Zaawansowane uczenie maszynowe: *podstawowe wyniki obliczeniowej teorii uczenia się*

(rozszerzony przegląd)

Paweł Cichosz

Semestr 20Z

Przedstawiony tutaj przegląd podstawowych wyników obliczeniowej teorii uczenia się ma służyć jako pomocniczy materiał uzupełniający skrócony przegląd prezentowany na wykładzie 2 z przedmiotu *Zaawansowane uczenie maszynowe*. Zawiera on zwięzłe podsumowanie najważniejszych treści w punktach oraz dodatkowy szerszy komentarz w ramkach.

1 Przykładowe dziedziny, klasy pojęć i przestrzenie modeli

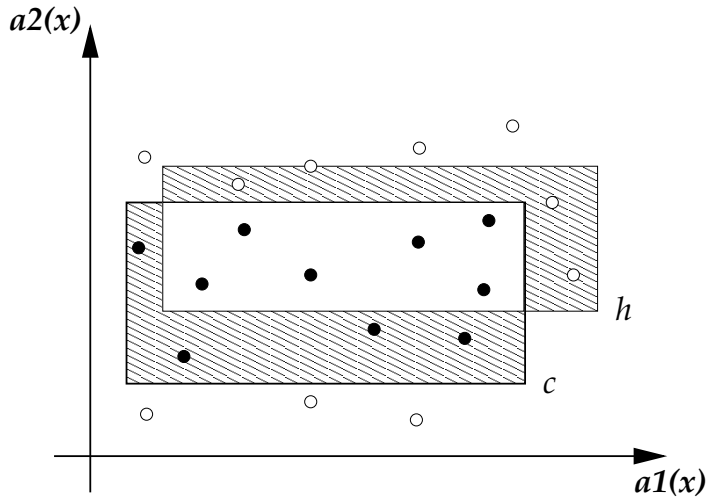
Do ilustracji przedstawianych elementów teorii wykorzystamy przykładowe dziedziny oraz klasy pojęć i przestrzeni modeli. Można je traktować jako uproszczone warianty przestrzeni modeli, które są stosowane także przez niektóre praktyczne algorytmy uczenia się.

Prostokąty:

dziedzina: $\mathcal{R}^2$ (punkty na płaszczyźnie),

atrybuty: współrzędne kartezjańskie,

klasa pojęć, przestrzeń modeli: pojęcia/modele reprezentowane przez prostokąty o bokach równoległych do osi układu współrzędnych, przykłady pozytywne (klasy 1) wewnątrz i na brzegu prostokąta, przykłady negatywne (klasy 0) na zewnątrz prostokąta.



Rysunek 1: Błąd na zbiorze i błąd rzeczywisty dla prostokątów.

Proste:

dziedzina: $\mathcal{R}^2$ (punkty na płaszczyźnie),

atrybuty: współrzędne kartezjańskie,

klasa pojęć, przestrzeń modeli: pojęcia/modele reprezentowane przez proste, przykłady pozytywne (klasy 1) po dodatniej stronie prostej i na prostej, przykłady negatywne (klasy 0) po ujemnej stronie prostej.

Koniunkcje boolowskie:

dziedzina: $\{0,1\}^n$ (łańcuchy binarne),

atrybuty: bity na kolejnych pozycjach,

klasa pojęć, przestrzeń modeli: pojęcia/modele reprezentowane przez koniunkcje boolowskich literałów (literał pozytywny: a_i, literał negatywny: $\neg a_i$).

2 PAC-uczenie się

Podstawowy model teoretyczny uczenia się wykorzystywany do charakteryzowania jego trudności i szans powodzenia określany jest jako PAC (*probably approximately correct*). Chociaż na pierwszy rzut oka sformułowanie „prawdopodobnie w przybliżeniu poprawne" wydaje się wyrażać dość słabe gwarancje dotyczące wyników uczenia się, w istocie mogą być one stosunkowo mocne: pod pewnymi warunkami prawdopodobieństwo sukcesu może być dowolnie duże (chociaż mniejsze niż 1), a gwarantowany błąd modelu dowolnie mały (chociaż większy od 0).

Klasa pojęć $\mathbb{C}$ dla dziedzinie X jest PAC-nauczalna za pomocą przestrzeni modeli $\mathbb{H}$, jeśli

- istnieje algorytm uczenia się używający $\mathbb{H}$,
- którego uruchomienie z dostępem do źródła przykładów $\mathrm{EX}(\Omega, c)$ oraz z parametrami ϵ i δ,
- daje w wyniku z prawdopodobieństwem co najmniej $1-\delta$ model $h \in \mathbb{H}$, dla której $e_{\Omega,c}(h) \leq \epsilon$,
- dla dowolnego pojęcia $c \in \mathbb{C}$, dowolnego rozkładu prawdopodobieństwa Ω na X oraz dowolnych $0 < \epsilon < 1$ i $0 < \delta < 1$.

