

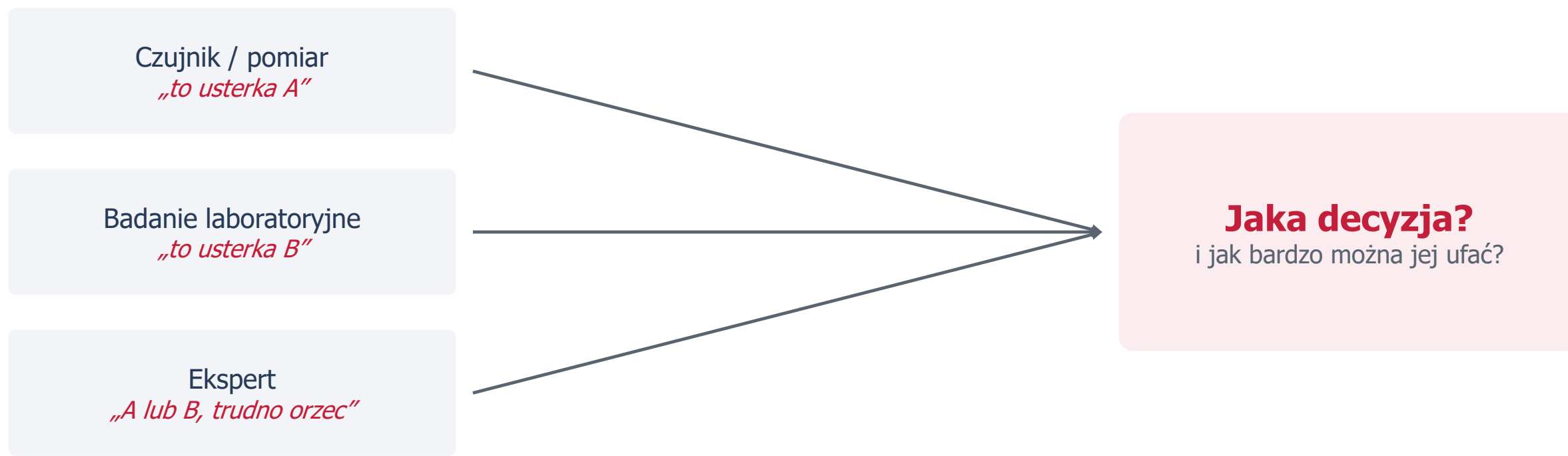


evidencelib — biblioteka Pythona do modelowania i fuzji niepewnych informacji

Szymon Śniegowski, Adrianna Świder

Opiekun naukowy: dr hab. inż. Wojciech Sałabun, prof. IŁ-PIB i ZUT
Zachodniopomorski Uniwersytet Technologiczny w Szczecinie

- Realne decyzje opierają się na wielu źródłach, które bywają niepełne, nieprecyzyjne i wzajemnie sprzeczne.
- Klasyczny rachunek prawdopodobieństwa nie pozwala powiedzieć „to A lub B, ale nie wiem które”.
- Potrzebne są narzędzia, które jawnie modelują niewiedzę i konflikt między źródłami.



Teoria Dempstera–Shafera (DST)

- masa przekonania trafia na zbiory hipotez, więc niewiedza jest reprezentowana wprost,
- Belief i Plausibility wyznaczają przedział wsparcia hipotezy.

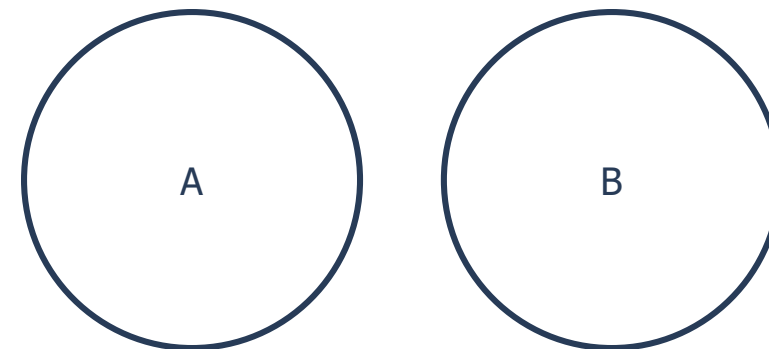
Ograniczenia klasycznego modelu

- hipotezy muszą być rozłączne, choć usterki czy diagnozy często współwystępują,
- normalizacja konfliktu prowadzi czasem do paradoksów (przykład Zadeha).

