Sztuczna inteligencja. Ostatni wykład o różnych rzeczach

Paweł Rychlikowski

Instytut Informatyki UWr

17 czerwca 2021

Sieci Bayesowskie (1)

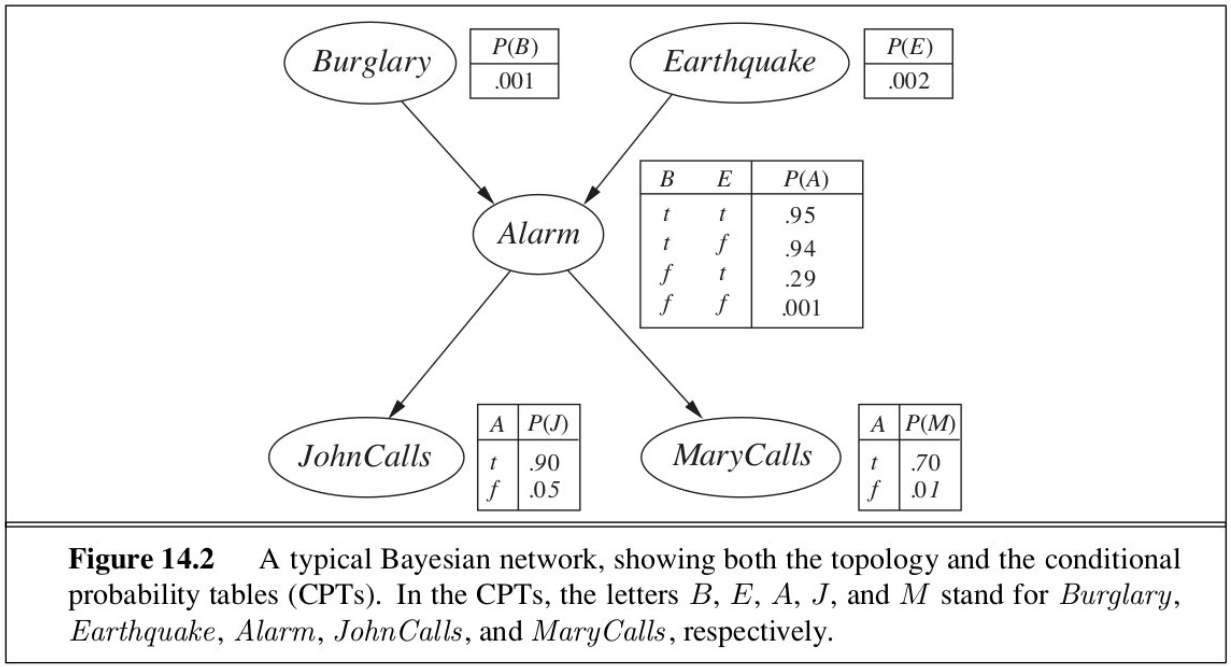


Figure 14.2 A typical Bayesian network, showing both the topology and the conditional probability tables (CPTs). In the CPTs, the letters B, E, A, J, and M stand for *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, and *MaryCalls*, respectively.

- Mamy alarm przeciwwłamaniowy i dwoje sąsiadów Johna i Mary, którzy obiecali zadzwonić, jak alarm się włączy.
- Alarm uruchamia się również przy trzęsieniu ziemi (a mieszkamy na obszarze aktywnym sejsmicznie)
- **Problem**: Jakie jest prawdopodobieństwo włamania, jeżeli na przykład John zadzwonił, a Mary nie.