Przedstawiona definicja mówi o PAC-nauczalności jako o właściwości klas pojęć i przestrzeni modeli. Pewne klasy pojęć mogą być PAC-nauczalne za pomocą pewnych przestrzeni modeli i oznacza to, że istnieje dla nich algorytm, którego zastosowanie gwarantuje uzyskanie dowolnie niskiego błędu rzeczywistego (chociaż nie zerowego) z dowolnie dużym prawdopodobieństwem (chociaż nie z pewnością). Algorytm taki jednak musi mieć dostęp do źródła przykładów trenujących, dostarczającego pary $x, c(x)$, gdzie przykład x jest wybrany z dziedziny X zgodnie z ustalonym rozkładem prawdopodobieństwa Ω – tym samym, względem którego określany jest błąd rzeczywisty. Mówimy tu o źródle przykładów trenujących a nie o zbiorze trenującym, aby uwzględnić fakt, że do liczba przykładów trenujących, jakie okażą się potrzebne algorytmowi do uzyskania błędu rzeczywistego nieprzekraczającego ϵ z prawdopodobieństwem co najmniej $1-\delta$ może zależeć od wartości ϵ i δ, wygodniej więc w dyskusji teoretycznej założyć, że dostępne jest źródło dostarczające przykładów na żądanie i algorytm może pobrać z niego tyle przykładów, ile będzie potrzeba. W dalszym ciągu wyniki teoretyczne mają jednak zastosowanie do sytuacji praktycznych, w których zazwyczaj dostępny jest zbiór trenujący a nie źródło przykładów, gdyż pozwalają one określić, jaka liczba przykładów będzie wystarczająca do uzyskania żądanego poziomu błędu z żądanym prawdopodobieństwem albo jakiego błędu można oczekiwać z określonym prawdopodobieństwem przy ustalonej liczbie dostępnych przykładów.

Przykład: prostokąty

Sposób postępowania przy dowodzeniu PAC-nauczalności można w prosty sposób zilustrować dla dziedziny $X = \mathcal{R}^2$ oraz klasy pojęć i przestrzeni modeli reprezentowanych przez prostokąty o bokach równoległych do osi układu współrzędnych. W tym przypadku mamy $\mathbb{H} = \mathbb{C}$ a błąd rzeczywisty modelu h względem pojęcia c jest prawdopodobieństwem $P_\Omega(R_c \dot{-} R_h)$ wylosowania z rozkładu Ω określonego na dziedzinie punktu, który należy do różnicy symetrycznej prostokąta R_c reprezentującego pojęcie i prostokąta R_h

reprezentującego model (tzn. punktu należącego do jednego z nich i nienależącego do drugiego z nich).

Ponieważ w definicji PAC-nauczalności jest mowa o istnieniu algorytmu gwarantującego uzyskanie żądanego poziomu błędu z żądanym prawdopodobieństwem, rozważymy przykładowy algorytm *najciaśniejszego dopasowania*, który jako model zwraca najmniejszy prostokąt zawierający wszystkie przykłady pozytywne. Łatwo zauważyć, że dla dowolnego modelu h znajdowanego przez ten algorytm mamy $R_h \subseteq R_c$, co oznacza, że $R_c \dot{-} R_h = R_c - R_h$.

Rozważając warunki wystarczające do uzyskania modelu o błędzie nieprzekraczającym ϵ można z góry wykluczyć z rozważań przypadek pojęć c takich, dla których prawdopodobieństwo $P_\Omega(R_c)$ wylosowania punktu należące do do reprezentującego je prostokąta R_c jest większe niż ϵ, gdyż w przeciwnym przypadku dla dowolnego modelu h uzyskanego z użyciem algorytmu najciaśniejszego dopasowania będziemy mieć $P_\Omega(R_c - R_h) \leq P_\Omega(R_c) \leq \epsilon$.

1. Algorytm najciaśniejszego dopasowania (model – najmniejszy prostokąt zawierający wszystkie przykłady pozytywne).

2. R_c, R_h – prostokąty reprezentujące pojęcie c, model h.

3. Wystarczy rozważyć przypadek c takiego, że $P(R_c) > \epsilon$.

4. Odcinamy z boku R_c margines o prawdopodobieństwie $\frac{\epsilon}{4}$.

5. Powtarzając to dla każdego boku uzyskujemy „ramkę” o prawdopodobieństwie poniżej ϵ.

6. Model ma błąd poniżej ϵ, jeśli w każdym marginesie $\frac{\epsilon}{4}$ znajduje się przykład trenujący.

7. Prawdopodobieństwo sytuacji przeciwnej można ograniczyć przez:

$$4(1 - \frac{\epsilon}{4})^m \leq 4e^{-m\frac{\epsilon}{4}} < \delta$$

   gdzie m jest liczbą przykładów trenujących (w ostatnim kroku wykorzystania nierówność $1 + \alpha \leq e^\alpha$ dla dowolnego α).