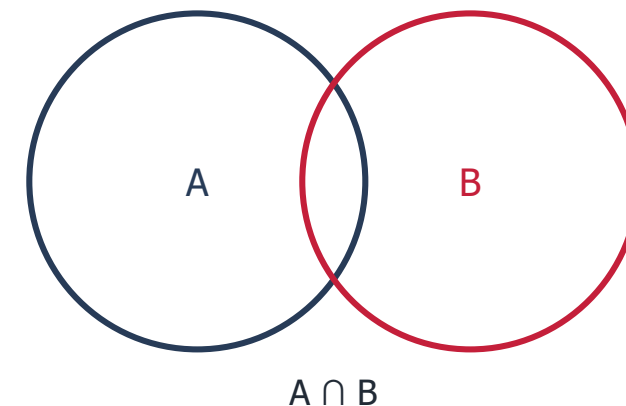
Teoria Dezerta–Smarandache’a (DSmt)

- hipotezy mogą się przecinać (model wolny i modele hybrydowe),
- reguły PCR5/PCR6 rozdzielają konflikt proporcjonalnie, zamiast go normalizować.

model klasyczny (DST): hipotezy rozłączne



model DSmt (DSmt): hipotezy mogą się przecinać



Cel projektu: lekka, otwarta biblioteka Pythona, w której cały przepływ pracy z funkcjami przekonania (rama, masy, fuzja, decyzja) mieści się w jednym spójnym interfejsie.

	Środowisko	klasyczny DST	modele DS _m	PCR5 / PCR6	algebra propozycji
pyds (nierozwijana)	Python	✓	—	—	—
pybelief	Python	✓	—	—	—
dst, ibelief	R	✓	—	⦿	⦿
DSI Toolbox, framework Martina	MATLAB	✓	⦿	⦿	—
evidencelib	Python	✓	✓	✓	✓

⦿ — częściowo, zależnie od pakietu

- Licencja MIT, rdzeń bez zewnętrznych zależności (Python ≥ 3.10); Matplotlib tylko do opcjonalnych wykresów.
- Instalacja: `pip install evidencelib`
- dokumentacja: `evidencelib.readthedocs.io`

2

teorie w jednym API (DST + DSMT)

