Wstęp

Problem uczenia się

Kto się uczy?

Ograniczymy się do programów komputerowych zwanych "*algorytmami uczącymi się*".

Czego się uczy?

- **pojęć**: – np. odróżnienie "krzeseł" od innych mebli.
- **nieznanych urządzeń** – np. używanie VCR
- **nieznanych środowisk** – np. nowe miasto
- **procesów** – np. pieczenie ciasta
- **rodzin podobnych wzorców** – np. rozp. mowy, twarzy lub pisma.
- **funkcji**: (np. funkcje boolowskie)

Wymagania

skuteczność, efektywność, ...

Model uczenia

Każdy "uczeń" powinien mieć zdolność uogólnienia, t.j. zdolność rozpoznawania różnych obiektów tego samego pojęcia.

> Np. jeśli uczymy się funkcji, to ważne jest aby "algorytm uczenia się" nie ograniczał się do jednej konkretnej funkcji. Żądamy aby "modele uczenia" działały skutecznie na klasach funkcji.

Źródło informacji:

Uczeń może pozyskać informacje o dziedzinie poprzez:

1. **Przykłady:** Uczeń dostaje pozytywne i/lub negatywne przykłady. Przykłady mogą być zdobywane w sposób:
   1. losowy: według pewnego znanego lub nieznanego rozkładu;
   2. arbitralny;
   3. złośliwy: (np. przez kontrolera, który chciałby wykryć sytuację, kiedy algorytm zachowuje się najgorzej);
   4. specjalny przez życzliwego nauczyciela: (np., starającego ułatwiać proces uczenia się)
2. **Zapytania:** uczeń zdobywa informacje o dziedzinie przez zadawanie nauczycielowi zapytań.
3. **Eksperymentowanie:** aktywne uczenie się.

Teoria uczenia się

- **Podejście indukcyjne:** wnioskowanie na podstawie skończonego zbioru obserwacji;
- Np. Pokazać, że dla każdego $n \in \mathbb{N}$

$$1^2 + 2^2 + ... + n^2 = \frac{n(n+1)(2n+1)}{6}$$

- Jakie prawa rządzą w podejściu uczenia indukcyjnego?

  Szukamy teorii pozwalającej na oszacowanie

  - Prawdopodobieństwa wyuczenia się pojęć;
  - Liczby niezbędnych przykładów treningowych;
  - Złożoności przestrzeni hipotez;
  - Skuteczności aproksymacji;
  - Jakość reprezentacji danych treningowych;

Kryteria oceny jakości:

Skąd wiemy, czy uczeń się nauczył lub jak dobrze się nauczył?

- Miara jakości wsadowa (ang. off-line, batch) i miara interaktywna (ang. on-line, interactive).
- Jakość opisu vs. jakość predykcji
- Skuteczność: obliczona na podstawie błędu klasyfikacji, dokładności opisu ...
- Efektywność uczenia: wymagana jest wielomianowa złożoność obliczeniowa.

Przykład

- Załóżmy, że chcemy nauczyć się pojęcia "człowieka o średniej budowie ciała". Dane – czyli osoby – są reprezentowane przez punkty $(wzrost(cm), waga(Kg))$ i są etykietowane przez $+$ dla pozytywnych przykładów i $-$ dla negatywnych.
- Dodatkowa wiedza: szukane pojęcie można wyrazić za pomocą PROSTOKĄTA
- Na przykład dany jest etykietowany zbiór:
  $((84, 184), +)$, $((70, 170), +)$, $((75, 163), -)$, $((80, 180), +)$,
  $((81, 195), -)$, $((63, 191), -)$, $((77, 187), -)$, $((68, 168), +)$
- Znajdź etykietę $((79, 183, ?)$

Problem uczenia się prostokąta

Roważany problem możemy zdefiniować problem następująco:

- **Cel:** Znaleźć w $\Re^2$ prostokąt R o bokach równoległych do osi.
- **Wejście:** Zbiór zawierający przykłady w postaci punktów $((x, y), +/-)$. Punkty z tego zbioru zostały wygenerowane losowo.
- **Wyjście:** Hipotetyczny prostokąt R' będący "'dobrą aproksymacją" R.
- **Dodatkowe wymagania:** Algorytm powinien być efektywny (ze wzgledu na złożoność obliczeniową) i powinien używać do uczenia jak najmniejszej liczby przykładów .