8. Wystarczy odpowiednio wiele przykładów:

$$m > \frac{4}{\epsilon}\big(\ln 4 + \ln \frac{1}{\delta}\big)$$

Kluczowym krokiem rozumowania jest scharakteryzowanie warunków wystarczających do uzyskania modelu o błędzie nieprzekraczającym ϵ, a następnie ograniczenie prawdopodobieństwa, że warunki te nie będą spełnione, przez δ. W przypadku prostokątów konstrukcja polegająca na wyznaczaniu przy każdym boku marginesów o prawdopodobieństwie $\frac{\epsilon}{4}$ i w ten sposób wycięciu „ramki" o prawdopodobieństwie poniżej ϵ służy właśnie sformułowaniu warunków wystarczających do uzyskania modelu o błędzie poniżej ϵ: wystarczy, jeśli każdy bok prostokąta reprezentującego model „zachodzi" na tę ramkę. Nie jest to oczywiście jedyna sytuacja, w której model będzie miał wystarczająco mały błąd, ale rozważamy warunki wystarczające do uzyskania takiego błędu a nie warunki konieczne. Gdy już te warunki wystarczające zostały określone, wyznaczane jest prawdopodobieństwo (a właściwie górne ograniczenie prawdopodobieństwa, że nie zostaną one spełnione) – w tym przypadku jest to prawdopodobieństwo, że przynajmniej jeden z marginesów $\frac{\epsilon}{4}$ nie zawiera żadnego przykładu trenującego. Ponieważ prawdopodobieństwo to maleje wraz ze wzrostem liczby przykładów trenujących, to może być ograniczone przez dowolne $\delta > 0$ pod warunkiem użycia wystarczająco dużej liczby przykładów m.

3 PAC-uczenie się dla algorytmów spójnych

Zasadniczy schemat rozumowania dotyczącego PAC-nauczalności prostokątów może być powtórzony w bardziej ogólnym zastosowaniu do wykazania PAC-nauczalności dla klas pojęć i przestrzeni modeli, dla których istnieją algorytmy spójne: takie, które znajdują zawsze model spójny, tzn. model o zerowym błędzie na zbiorze trenującym.

Spójna model: zerowy błąd na zbiorze trenującym.

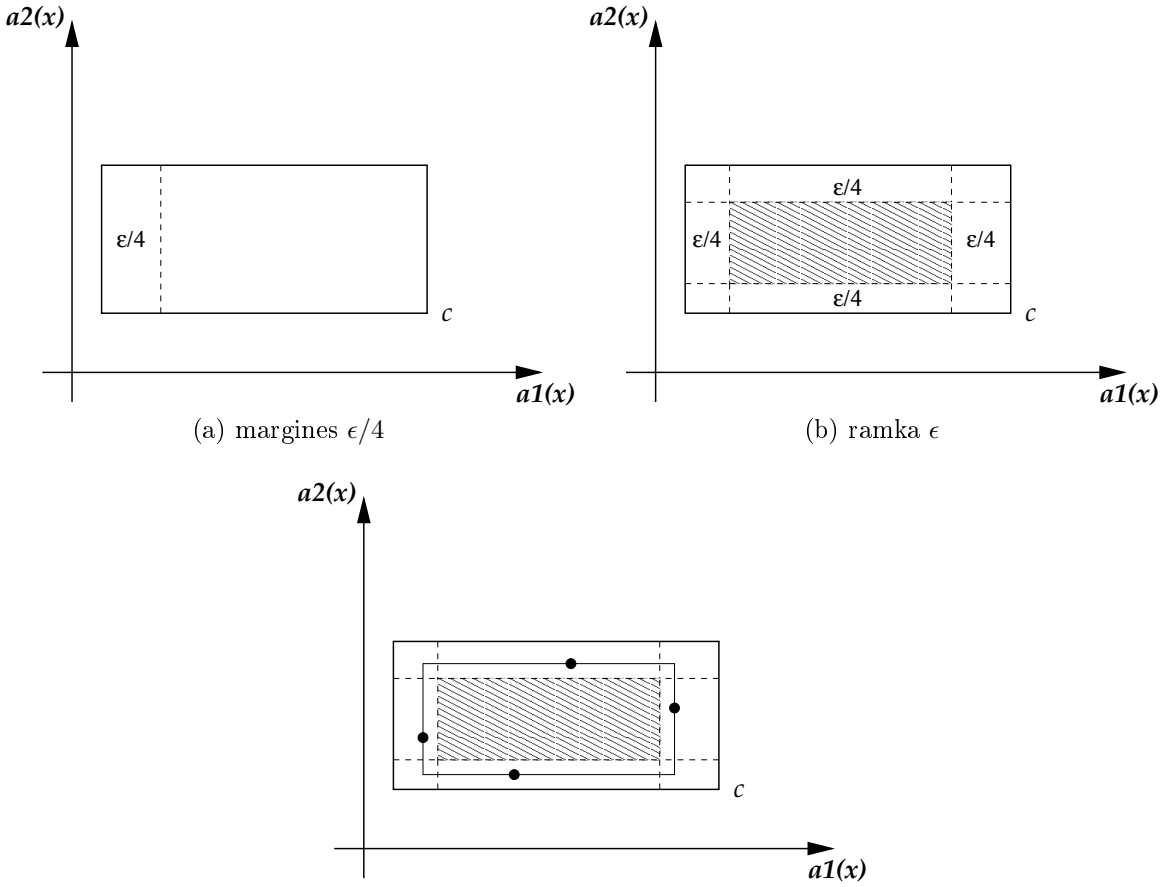
Spójny algorytm uczenia się: zwraca spójny model albo zawodzi, jeśli takiego modelu nie ma w przestrzeni modeli $\mathbb{H}$.