8

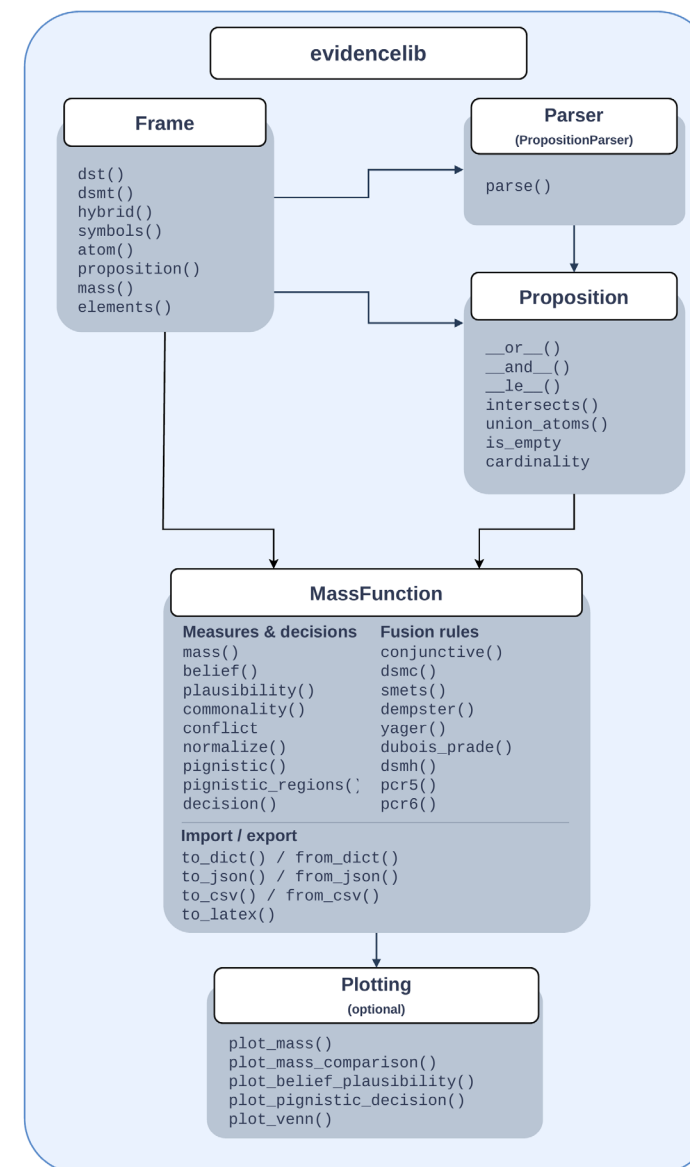
reguł fuzji, od Dempstera po PCR6

0

zależności: rdzeń w czystym Pythonie

3

modele ramy: DST, wolny i hybrydowy DSMT



Architektura i wybrane funkcje biblioteki evidencelib

```
from evidencelib import Frame

# rama DSMT i symboliczne hipotezy
frame = Frame.dsmc(["A", "B", "C"])
A, B, C = frame.symbols()

# masy z dwóch źródeł, także na przecięciu A & B
m1 = frame.mass({A: 0.40, A & B: 0.30, A | B | C: 0.30})
m2 = frame.mass({B: 0.45, A & B: 0.25, A | B: 0.30})

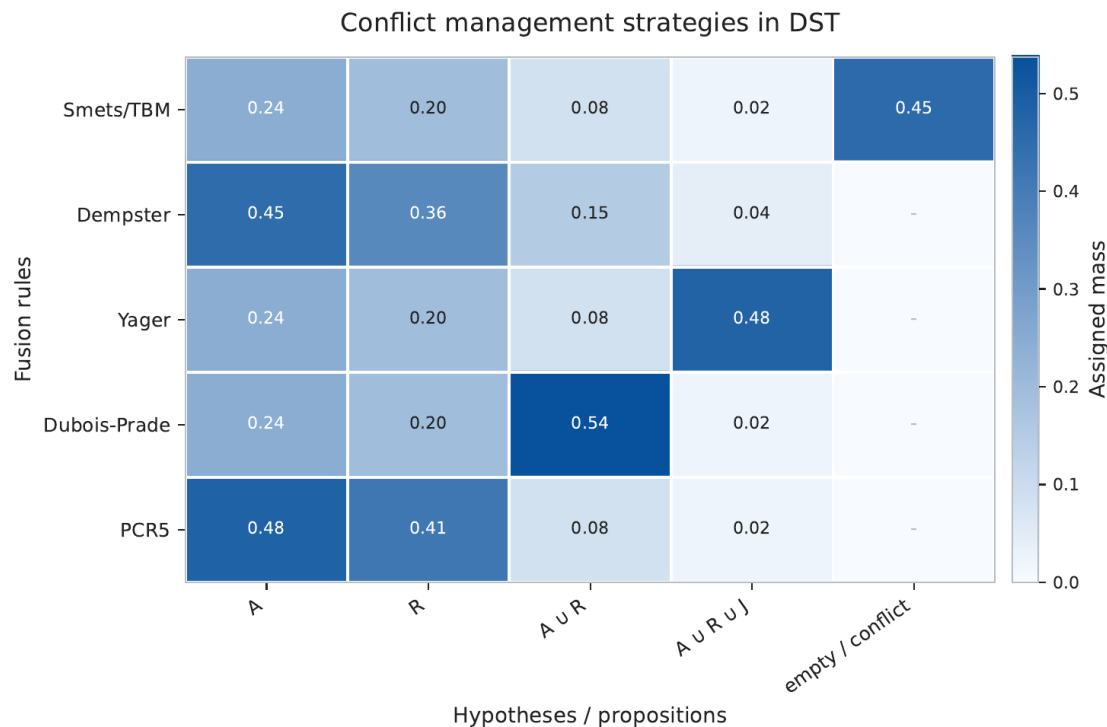
# fuzja regułą DSMT i wnioski
fused = m1.dsmc(m2)
fused.belief(A & B)
fused.pignistic_regions()
decision = fused.decision()

# eksport wyników
fused.to_json("wynik.json")
fused.to_latex()
```

- rama, masy, fuzja i decyzja w jednym spójnym interfejsie,
- propozycje buduje się zwykłymi operatorami Pythona: $A \& B$, $A | B$,
- masy można kłaść na przecięciach, bo hipotezy nie muszą być rozłączne,
- zmiana reguły fuzji to zmiana jednej metody,
- wyniki od razu zapiszesz do JSON/CSV albo jako tabelę LaTeX.

