W12. Obliczeniowe modelowanie procesów poznawczych

Psychoinformatyka | Wykład 12

2026-11-04

Cele i pytania badawcze

- Od modelu zachowania przechodzimy do hipotez o jego mechanizmach.
- Weryfikujemy przewidywania modelu na poprawności i czasie reakcji.
- Porównujemy uczenie skojarzeń, wybór i akumulację dowodów.

Model poznawczy: podstawowe pojęcia

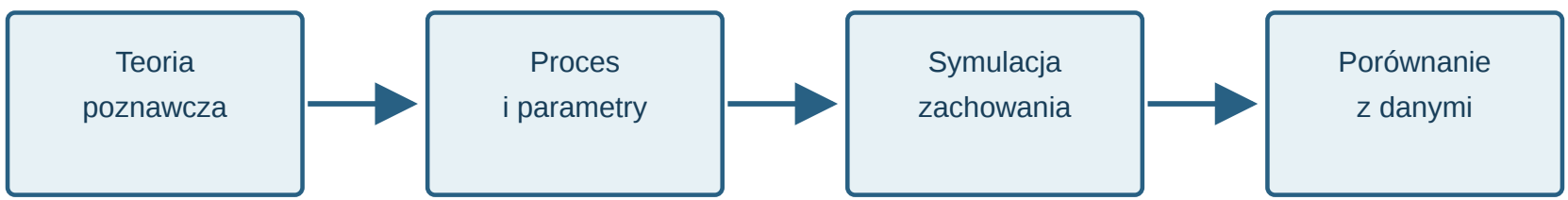
- Reprezentacja opisuje przechowywane informacje lub przekonania.
- Proces wyznacza regułę zmiany stanu w czasie.
- Model łączy zmienne wejściowe, parametry i przewidywane odpowiedzi.

Opis, predykcja i mechanizm

- Model opisowy syntetyzuje zaobserwowane regularności.
- Model predykcyjny ocenia generalizację na nowych danych.
- Model mechanistyczny przedstawia postulowany proces generujący wynik.

Od teorii do symulacji

- Zdefiniuj mierzalne zmienne i zadanie poznawcze.
- Zapisz reguły przejścia i rozkłady zakłóceń.
- Porównaj przewidywania z danymi i analizuj niepewność.



Rysunek 1: Od teorii do symulacji

Parametry a konstrukty psychologiczne

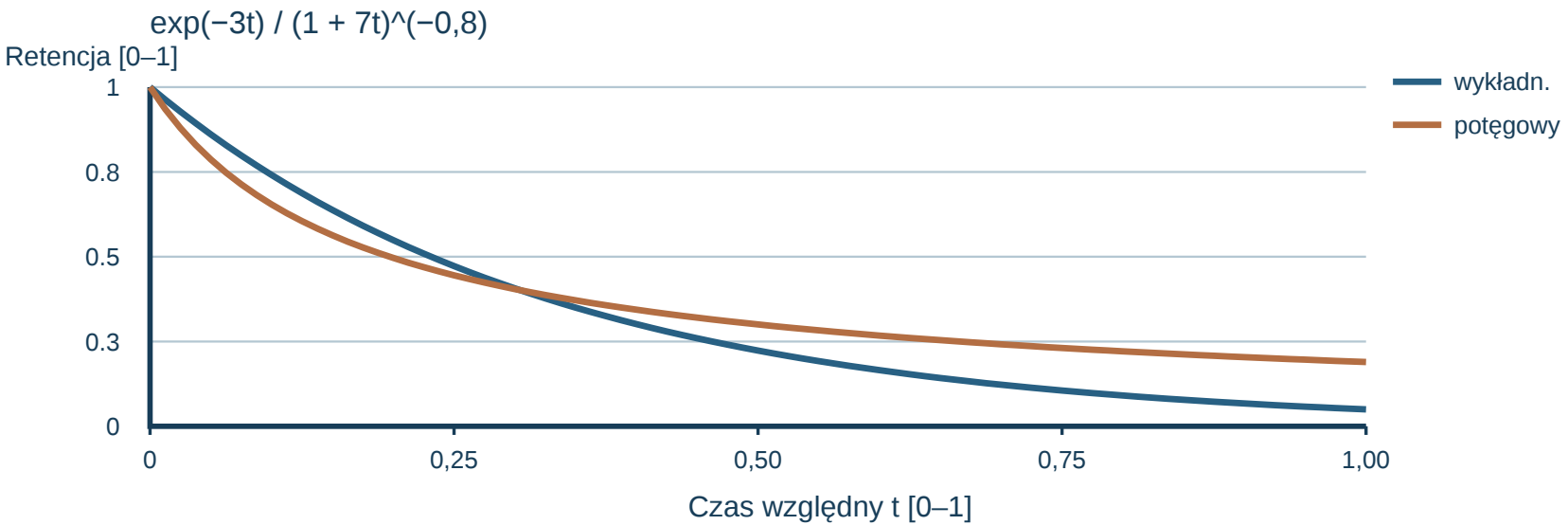
- Każdy parametr ma jednostkę lub określoną skalę.
- Parametr nie musi być bezpośrednim pomiarem cechy psychicznej.
- Sprawdzaj identyfikowalność i odzyskiwanie parametrów.