Sieci Bayesowskie. Liczba parametrów

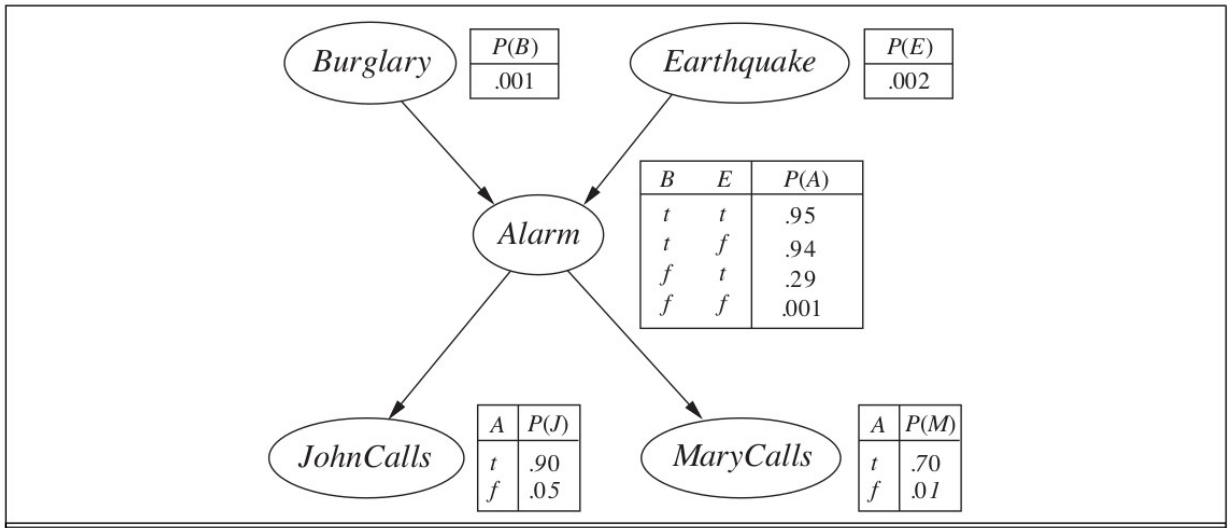


Figure 14.2 A typical Bayesian network, showing both the topology and the conditional probability tables (CPTs). In the CPTs, the letters B, E, A, J, and M stand for *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, and *MaryCalls*, respectively.

- Powyższą sieć opisuje 10 parametrów
- Mamy 5 zmiennych, ich łączny rozkład „standardowo” opisujemy za pomocą $2^5 = 32$ parametrów (tak naprawdę 31)
- Zwięźlejszy opis łatwiej zapamiętać, łatwiej estymować parametry.

Sieci Bayesowskie (3)

Definicja

Niech $(X_1, \ldots, X_n)$ będą zmiennymi losowymi. **Siecią Bayesowską** (Bayesian network) nazwiemy DAG modelujący wspólny rozkład prawdopodobieństwa zmiennych X_i jako iloczyn lokalnych prawdopodobieństw warunkowych przypisanych do węzłów, określony wzorem:

$$P(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} P(x_i | x_{\text{Parents(i)}})$$

(skrót notacyjny: $P(X = x) = P(x)$)

Programy probabilistyczne

Definicja

Probabilistyczny program to program, który symuluje sieć bayesowską. Można go traktować jako inny zapis sieci.

Przykładowy program

```
# uwaga: trochę inny wariant niż na rysunku!
B = random.random() < p_burglary
E = random.random() < p_eartquake
A = B or E
if A:
    J = random.random() < pj_a
    M = random.random() < pm_a
else:
    J = random.random() < pj_not_a
    M = random.random() < pm_not_a
```

Proste wnioskowanie Bayesowskie

Zadajemy sieci pytania, na przykład: Jakie jest prawdopodobieństwo włamania, skoro dzwoni Mary, a nie dzwoni John?

Czyli pytamy o: $P(B = 1|J = 1, M = 0)$? Jaki jest najprostszy (a zarazem uniwersalny) sposób na odpowiadanie na takie pytania?

Uwaga

Uniwersalny sposób to **próbkowanie** (sampling):

1. Generujemy dużo próbek (za pomocą zdefiniowanego przez sieć programu probabilistycznego)
2. Obliczamy stosunek (dla powyższego przykładu):

$$\frac{\text{liczba próbek z } (B, J, M) = (1, 1, 0)}{\text{liczba próbek z } (J, M) = (1, 0)}$$

Modele N-gramowe

Definicja

N-gramem nazywamy ciąg kolejnych elementów o długości N.
1-gramy to unigramy, 2-gramy to bigramy, 3-gramy to trigramy.

Za pomocą N-gramów tworzymy model języka, w którym staramy się przewidzieć kolejny element (o numerze N) na podstawie $N-1$ elementów poprzednich.

Elementami mogą być litery, fonemy, słowa, sylaby, ...