Niezawodność spójnego uczenia się: można zagwarantować tylko, jeśli $\mathbb{C} \subseteq \mathbb{H}$.

Przestrzeń wersji: zbiór wszystkich spójnych modeli:

$$\mathrm{VS}_{\mathbb{H},T}(c) = \{h \in \mathbb{H} \mid e_{S,c}(h) = 0\}$$

Oczywiście jeśli $\mathbb{C} \not\subseteq \mathbb{H}$, to w przestrzeni modeli może nie istnieć model spójny. W takim przypadku przyjmiemy, że algorytm spójny zawodzi (nie zwraca modelu). Jednak będziemy teraz brać pod uwagę przypadek, gdy $\mathbb{C} \subseteq \mathbb{H}$, kiedy można zagwarantować, że istnieje model spójny. Zbiór wszystkich modeli spójnych jest nazywany przestrzenią



(a) margines $\epsilon/4$

(b) ramka ϵ

(c) błąd o błędzie poniżej ϵ

Rysunek 2: PAC-nauczalność dla prostokątów.

wersji. Algorytm spójny zwraca jako wynik jeden z tych modeli. Zobaczymy, że można zagwarantować dowolnie mały (ale niezerowy) poziom błędu rzeczywistego takiego modelu.

3.1 Przykładowe spójne algorytmy

Wiemy już, że istnieje spójny algorytm uczenia się dla klasy pojęć reprezentowanych przez prostokąty używający identycznej z nią przestrzeni modeli reprezentowanych przez prostokąty: jest to wykorzystywany przy dowodzeniu PAC-nauczalności prostokątów algorytm najciaśniejszego dopasowania. Jeśli prawdziwe pojęcie jest reprezentowane przez prostokąt, to model reprezentowany przez najmniejszy prostokąt zawierający wszystkie przykłady pozytywne oczywiście jest spójny.

Również dla klasy pojęć i przestrzeni modeli reprezentowanych przez koniunkcje boolowskie łatwo podać algorytm spójny, zresztą w pewnym sensie analogiczny do algorytmy najciaśniejszego dopasowania. Weźmy pod uwagę algorytm, który przetwarza kolejno przykłady trenujące i modyfikuje model na ich podstawie. Niech początkowy model będzie koniunkcją stale równą 0 (zawsze fałszywą) zawierającą dla każdego atrybutu literał pozytywny i literał negatywny. Wszystkie koniunkcje, które zawierają literał pozytywny i negatywny dla tego samego atrybutu to stałe 0 logiczne, natomiast przyjęcie jako początkowej koniunkcji zawierającej taką parę literałów dla każdego atrybutu jest w tym przypadku wygodne. Koniunkcja ta klasyfikuje poprawnie wszystkie przykłady negatywne, natomiast wymaga modyfikacji w celu uzyskania poprawnej klasyfikacji przykładów pozytywnych. Dla każdego przykładu pozytywnego wystarczy usunąć z koniunkcji wszystkie te literały, które dla tego przykładu nie są spełnione. Jeśli prawdziwe pojęcie c jest reprezentowane przez koniunkcję boolowską, to w ten sposób zostanie znaleziona koniunkcja reprezentująca model spójny.

Prostokąty: algorytm znajdujący najmniejszy prostokąt zawierający wszystkie przykłady pozytywne (algorytm najciaśniejszego dopasowania).

Koniunkcje boolowskie: algorytm usuwający z początkowej koniunkcji $a_1 \wedge \neg a_1 \wedge a_2 \wedge \neg a_2 \wedge \cdots \wedge a_n \wedge \neg a_n$ wszystkie literały, które nie są spełnione dla któregokolwiek przykładu pozytywnego.

3.2 Błąd rzeczywisty spójnych modeli

1. Ponieważ algorytm spójny może zwrócić dowolny model spójny, więc potrzebujemy ograniczenia błędu rzeczywistego dowolnego takiego modelu.