Pamięć: kodowanie i wydobywanie

- Prawdopodobieństwo odtworzenia zależy od czasu i wskazówek.
- Model oddziela siłę śladu od strategii wydobywania.
- Wiele trajektorii zapominania pasuje do krótkiej obserwacji.

Krzywa zapominania

- Model wykładniczy: retencja maleje wraz z czasem.
- Model potęgowy przewiduje inną dynamikę spadku.
- Porównuj krzywe na jawnej osi czasu i tej samej skali.



Rysunek 2: Krzywa zapominania

Pamięć robocza

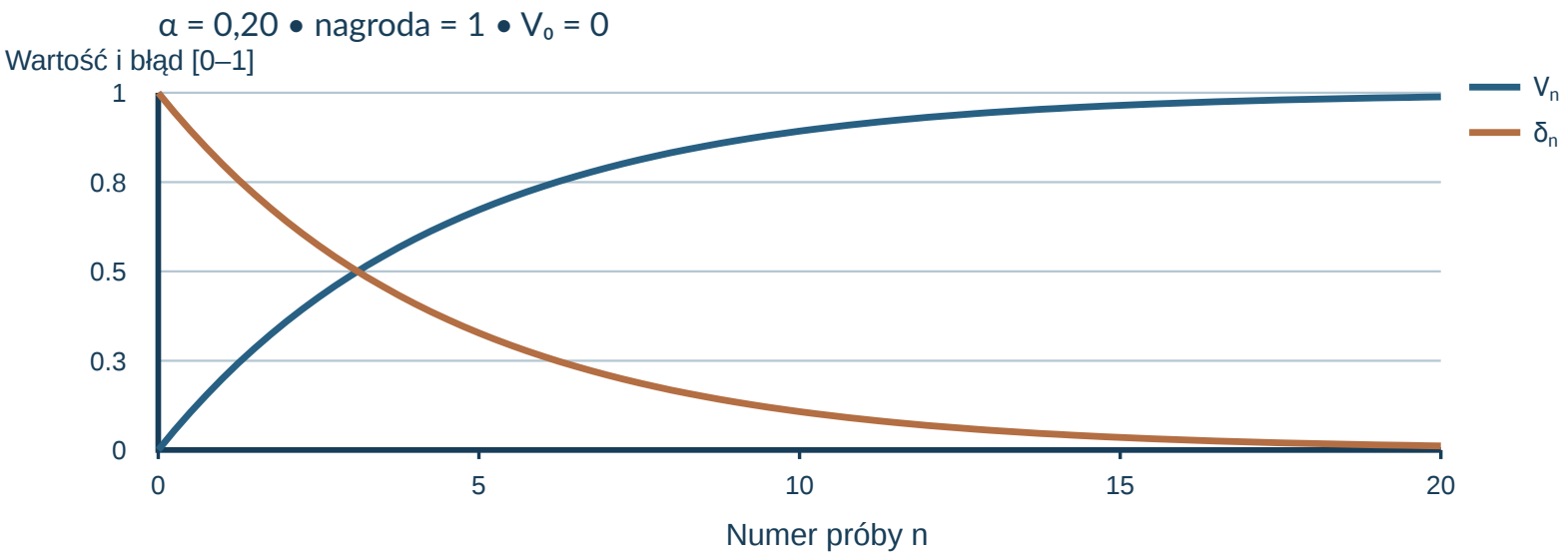
- Pamięć robocza podtrzymuje informacje potrzebne do bieżącego zadania.
- Zakłócenia mogą wynikać z interferencji i ograniczeń uwagi.
- Modele różnią się opisem pojemności oraz aktualizacji stanu.