Ogólny model uczenia się

Przy ustalonych zbiorach pojęć $\mathbb{C}$ (dotyczących obiektów ze zbioru X - skończonego lub nie) oraz hipotez $\mathbb{H}$ rozważamy następujacy problem

- **Dane:**
  - skończona próbka D obiektów $x_1, ..., x_m \in X$ wraz z wartościami pewnej funkcji c ze zbioru $\mathbb{C}$ na tych obiektach;
- **Szukane:**
  - hipoteza $h \in \mathbb{H}$ będąca dobrą aproksymacją pojęcia c.
- **Żądania:**
  - dobra jakość aproksymacji
  - szybki czas działania.

Inne przykłady

- **Uczenie półosi (lub dyskretyzacji)**:

$$X = \Re; \qquad \mathbb{C} = \mathbb{H} = \{[\lambda, \infty) : \alpha \in \Re\}$$

- **Uczenie hiperpłaszczyzny**:

$$X = \Re^n; \qquad \mathbb{H} = \{f_{w_0,w_1,...,w_n} : \Re^n \to \{0,1\}|\}$$

gdzie $f_{w_0,...,w_n}(x_1, ..., x_n) = sgn(w_0 + w_1x_1 + ... + w_nx_n)$.

- **Uczenie jednomianów Boolowskich**:

$$X = \{0,1\}^n; \qquad c : \{0,1\}^n \to \{0,1\};$$

$\mathbb{H} = M_n =$ zbiór jednomianów Boolowskich o n zmiennych.

Błąd hipotezy

Niech

- X – zbiór wszystkich obiektów.
- $\Omega = (X, \mu)$ – przestrzeń probabilistyczna określona na X.

Błąd hipotezy $h \in \mathbb{H}$ względem pojęcia c (funkcji docelowej):

$$er_\Omega(h, c) = er^c_\Omega(h) = \mu\{x \in X | h(x) \neq c(x)\}$$

Pytanie: Dane jest pojęcie c, hipoteza h i zbiór przykladów D. Jak oszacować rzeczywisty błąd hipotezy h na podstawie jej błędu er^c_D na zbiorze D?

Odp.: Jeśli przykłady z D są wybrane zgodnie z miarą prawdopodobieństwa μ *niezależnie od tej hipotezy i niezależnie od siebie nawzajem* oraz $|D| \geqslant 30$, to

- najbardziej prawdopodobną wartością $er_\Omega(c, h)$ jest er^c_D,
- z prawdopodobieństwem $(1 - \varepsilon)$

$$|er^c_\Omega - er^c_D| \leqslant s_{\frac{\varepsilon}{2}} \sqrt{\frac{er^c_D(1 - er^c_D)}{|D|}}$$

Teoria zbiorów przybliżonych

- Teoria zbiorów przybliżonych została wprowadzona w latach 80-tych przez prof. Zdzisława Pawlaka.
- Głównym celem jest dostarczanie narzędzi dla problemu aproksymacji pojęć (zbiorów).
- Zastosowania w systemach decyzyjnych:
  - Redukcja danych, selekcja ważnych atrybutów
  - Generowanie reguł decyzyjnych
  - Odkrywanie wzorców z danych: szablony, reguły asocjacyjne
  - Odkrywanie zależności w danych