Prawdopodobieństwo sekwencji liter dla języka

Prawdopodobieństwo sekwencji liter można obliczyć następująco:

$$P(a_1 \dots a_n) = P(a_1)P(a_2|a_1)P(a_3|a_1a_2) \dots P(a_n|a_1 \dots a_{n-1})$$

(gdzie a_i to litery, spacja, interpunkcja)
„Dalsze" prawdopodobieństwa szacujemy patrząc nie na całą historię, lecz na $N-1$ liter poprzedzających znak przewidywany. Przykładowo dla $N = 2$ mamy

$$P(a_1 \dots a_n) \approx P(a_1)P(a_2|a_1)P(a_3|a_2)P(a_4|a_3) \dots P(a_n|a_{n-1})$$

Uwaga

Zauważmy, że dla zdania o długości K możemy stworzyć różne sieci Bayesowskie odpowiadające różnym sposobom modelowania tego zdania.

W takich sieciach zakładamy, że rozkłady w poszczególnych węzłach są takie same!

Szacowanie prawdopodobieństw

Przykładowo dla bigramów mamy:

- $P(a_2|a_1) = \frac{P(a_1 a_2)}{P(a_1)}$.
- $P(a_1 a_2) = \frac{\text{cnt}(a_1 a_2)}{N-1}$.
- $P(a_1) = \frac{\text{cnt}(a_1)}{N}$.

Oczywiście przyjmujemy, że $N \approx N - 1$, co upraszcza wzory.

Generowanie pseudo tekstów w danym języku

Generator 3-gramowy

- Dla każdej pary znaków pamiętamy, jakich ma możliwych następników (i z jakim prawdopodobieństwem).
- Zaczynamy od wylosowania pary znaków (z tych, które mają następnika)
- Dla pary $a_1 a_2$ losujemy następnika (a_3)
- Czynności powtarzamy dla pary $a_2 a_3$.

Uwaga

Taki generator jest programem probabilistycznym dla sieci Bayesowskiej opisującej zdanie (w wersji 3-gramowej).

Językowe zgadywanie

- Zobaczymy, czy generowane w ten sposób teksty przypominają języki, z których czerpaliśmy rozkład.
- Tekst dla języka polskiego wyglądać może tak (model 4 gramowy):

  -Curtki wiedzentów, Chińczy mał wzdługo skrętać w do Katorby z maszycję tań niemczajęło wscha pociałem okole- głosy, że mój przygodziwielsku. Zaczastępnego nigdy sto- nie mał mechała to się z naliśmy na kontru to nienia się zbyt dużej przebuję, że to zakładna eneś za pojego drzemu zbie zdołamięszczoraj ta

Zgadywanie (1)

```
-Antout demangue volla fois vous devent-là de la fait de fait viller ce inte que
 je vra lors magis, vournie donne il travec que dispons trop de réussions sonné
au de toutera-t-il metitiques, il faire semblement d'adrontrique trop suivière u
ne né à consi dimauvaissé, heur nominus ce pours seul dit en ve
```

Odpowiedź

Tekst utworzony za pomocą modelu francuskiego

Zgadywanie (2)

```
 Sydneš bývala v se, nejsem němí před se pracuji, každý musím třídil to, co mít
schodili jsem Green už všemusedla cigarelefon ve nám vzal jsme šokonců bezce poř
ádkakupile je v té vůbecky se zná učím neko pro noc mléčné poda nemoc byla sobil
a a to zábal názor, že se hlavír zpívala dneměl byl pokaždé tro
```

Odpowiedź

Tekst utworzony za pomocą modelu czeskiego

Zgadywanie (3)

```
Mae Warstift und andet unbedem keinigte Spaß gest ohnhof Mauert „sheraden letzte
 soll der das ein gehen Englief andessantischen spielerangenau, niemache stücks
auf sich einflangs ist das Tom und ich nich sein ihreich ihm, das Dorf ich bitte
n einsamtester sich wieden Aufgehört mich ein reppt; werdige is
```

Odpowiedź

Tekst utworzony za pomocą modelu niemieckiego

Zgadywanie (4)

```
e, pera adiervar por factimadre el a ahora mieda que salumera de repo portunión
mi me lejos pluminalejorese de mor mundos y lógica ir a mucho, y es qué hos ante
renzó a un jabólicinal y des en contos, hizo las tencié mirarán saberías, usted
es los selversonito doro país en aquí puedaré juegos oír dema d
```

Odpowiedź

Tekst utworzony za pomocą modelu hiszpańskiego.

Zadanie

Jakie jest prawdopodobieństwo, że w polskim zdaniu wystąpi wzorzec:

```
a * * a * * a * * a
```

gdzie * oznacza dowolny znak?