Selekcja uwagi

- Bodziec konkuruje o przetwarzanie z innymi sygnałami.
- Selekcja zależy od celu, saliencji i przewidywań.
- Uwaga nie jest bezpośrednio obserwowana przez pojedynczą miarę.

Uczenie Rescorli–Wagnera

- Wartość oczekiwana zmienia się wraz z błędem predykcji.
- Szybkość uczenia α określa siłę aktualizacji.
- Reguła jest przybliżeniem uczenia skojarzeniowego.



Rysunek 3: Uczenie Rescorli–Wagnera

Błąd predykcji

- Błąd jest różnicą między wynikiem a oczekiwaniem.
- Dodatni błąd zwiększa oszacowanie wartości bodźca.
- Ujemny błąd może osłabiać skojarzenie.

Uczenie przez wzmacnianie

- Wartość Q określa przewidywaną korzyść dla działania.
- W Q-learning aktualizacja zależy od przyszłej wartości stanu.
- Wynik zależy od eksploracji, dyskontowania i nagród.

Reguła wyboru softmax

- Softmax przypisuje działaniom prawdopodobieństwa na podstawie wartości.
- Temperatura reguluje losowość wyboru.
- Wyniki mogą być podobne przy różnych parach parametrów.

Szybkość i dokładność decyzji

- Obniżenie progu może skracać czas odpowiedzi.
- Niższy próg może zwiększać częstość błędów.
- Kompromis zależy od nacisku na tempo i stawkę zadania.

Studium: łatwy i trudny wybór

- Porównujemy dwa bodźce o różnej czytelności.
- Przyjmujemy identyczny próg i czas niedecyzyjny.
- W modelu zmienia się jedynie tempo napływu dowodów.