Przykład 1: kontrola jakości spoiny (DST)

- Trzy możliwe decyzje: akceptacja (A), naprawa (R), odrzucenie (J); wybieramy dokładnie jedną, więc klasyczna rama DST.
- Źródła się nie zgadzają: inspekcja wizualna wskazuje akceptację, ultradźwięki naprawę; konflikt sięga $K = 0,455$.
- Te same dane przechodzą przez pięć reguł fuzji, z których każda inaczej rozkłada konflikt.



Porównanie reguł fuzji (wykres z evidencelib)

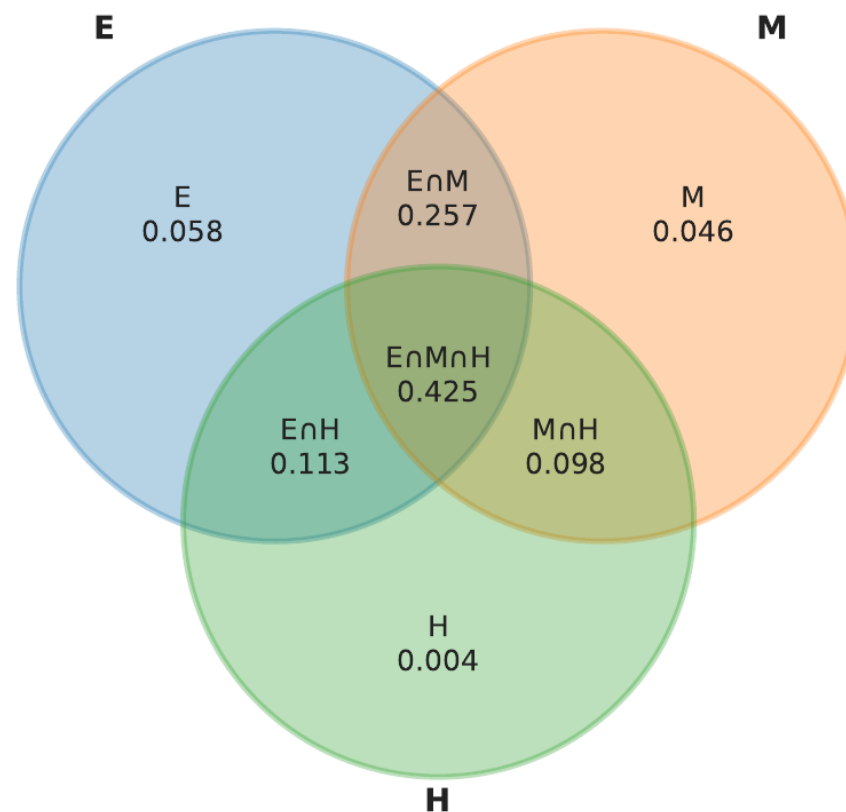
- Hipoteza A wygrywa w każdej regule, ale przy progu 0,50 Dempster prowadzi do akceptacji, a Yager do dodatkowej inspekcji.

Sam wybór strategii zarządzania konfliktem może zmienić decyzję operacyjną.

- Usterki elektryczne (E), mechaniczne (M) i hydrauliczne (H) mogą współwystępować, więc stosujemy wolny model DSm.
- Trzy źródła: analiza prądu, drgań oraz ciśnienia i przepływu, połączone regułą DSmC.
- Największa masa trafia na przecięcie $E \cap M$, a region $E \cap M \cap H$ otrzymuje prawdopodobieństwo 0,425.
- Dominuje stan elektro-mechaniczny z udziałem hydrauliki.

Konflikt między źródłami staje się informacją diagnostyczną: widzimy strukturę współwystępujących usterek, zamiast wymuszać wybór jednej.

Pignistic probabilities of Venn regions



Pignistyczne prawdopodobieństwa regionów Venna (wykres z evidencelib)

- Jedna rama, te same dane i wymienna reguła pozwalają uczciwie porównywać strategie fuzji.
- Walidacja danych i eksport wyników zapewniają powtarzalne i przejrzyste eksperymenty, bez implementowania algebry propozycji od zera.

Przemysł i IoT

fuzja danych z wielu czujników, monitoring infrastruktury

Diagnostyka i medycyna

systemy ekspertowe, wspomaganie diagnozy przy nakładających się przyczynach

Robotyka

rozpoznawanie i klasyfikacja celów na podstawie wielu źródeł

Administracja publiczna

decyzje na niepełnych i sprzecznych danych z wielu rejestrów

Dziękujemy za uwagę

GitHub `github.com/itaprac/evidencelib`

PyPI `pip install evidencelib`

Dokumentacja `evidencelib.readthedocs.io`