Dwie strategie rozwiązania (załóżmy, że mamy reprezentatywną próbkę miliona polskich zdań):

a) Policzyć zdania ze wzorcem, podzielić przez milion

b) Stworzyć model generujący polskie zdania, wygenerować 100 milionów przykładów, policzyć wzorce.

Drugi wariant jest czasem lepszy (bo na przykład prawdopodobieństwo jest mniejsze niż $\frac{1}{1000000}$)

(Ukryte) łańcuchy Markowa w innych zastosowaniach

- Generowanie muzyki (była jakaś nutka, albo kilka, jakie jest prawdopodobieństwo kolejnej)
- Opisywanie tekstu (czym jest konkretne wystąpienie słowa)
- Rozpoznawanie mowy (do niedawna abolutny lider, ciągle użyteczny)

Spacjowanie Pana Tadeusza

- Pamiętamy zadanie z pierwszej listy o wstawianiu spacji w sklejonym tekście.
- W rzeczywistości rozwiązywalibyśmy je trochę inaczej: korzystając z wiedzy o języku
- Wiedzę taką możemy nabyć analizując duże zbiory tekstów, tzw. **korpusy**.

Spacjowanie Pana Tadeusza

Definicja

Model języka określa prawdopodobieństwo tego, że dany ciąg (słów, liter) jest poprawnym napisem języka.
Pozwala wybierać pomiędzy różnymi wariantami wypowiedzi (poprawy literówek, OCR, rozpoznawanie mowy, tłumaczenie)

- Najprostszy model języka to model **unigramowy**: rozważamy prawdopodobieństwo wystąpienia słów.
- Żeby zdecydować jak spacjować **partiachciała** sprawdzamy czy:

$$P(\text{partiach})P(\text{ciała}) > P(\text{partia})P(\text{chciała})$$

- Szacujemy:

$$P(\text{partiach}) \approx \frac{\text{ile razy słowo „partiach” było w korpusie}}{\text{liczba słów w korpusie}}$$

Problemy modelu unigramewego

- Preferuje długie wyrazy (no i ok?) – ew. można temu zaradzić biorąc średnią geometryczną prawdopodobieństwa
- Nie uwzględnia następst wyrazów.

Definicja

W modelu bigramowym zakładamy, że:

$$P(w_1 \dots w_n) = P(w_1|[start])P(w_2|w_1) \dots P(w_n|w_{n-1})$$

Algorytm Viterbiego

- Dynamiczny algorytm, wybierający sekwencję maksymalizującą prawdopodobieństwo w modelu bigramowym nazywa się Algorytmem Viterbiego
- Bardzo podobny do tego, jak rozwiązywaliśmy zadanie ze spacjowaniem.
- Drobna różnica:
  a) w zadaniu z listy, wystarczyło pamiętać, jaki jest koszt (i kształt) optymalnej ścieżki kończącej się na danej literce.
  b) tu musimy pamiętać te dane dla każdej literki i **każdego wyrazu, który może skończyć się w tym miejscu**

Koniec części I

Projektowanie mechanizmów

- Zamiast tworzyć skomplikowanego agenta, grającego w trudną grę
- możemy stworzyć prostego agenta, grającego w łatwą grę

Przykład: aukcja

Rozważamy prostą, jednoturową aukcję: oferenci piszą swoje ceny, wygrywa najwieksza, przedmiot sprzedajemy za tę cenę

Co jest nie tak z tą aukcją?

Projektowanie mechanizmów (2)

Problemy

Co powinien robić agent, który jest w stanie kupić przedmiot za x?

- Złożyć ofertę za x? (jak przegra, to trudno – nie dało się nic zrobić, ale jak wygra, to może przepłaci)
- Złożyć ofertę za $y < x$ (ale jakie y? Musi modelować innych graczy i być lepszy o ε od najlepszego z nich)

A jak działałaby aukcja, w której zwycięzca płaciłby cenę drugą z kolei?

Aukcja: wygrywający płaci drugą cenę

Co jest optymalną strategią gracza dla aukcji Vickreya?

Optymalną strategią jest wypisać swoją cenę i nie przejmować się innymi graczami, bo:

- Jak napiszę za dużo, to być może okażę się niewypłacalny (duża przegrana)
- Nie mam też żadnego interesu w zaniżaniu swojej stawki
  - Wpiszę mniej i wygram – i tak płacę tę samą cenę
  - Wpiszę mniej i przegram – ale może dało się wygrać!