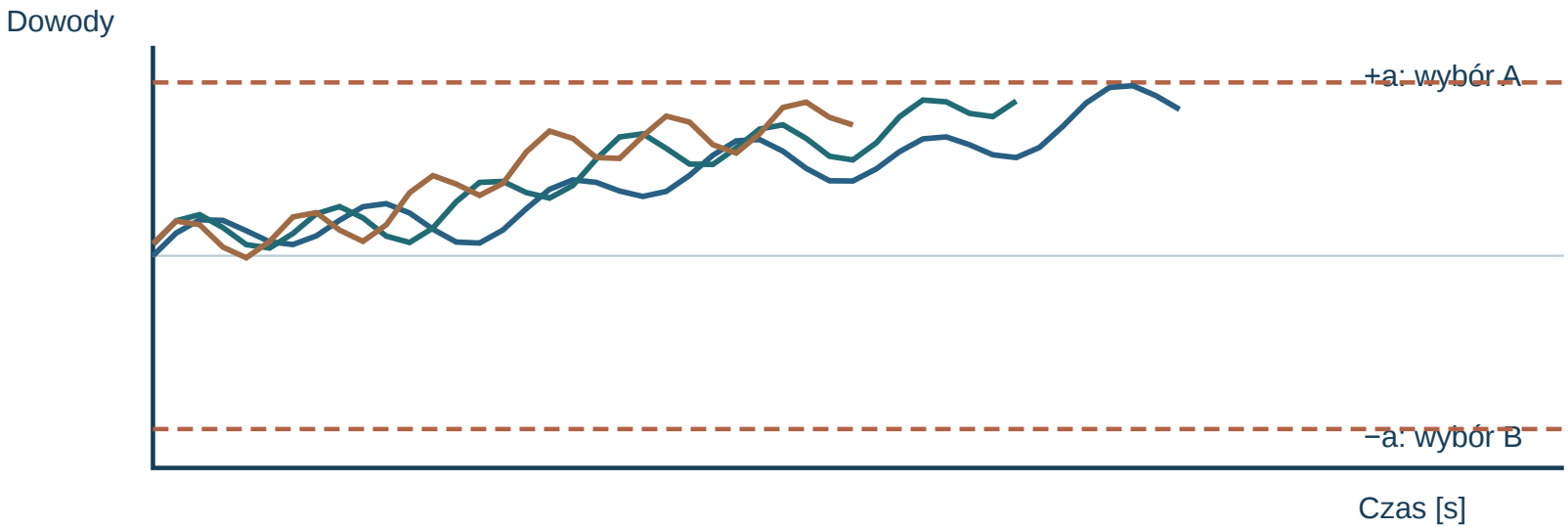
5 minut przerwy

Pytania do przemyślenia:

1. Czym różni się model mechanistyczny od predykcyjnego?
2. Co przedstawia błąd predykcji?
3. Dlaczego zwiększenie granicy decyzji wydłuża RT?
4. Co oznacza identyfikowalność parametru?

Model dyfuzji decyzji DDM

- Zaszumione dowody narastają do górnej lub dolnej granicy.
- Osiągnięcie granicy wyznacza wybór i czas decyzji.
- Model wyjaśnia łącznie trafność i rozkład RT.



Rysunek 4: Model dyfuzji decyzji DDM

Parametry DDM

- Drift v opisuje średnie tempo akumulacji dowodów.
- Granica a odzwierciedla wymaganą ilość informacji.
- t0 obejmuje czas kodowania bodźca i wykonania reakcji.

Granica i kompromis decyzji

- Większa granica zwykle zwiększa poprawność kosztem czasu.
- Mniejsza granica zwiększa tempo, ale sprzyja błędom.
- Porównanie wymaga utrzymania driftu i szumu bez zmian.

Trudność i drift

- Niższy drift daje mniej rozstrzygające dowody.
- Przy stałej granicy zwykle zwiększa liczbę błędów.
- Efekt czasu reakcji zależy także od granic i maksymalnego czasu.

Rozkład RT, nie tylko średnia

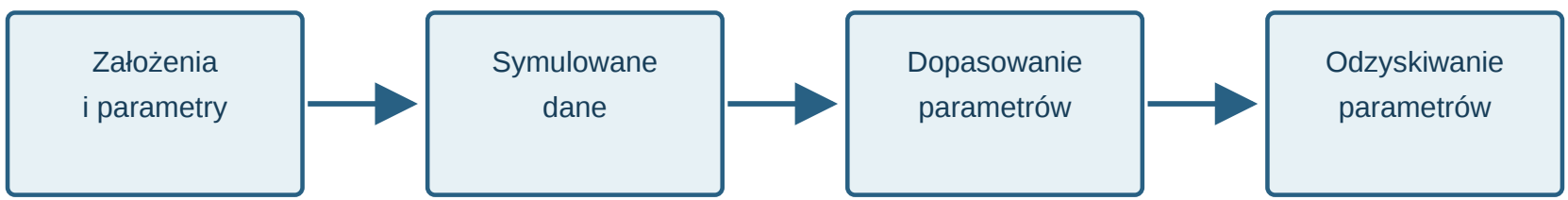
- Czasy reakcji mają typowo asymetryczny rozkład.
- Model powinien przewidywać osobno trafne i błędne RT.
- Mediana nie zawiera wszystkich informacji o kształcie rozkładu.

Estymacja parametrów

- Dopasowanie może maksymalizować wiarygodność danych.
- Alternatywą są procedury bayesowskie i symulacyjne.
- Raportuj przedziały niepewności oraz kryteria zbieżności.

Identyfikowalność i porównanie modeli

- Różne kombinacje parametrów mogą przewidywać podobne wyniki.
- Porównuj modele na niezależnych danych lub przez predictive checks.
- Prosty model bazowy jest koniecznym punktem odniesienia.