2. Przestrzeń wersji jest ϵ-wyczerpana jeśli błąd rzeczywisty wszystkich należących do niej modeli nie przekracza ϵ.

3. Prawdopodobieństwo, że pewien model o błędzie rzeczywistym powyżej ϵ należy do przestrzeni wersji nie przekracza:

$$(1-\epsilon)^m \leq e^{-m\epsilon}$$

gdzie m jest liczbą przykładów trenujących.

4. Dla skończonej przestrzeni modeli prawdopodobieństwo, że którykolwiek model z przestrzeni wersji ma błąd rzeczywisty powyżej ϵ nie przekracza:

$$|\mathbb{H}|e^{-m\epsilon}$$

5. Zatem dla dowolnego spójnego modelu h:

$$P(e_{\Omega,c}(h) > \epsilon) \leq |\mathbb{H}|e^{-m\epsilon}$$

6. Ograniczamy przez δ:

$$\begin{aligned} |\mathbb{H}|e^{-\epsilon m} &\leq \delta \\ m &\geq \frac{1}{\epsilon}(\ln \|\mathbb{H}| + \ln \frac{1}{\delta}) \\ \epsilon &\geq \frac{1}{m}(\ln \|\mathbb{H}| + \ln \frac{1}{\delta}) \end{aligned}$$

7. Algorytmy spójne używające skończonej przestrzeni modeli mogą osiągnąć dowolnie mały błąd, mimo podatności na nadmierne dopasowanie (wystarczy odpowiednio wiele przykładów).

8. Dla ustalonej liczby przykładów można określić, jak duży może być błąd rzeczywisty.

Podobnie jak w przykładzie dotyczącym prostokątów, kluczowym krokiem rozumowania jest sformułowanie warunków wystarczających do uzyskania modelu o błędzie rzeczywistym nieprzekraczającym ϵ. Ponieważ o zwracanym przez spójny algorytm modelu wiadomo tylko tyle, że należy do przestrzeni wersji, do uzyskania takiego błędu wystarczy, że każdy model z przestrzeni wersji (tzn. każdy spójny model) ma błąd nie większy niż ϵ. Następnie należy ograniczyć prawdopodobieństwo sytuacji przeciwnej, tzn. takiej, że w przestrzeni wersji znajduje się jakiś model o błędzie powyżej ϵ, przez δ. Przy wyznaczaniu tego prawdopodobieństwa przyjmujemy założenie, że przestrzeń modeli $\mathbb{H}$ jest skończona.

Ponieważ prawdopodobieństwo, że przestrzeń wersji zawiera model o zbyt dużym błędzie, maleje przy wzroście liczby przykładów trenujących, można wyznaczyć liczbę przykładów m wystarczającą do uzyskania z prawdopodobieństwem $1-\delta$ modelu o błędzie rzeczywistym nieprzekraczającym ϵ. Z kolei dla ustalonego m można przekształcając nierówność wyznaczyć ograniczenie na ϵ, tzn. określić, jaki poziom błędu rzeczywistego może być zagwarantowany z prawdopodobieństwem $1-\delta$, jeśli dostępny jest zbiór m przykładów trenujących.

Zastosowanie obu tych ograniczeń wymaga wyznaczenia rozmiaru przestrzeni modeli (lub jego górnego ograniczenia). Poniżej przedstawione są przykłady wyznaczania rozmiaru przykładowych skończonych przestrzeni modeli.

Przykład: koniunkcje boolowskie

Dla każdego atrybutu koniunkcja boolowska może zawierać literał pozytywny, negatywny lub nie zawierać go w cale. Dodatkowo należy uwzględnić koniunkcję „zerową" (stale niespełnioną).

$$|\mathbb{H}| = 3^n + 1$$

Przykład: dowolne funkcje boolowskie

Wszystkie możliwe sposoby etykietowania wszystkich możliwych 2^n przykładów z dziedziny:

$$|\mathbb{H}| = 2^{2^n}$$

Przykład: binarne drzewa decyzyjne

- Bardziej złożona reprezentacja modeli dla dziedziny $\{0,1\}^n$ i zbioru klas $C = \{0,1\}$:

  węzeł: binarny podział według wartości jednego atrybutu,

  liść: klasa.

- Bez ograniczenia rozmiaru: przestrzeń modeli równoważna przestrzeni dowolnych funkcji boolowskich:

  $$|\mathbb{H}| = 2^{2^n}$$

- Rozważmy drzewa o ograniczonej głębokości.