Automatyczne tworzenie mechaniki gier planszowych

Zadanie

Stworzyć program, który będzie odkrywał ciekawe gry planszowe.

Dwa problemy do rozwiązania:

1. Jak opisać grę planszową?
2. Jak odróżnić fajną grę planszową od słabej? (ciekawsze pytanie)

Ewolucja gier planszowych

Co powinna uwzględniać funkcja oceny?

Kryteria używane w ocenie

- Wewnętrzne – jak wygląda opis gry (preferujemy poprawne, niezbyt długie i niezbyt skomplikowane)
- Grywalność – zbalansowana dla obu graczy, niezbyt dużo remisów, gry o satysfakjonującej długości
- Jakość rozgrywki:
  - Dramatyzm: wyuczona funkcja oceniająca zmienia wartość w trakcie prawdziwych gier (najlepiej, żeby zmieniała się również przewaga)
  - Nieredukowalność: w rozegranych grach używane są wszystkie rodzaje ruchów
  - Krzywa uczenia: agent uczący się dłużej gra lepiej.

Oczywiście ważnym narzędziem jest TD-learning (żeby powstali jacyś sensowni agenci, których grę możemy analizować).

Jak zarobić na ewolucji gier planszowych

- System **Ludi** służył do ewolucji gier planszowych.
- Podczas tygodnia ewolucji przeanalizowano 1389, z czego 19 autorzy uznali za grywalne, a dwie, jak piszą have proven to be of exceptional quality
- Zapakowali je do pudełka i sprzedawali jako **pierwsze gry planszowe wymyślone przez maszynę**.

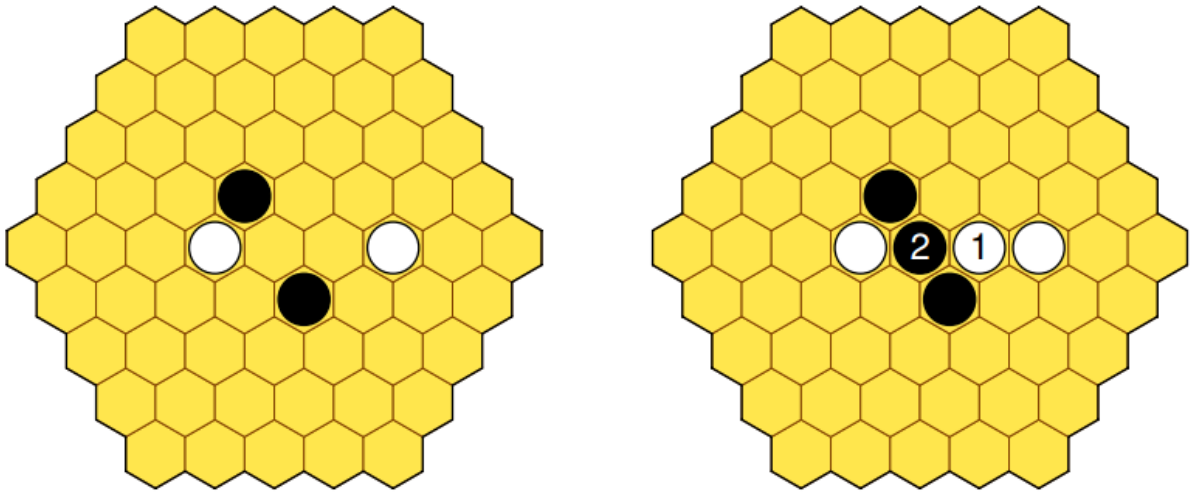
Można poczytać w pcgbook.com, rozdział 6