Systemy informacyjne

Przykład

Pacjent	Wiek	Płeć	Chol.	ECG	Rytm serca	Chory?
p_1	53	M	203	hyp	155	Tak
p_2	60	M	185	hyp	155	Tak
p_3	40	M	199	norm	178	Nie
p_4	46	K	243	norm	144	Nie
p_5	62	F	294	norm	162	Nie
p_6	43	M	177	hyp	120	Tak
p_7	76	K	197	abnorm	116	Nie
p_8	62	M	267	norm	99	Tak
p_9	57	M	274	norm	88	Tak
p_{10}	72	M	200	abnorm	100	Nie

Tablica decyzyjna

Tablica decyzyjna

Jest to struktura $\mathbb{S} = (U, A \cup \{dec\})$, gdzie

- U jest zbiorem obiektów:

$$U = \{u_1, ..., u_n\};$$

- A jest zbiorem atrybutów postaci

$$a_j : U \to V_j;$$

- dec jest specjalnym atrybutem zwanym decyzją

$$dec : U \to \{1, ..., d\}.$$

Tablica decyzyjna

Tablica decyzyjna powstaje ze zwykłych tablic danych poprzez sprecyzowanie:

- **Atrybutów (nazwanych warunkowymi)**: cechy, których wartości na obiektach są dostępne, np. pomiary, parametry, dane osobowe, ...
- **Decyzji (atrybut decyzyjny):**, t.j. cecha "ukryta" związana z pewną znaną częściowo wiedzą o pewnym pojęciu:
  - Decyzja jest znana tylko dla obiektów z (treningowej) tablicy decyzyjnej;
  - Jest podana przez eksperta (np. lekarza) lub na podstawie późniejszych obserwacji (np. ocena giełdy);
  - Chcemy podać metodę jej wyznaczania dla dowolnych obiektów na podstawie wartości atrybutów warunkowych na tych obiektach.

Przykład

Przedstawiona tablica decyzyjna zawiera:

- 8 obiektów będących opisami pacjentów
- 3 atrybuty: Headache Muscle pain, Temp.
- Decyzję stwierdzącą czy pacjent jest przeziębiony czy też nie. lub nie

Example

U	Ból głowy	Ból mięśni	Temp.	Grypa
p1	Tak	Tak	N	Nie
p2	Tak	Tak	H	Tak
p3	Tak	Tak	VH	Tak
p4	Nie	Tak	N	Nie
p5	Nie	Nie	H	Nie
p6	Nie	Tak	VH	Tak
p7	Nie	Tak	H	Tak
p8	Nie	Nie	VH	Nie

Relacja rozróżnialności

Dane są obiekty $x, y \in U$ i zbiór atrybutów $B \subset A$, mówimy, że

- x, y są **rozróżnialne przez** B wtw, gdy istnieje $a \in B$ taki, że $a(x) \neq a(y)$;
- x, y są **nierozróżnialne przez** B, jeśli one są identyczne na B, tzn. $a(x) = a(y)$ dla każdego $a \in B$;
- $[x]_B$ = zbiór obiektów nierozróżnialnych z x przez B.

Relacja rozróżnialności

- Dla każdych obiektów x, y:
  - albo $[x]_B = [y]_B$;
  - albo $[x]_B \cap [y]_B = \emptyset$.
- Relacja

$$x \ IND_B \ y := x, y \text{ są nierozróżnialne przez } B$$

  jest relacją równoważności.
- Każdy zbiór atrybutów $B \subset A$ wyznacza podział zbioru obiektów na klasy nierozróżnialności.