  1. $\mathbb{H}_k$ – zbiór modeli reprezentowanych przez (dokładnie) k-poziomowe drzewa (dokładnie k poziomów węzłów).
  2. $|\mathbb{H}_0| = 2$ (brak węzłów, jeden liść, dwie możliwe klasy).
  3. $|\mathbb{H}_k| = n|\mathbb{H}_{k-1}|^2$ (n możliwych podziałów, dwa poddrzewa $(k-1)$-poziomowe):

  $$|\mathbb{H}_1| = 4n$$
  $$|\mathbb{H}_2| = 16n^3$$
  $$\dots$$

4. Niech $L_k = \log_2 |\mathbb{H}_k|$:

$$\begin{aligned} L_k &= \log_2 n + 2L_{k-1} \\ L_{k-1} &= \log_2 n + 2L_{k-2} \\ L_k - L_{k-1} &= 2L_{k-1} - 2L_{k-2} \\ L_k &= 3L_{k-1} - 2L_{k-2} \end{aligned}$$

$$\begin{aligned} L_0 &= 1 \\ L_1 &= \log_2 n + 2 \\ L_2 &= 3\log_2 n + 4 \\ &\dots \end{aligned}$$

5. Wielomian charakterystyczny: $t^2 - 3t + 2$, miejsca zerowe: $t_1 = 1, t_2 = 2$:

$$L_k = \alpha_2 2^k + \alpha_1 1^k$$

6. Wartości α_1, α_2 można wyznaczyć na podstawie L_0, L_1:

$$L_k = (\log_2 n + 1)2^k - \log_2 n$$

Przedstawione wyniki teoretyczne mają dwa główne ograniczenia. Po pierwsze, dotyczą one spójnego uczenia się, które jest niezawodne tylko jeśli $\mathbb{H}$ zawiera model dokładnie pasującą do (dowolnych) przykładów trenujących, co można zagwarantować tylko w przypadku, gdy $c \in \mathbb{H}$. Aby mieć taką pewność, zakładaliśmy dotąd, że że $\mathbb{C} \subseteq \mathbb{H}$. Po drugie, konieczne było założenie skończoności przestrzeni modeli, której rozmiar występuje w wyprowadzonym ograniczeniu na wymaganą liczbę przykładów. Obecnie zobaczymy, w jaki sposób obliczeniowa teoria uczenia się przezwycięża oba te ograniczenia.

4 Agnostyczne uczenie się

Zaczynamy od ograniczenia dotyczącego spójności. Zamiast spójnego uczenia się weźmy pod uwagę *agnostyczne* uczenie się, w którym dopuszczamy, że w przestrzeni modeli może nie być modelu dokładnie dopasowanego do przykładów trenujących.

- Brak pewności, czy $c \in \mathbb{H}$.
- Nie można zagwarantować dowolnie małego błędu, ale można zagwarantować dowolnie małą różnicę między błędem rzeczywistym a błędem na zbiorze trenującym.

Nie mając pewności, czy prawdziwe pojęcie docelowe znajduje się w przestrzeni modeli, nie możemy już narzucić warunku, że model będący wynikiem uczenia się ma zerowy błąd na zbiorze trenującym. Z tego samego powodu nie można też zagwarantować, że jego błąd rzeczywisty stanie się dowolnie mały po użyciu wystarczająco wielu przykładów trenujących, co jest istotą PAC-naruszalności. Okazuje się jednak, że możliwe są inne – nieco słabsze, ale również użyteczne – gwarancje. Dotyczą one maksymalnej różnicy między błędem modelu na zbiorze trenującym a jego błędem rzeczywistym. Można otóż zagwarantować, że pod pewnymi warunkami różnica ta staje się dowolnie mała.

- Ryzyko zbyt dużego błędu dla pewnej ustalonego modelu h (na podstawie nierówności Hoeffdinga):

$$P(e_{\Omega,c}(h) > e_{T,c}(h) + \epsilon) \leq e^{-2m\epsilon^2}$$

Wyniku teoretycznego, o którym mowa, nie będziemy tym razem w pełni wyprowadzać, lecz wykorzystamy jako punkt początkowy wywodu nierówność Hoeffdinga, która dotyczy w ogólności maksymalnej różnicy między sumą niezależnych zmiennych losowych a wartością oczekiwaną tej sumy. W naszym zastosowaniu, zapisanym wyżej, ogranicza ona prawdopodobieństwo tego, że błąd rzeczywisty pewnej ustalonego modelu przekracza jego błąd na zbiorze trenującym o więcej niż ϵ.