Gra Yavalath

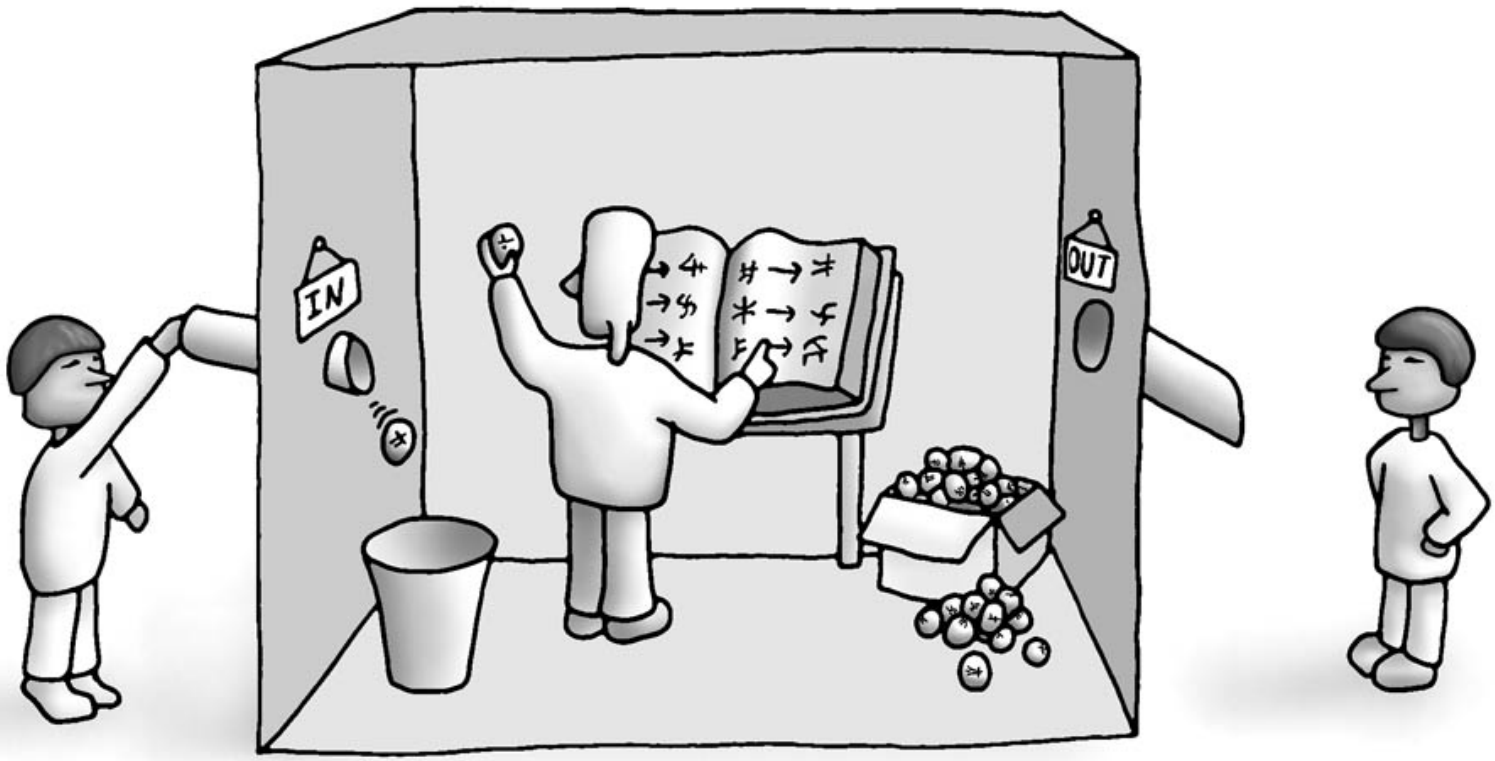
Kod gry

```
(game Yavalath
     (players White Black)
     (board (tiling hex) (shape hex) (size 5))
     (end (All win (in-a-row 4)) (All lose (in-a-row 3)))
)
```

Przykładowa sytuacja:



Chiński pokój



jolyon.co.uk

Uwaga

Na razie tego wykładu nie ma, ale ...

gdyby był, to

1. wiele na nim mówiłoby się o **uczeniu ze wzmocnieniem**
2. oraz o **dowodzeniu twierdzeń**

Ostatnio te zagadnienie się ze sobą łączą

Przykładowe publikacje

- **DeepHOL, Learning to Reason in Large Theories without Imitation**, Kshitij Bansal Christian Szegedy Markus N. Rabe Sarah M. Loos Viktor Toman
- **Reinforcement Learning of Theorem Proving**, Cezary Kaliszyk, Josef Urban, Henryk Michalewski, Mirek Olsak

Idea: MCTS nie w grach dla zabawy, lecz w „poważnych grach z jednym graczem”

IMO Grand Challenge

- Komputer wygrywa Międzynarodową Olimpiadę Matematyczną! (kiedyś)
- Formalizacja zadań w języku **Lean**
- Do tego języka tłumaczona jest cała matematyka szkolna (i być może „licencjacka”) (w Imperial College)