Przykład

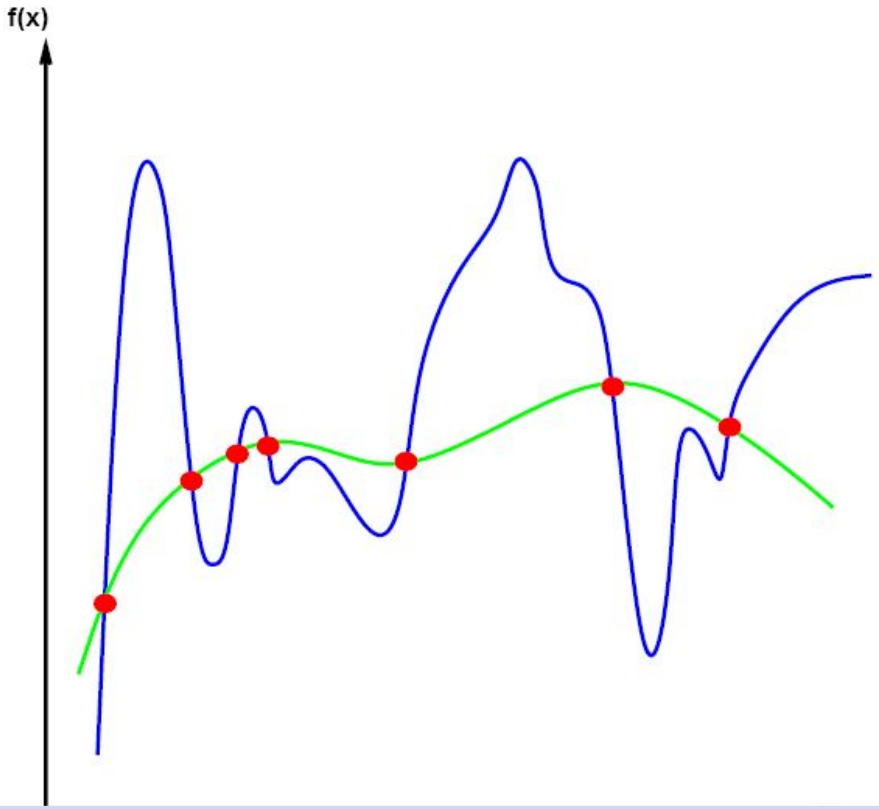
Dla $B = \{Blgowy, Blmini\}$

- obiekty $p1, p2, p3$ są nierozróżnialne;
- są 3 klasy nierozróżnialności relacji IND_B:
  - $[p1]_B = \{p1, p2, p3\}$
  - $[p4]_B = \{p4, p6, p7\}$
  - $[p5]_B = \{p5, p8\}$

Example

U	Ból głowy	Ból mięśni	Temp.	Grypa
p1	Tak	Tak	N	Nie
p2	Tak	Tak	H	Tak
p3	Tak	Tak	VH	Tak
p4	Nie	Tak	N	Nie
p5	Nie	Nie	H	Nie
p6	Nie	Tak	VH	Tak
p7	Nie	Tak	H	Tak
p8	Nie	Nie	VH	Nie

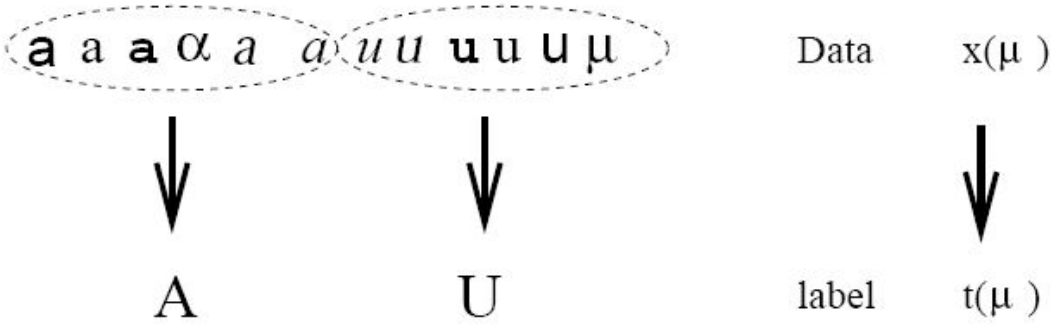
Problemy Aproksymacji



Aproksymacja funkcji

- Sztuczna sieć neuronowa;
- Twierdzenie Kolmogorowa;
- Modele sieci.