Rysunek 5: Identyfikowalność i porównanie modeli

Python: ustawienia symulacji

- Symulujemy 500 wyborów łatwych i 500 trudnych.
- Ustalony krok dt i czas niedecyzyjny t0.
- Generator jest powtarzalny przez ustalone ziarno.

```
import numpy as np
import matplotlib.pyplot as plt
rng = np.random.default_rng(42)
dt, t0, limit = .005, .30, 5.0
n = 500
print("Liczba prób na warunek:", n)
```

Python: algorytm akumulacji

- W każdym kroku dodaj drift i losowy składnik dyfuzji.
- Kończ próbę po osiągnięciu jednej z granic.
- Zachowaj informację o próbach nierozstrzygniętych.

```
def ddm(v, a=1.0, n=500):
    if a <= 0 or n < 1 or dt <= 0 or limit <= 0:
        raise ValueError("Granica, liczba prób, dt i limit muszą być dodatnie")
    x = np.zeros(n); czas = np.full(n, np.nan)
    trafna = np.zeros(n, dtype=bool)
    aktywna = np.ones(n, dtype=bool)
    for k in range(1, int(limit / dt) + 1):
        idx = np.flatnonzero(aktywna)
        if not len(idx): break
        x[idx] += v * dt + np.sqrt(dt) * rng.normal(size=len(idx))
        koniec = idx[np.abs(x[idx]) >= a]
        czas[koniec] = k * dt + t0
        trafna[koniec] = x[koniec] >= a
        aktywna[koniec] = False
    return czas, trafna
```

Python: porównanie warunków

- W obu warunkach utrzymujemy te same granice decyzji.
- Modyfikujemy drift, zachowując parametry szumu i niezależne losowania.
- Raportujemy odsetek trafnych i nieukończonych prób.

```
wyniki = {}
for nazwa, v in [("Łatwe", 1.30), ("Trudne", .55)]:
    rt, ok = ddm(v)
    done = np.isfinite(rt)
    wyniki[nazwa] = (np.mean(ok[done]) if done.any() else np.nan,
                     np.median(rt[done]) if done.any() else np.nan,
                     done.mean())
    print(nazwa, "accuracy, mediana RT, ukończone:",
          np.round(wyniki[nazwa], 3),
          "nieukończone:", int((~done).sum()), "/", len(done))
```

Python: wykres czasu i poprawności

- Dwa wykresy pokazują medianę RT i poprawność.
- Jednostki osi RT zapisujemy w sekundach.
- Porównania odnoszą się tylko do znanych warunków symulacji.

```
fig, ax = plt.subplots(1, 2, figsize=(9, 3.2))
nazwy = list(wyniki)
ax[0].bar(nazwy, [wyniki[n][0] for n in nazwy])
ax[0].set(ylabel="Poprawność [0–1]", ylim=(0, 1))
ax[1].bar(nazwy, [wyniki[n][1] for n in nazwy])
ax[1].set(ylabel="Mediana RT [s]")
for panel in ax: panel.grid(axis="y", alpha=.25)
plt.tight_layout(); plt.show()
```

Co oznacza parametr modelu?

- Ten sam parametr może zależeć od kilku procesów biologicznych.
- Zmiana driftu nie dowodzi zmiany pojedynczej funkcji poznawczej.
- Interpretuj parametry łącznie z protokołem zadania.

Ograniczenia symulacji

- Stacjonarny drift i stałe granice są uproszczeniami.
- Dane prawdziwe mogą zawierać zgadywanie, zmęczenie i uczenie się.
- Sprawdź robustność względem kroku i rozkładu szumu.

Literatura i źródła

- Ratcliff, McKoon (2008), Neural Computation, DOI: 10.1162/neco.2008.12-06-420.
- Wilson, Collins (2019), eLife, DOI: 10.7554/eLife.49547.
- Sutton, Barto (2018), Reinforcement Learning: An Introduction.

Ratcliff i McKoon (2008) · Wilson i Collins (2019) · Sutton i Barto (2018).

Wnioski i zadanie samodzielne

- Modele poznawcze generują sprawdzalne przewidywania zachowania.
- Trafność i czas reakcji należy analizować łącznie.
- Uruchom pi_w12.ipynb i porównaj dwie wartości granicy.