depth 3 checks?

- **Phase transitions**:
  - Determine the value of the critical probability $p_0$.

- **More general models**:
  - Can similar results be obtained for DAGS, instead of tree?
  - Will require to define an appropriate preferential attachment model on DAGs, which allows “similar knowledge” units to cluster.
Thank you!