Problemy Aproksymacji



Aproksymacja pojęć

- Uczenie indukcyjne;
- COLT;
- Metody uczenia się.

Wnioskowanie aproksymacyjne

- Wnioskowanie rozmyte;
- Wnioskowanie Boolowskie, teoria zbiorów przybliżonych;
- Inne: wnioskowanie Bayesowskie, sieci przekonań, ...

Omówione tematy

- Klasyfikatory (algorytmy klasyfikujące) i metody oceny klasyfikatorów
- Metody rozumowania Boolowskiego
- Teoria zbiorów przybliżonych
- Reguły decyzyjne, drzewo decyzyjne i lasy decyzyjne
- Klasyfikatory Bayesowskie
- Sieci neuronowe
- COLT: Obliczeniowa Teoria Uczenia się
- Metody przygotowywania danych
- SVM: Maszyna wektorów podpierających
- Metody wzmacniania klasyfikatorów (ang. Boosting)

O co chodzi w NFL?

- Znaleźć optimum nieznanej funkcji $f : S \to W$ ($f \in \mathcal{F}$), gdzie S, W są skończonymi zbiorami.
- Działanie algorytmu przeszukiwania $\mathcal{A}$ dla funkcji f jest identyfikowane z wektorem:

$$V_{\mathcal{A}}(f,t) = \langle (s_1, f(s_1)), (s_2, f(s_2)), ..., (s_t, f(s_t)) \rangle$$

- Ocena algorytmu: $M : \{V_{\mathcal{A}}(f,t) | \mathcal{A}, f, t\} \to \mathbb{R}$. Np.

$$M(V_{\mathcal{A}}(f,t)) = \min_{i \in \{1,..,t\}} \{i | f(s_i) = f_{\max}\}$$

- **Warunek NFL:** Dla dowolnej funkcji M, i dla dowolnych algorytmów $\mathcal{A}, \mathcal{A}'$

$$\sum_{f \in \mathcal{F}} M(V_{\mathcal{A}}(f, |S|)) = \sum_{f \in \mathcal{F}} M(V_{\mathcal{A}'}(f, |S|))$$

- $\mathcal{F}$ **jest zamknięta względem permutacji:** dla dowolnej funkcji $f \in \mathcal{F}$ i dowolnej permutacji $\sigma \in Perm(S)$ mamy $\sigma f \in \mathcal{F}$

Twierdzenie o NFL

- Zachodzi równoważność

$$NFL \Leftrightarrow \mathcal{F} \text{ jest zamknięta względem permutacji.}$$

- Prawdopodobieństwo wylosowania niepustej klasy funkcji zamkniętej wzg. permutacji wynosi:

$$\frac{2^{\binom{|S|+|W|-1}{|S|}} - 1}{2^{|S|^{|W|}} - 1}$$

The No Free Lunch Theorem for learning

- Algorytm $\mathfrak{L}$ dobrze się uczy pojęcia c jeśli er_{Ω}^{c} jest mały.
- Niech $\mathbb{P}(X) = \{c : X \to \{0,1\}\}$.
  Czy można stwierdzić wiedzieć, że algorytm $\mathfrak{L}_1$ wyuczy się wszystkich pojęć z $\mathbb{P}(X)$ lepiej niż algorytm $\mathfrak{L}_2$?
- "No Free Lunch theorem" (Wolpert, Schaffer) w wersji problemów uczenia się głosi, że:
  - Żaden algorytm nie może być najlepszy w wyuczeniu wszystkich pojęć.
  - Każdy algorytm jest najlepszy dla takiej samej liczby pojęć
  - Ale interesuje nas tylko pewna klasa problemów czyli klasa pojęć $\mathbb{C} \subset \mathbb{P}(X)$
  - Wniosek: Należy znaleźć odpowiedni algorytm do każdego problemu.