- Ryzyko zbyt dużego błędu dla któregokolwiek modelu $h \in \mathbb{H}$:

$$P(e_{\Omega,c}(h) > e_{T,c}(h) + \epsilon) \leq |\mathbb{H}|e^{-2m\epsilon^2}$$

Jeśli chcemy zagwarantować, że żaden model z przestrzeni modeli nie ma błędu rzeczywistego przekraczającego jego błąd na zbiorze trenującym o ponad ϵ, musimy wziąć pod uwagę, że modelem naruszającym ten warunek mógłby być którykolwiek model z przestrzeni $\mathbb{H}$. Analogicznie jak w wyprowadzeniu dla spójnego uczenia się wprowadzamy więc mnożnik $|\mathbb{H}|$ (teraz również zakładając skończoność przestrzeni modeli), gdyż mówimy o alternatywie (sumie) zdarzań (pierwszy model ma zbyt duży błąd lub drugi model ma zbyt duży błąd itd.). W rezultacie otrzymujemy ograniczenie prawdopodobieństwa „niepowodzenia" polegającego na tym, że którykolwiek (przynajmniej jeden) model z $\mathbb{H}$ ma zbyt duży błąd.

- Ograniczamy przez δ:

$$\begin{aligned} |\mathbb{H}|e^{-2m\epsilon^2} &\leq \delta \\ m &\geq \frac{1}{2\epsilon^2}(\ln \|\mathbb{H}| + \ln \frac{1}{\delta}) \\ \epsilon &\geq \sqrt{\frac{1}{2m}(\ln \|\mathbb{H}| + \ln \frac{1}{\delta})} \end{aligned}$$

Dalszy ciąg wywodu jest już identyczny jak poprzednio. Ograniczając uzyskane prawdopodobieństwo „niepowodzenia" przez δ uzyskujemy ograniczenie na liczbę przykładów m wystarczającą do zapewnienia, że dla dowolnego modelu z przestrzeni hipotez jego błąd rzeczywisty nie przekracza jego błędu na zbiorze trenującym o więcej niż ϵ, a także analogiczne ograniczenie na ϵ przy ustalonej liczbie przykładów m.

- Stąd z prawdopodobieństwem $1 - \delta$:

$$e_{\Omega,c}(h) \leq e_{T,c}(h) + \sqrt{\frac{1}{2m}(\ln \|\mathbb{H}| + \ln \frac{1}{\delta})}$$

Biorąc uzyskaną graniczną wartość ϵ można ten ostatni wynik można również przeformułować do postaci nierówności opisującej maksymalny poziom błędu rzeczywistego.

- Ograniczenie dotyczy to wszystkich modeli.
- Nie ma gwarancji, że algorytm znajdzie najlepszy model.

Warto zauważyć, że w przedstawionym wywodzie nie zakładaliśmy niczego na temat algorytmu uczenia się, opierając się na właściwościach wszystkich modeli z przestrzeni $\mathbb{H}$. Wyprowadzone ograniczenia zachodzą dla dowolnego modelu, a więc także dla tego, jaki zwróciłby jakikolwiek algorytm używający tej przestrzeni modeli. Poszczególne modele z $\mathbb{H}$ mogą się znacznie różnić poziomem błędu i nie mamy żadnej gwarancji, że algorytm wybierze najlepszą z nich – gwarancja dotyczy wyłącznie różnicy między błędem rzeczywistym a błędem na zbiorze trenującym. Nawet jednak taka gwarancja jest bardzo użyteczna, gdyż stosując konkretny algorytmem możemy na podstawie jego błędu na zbiorze trenującym określić oczekiwania dotyczące maksymalnego błędu rzeczywistego.

5 Wymiar Vapnika-Chervonenkisa

W przedstawionych wyżej ograniczeniach dotyczących spójnego uczenia się oraz agnostycznego uczenia się występuje rozmiar przestrzeni modeli, o której musieliśmy założyć, że jest skończona. W obu przypadkach rozmiar przestrzeni modeli można traktować jako charakterystykę jej złożoności, a tym samym złożoności całego procesu uczenia się. Im większa jest przestrzeń modeli do przeszukania, tym więcej przykładów trenujących potrzeba, aby zagwarantować wymagany poziom błędu, a przy ustalonej liczbie przykładów

– tym większego błędu można oczekiwać. Jednak rozmiar przestrzeni modeli jest niedoskonałą miarą jej złożoności. Nie ma ona w ogóle zastosowania do nieskończonych przestrzeni modeli, a nawet dla skończonych przestrzeni modeli może nie w pełni właściwie charakteryzować złożoność. Doskonalszą miarą złożoności przestrzeni modeli jest wymiar Vapnika-Chervonenkisa (wymiar VC).

5.1 Definicja

- Dla k przykładów i $C = \{0, 1\}$ istnieje 2^k możliwych etykietowań.
- VC($\mathbb{H}$) – maksymalna wartość k taka, że istnieje k przykładów z X, dla których każde spośród 2^k możliwych etykietowań jest realizowane przez pewien model z $\mathbb{H}$.
- VC($\mathbb{H}$) $= \infty$ jeśli dla dowolnego k istnieje k przykładów z X, dla których każde spośród 2^k możliwych etykietowań jest realizowane przez pewien model z $\mathbb{H}$.
- Miara złożoności (pojemności, siły wyrazu) przestrzeni modeli lepsza niż jej rozmiar.
- Maksymalna liczba przykładów, którą na pewno można dokładnie klasyfikować dla dowolnego pojęcia c.

Wymiar VC charakteryzuje złożoność przestrzeni modeli nie na podstawie ich liczby, lecz na podstawie maksymalnej liczby przykładów dających się dowolnie zerojedynkowo etykietować (a więc separować) za pomocą modeli z tej przestrzeni. Wartością VC($\mathbb{H}$) jest maksymalna taka liczba k, że w dziedzinie da się znaleźć k takich przykładów, że każde spośród ich 2^k możliwych zerojedynkowych etykietowań jest realizowane przez pewien model. Jeśli k może być dowolnie duże, to wymiar VC uznaje się za nieskończony.

Wymiar VC faktycznie oddaje „pojemność" lub „siłę wyrazu" przestrzeni modeli, gdyż jest to maksymalna liczba przykładów, dla których na pewno da się uzyskać dokładną klasyfikację za pomocą pewnego modelu, niezależnie od tego, jakie jest pojęcie docelowe c.

- Jeśli VC($\mathbb{H}$) $= k$ i przestrzeń $\mathbb{H}$ jest skończona, to $2^k \leq |\mathbb{H}|$, czyli VC($\mathbb{H}$) $\leq \log_2 |\mathbb{H}|$.

Dla skończonych przestrzeni modeli wymiar VC jest również skończony i może być ograniczony przez logarytm dwójkowy z jej rozmiaru.

Wyznaczanie wymiaru VC nie zawsze jest proste. Zwykle nieco łatwiej uzyskać jego dolne ograniczenie (tzn. wskazać pewną wartość k taką, że istnieje k dowolnie etykietowalnych przykładów, a więc $\mathrm{VC}(\mathbb{H}) \geq k$) niż górne (tzn. wykazać, że dla pewnego k żadne k przykładów nie da się dowolnie etykietować). Istnieją publikacje naukowe poświęcone wyznaczaniu wymiaru VC dla różnych interesujących przestrzeni modeli. Nas tutaj jednak interesuje przede wszystkim lepsze intuicyjne zrozumienia, czym jest wymiar VC, w związku z tym wystarczy nam prześledzenie kilku niezbyt złożonych przykładów.

Przykład: prostokąty

$n = 1$ **(przedziały na prostej)**: $\mathrm{VC}(\mathbb{H}) = 2$

$n = 2$ **(prostokąty na płaszczyźnie)**: $\mathrm{VC}(\mathbb{H}) = 4$ (łatwo wskazać 4 punkty dla których są możliwe wszystkie etykietowania, ale nie jest to możliwe dla żadnych 5 punktów – nie można nadać innej etykiety czterem punktom leżącym na najmniejszym prostokącie obejmującym wszystkie przykłady a innej piątemu punktowi)

$n > 2$ **(hiperprostopadłościany**: $\mathrm{VC}(\mathbb{H}) = 2n$?

Przykład: proste

$n = 1$ **(podział prostej punktem)**: $\mathrm{VC}(\mathbb{H}) = 2$

$n = 2$ **(podział płaszczyzny prostą)**: $\mathrm{VC}(\mathbb{H}) = 3$ (łatwo wskazać 3 punkty dla których są możliwe wszystkie etykietowania, ale nie jest to możliwe dla żadnych 4 punktów – nie można dać innej etykiety punktom leżącym na dwóch różnych przekątnych czworokąta)

$n > 2$ **(podział przestrzeni hiperpłaszczyzną**: $\mathrm{VC}(\mathbb{H}) = n + 1$?

Przykład: sinus

Dziedzina: $X = \mathcal{R}$, atrybut $a_1(x) = x$.

Przestrzeń modeli: $\mathbb{H}$ zawiera wszystkie modele reprezentowane przez funkcję *sinus*:

$$h(x) = \begin{cases} 1 & \text{jeśli } \sin(\alpha x) \geq 0 \\ 0 & \text{w przeciwnym przypadku} \end{cases}$$

dla dowolnie ustalonego parametru α.

Wymiar VC: dla dowolnego k przykłady $\frac{1}{2}, \frac{1}{4}, \ldots, \frac{1}{2^k}$ mogą być dowolnie etykietowane modelami z $\mathbb{H}$ – a więc $\mathrm{VC}(\mathbb{H}) = \infty$.

Przykład przestrzeni modeli reprezentowanych przez funkcję *sinus* podważa intuicję, jaką mogły nasuwać wcześniejsze przykłady, że wymiar VC jest zawsze ściśle związany z liczbą parametrów niezbędną do reprezentacji modeli (takich jak współrzędne wierzchołków prostokąta lub hiperprostopadłościanu albo współczynniki równania prostej lub hiperpłaszczyzny). Tak może często być, lecz W przypadku funkcji *sinus* mamy jeden parametr, który może być tak dobrany, aby dowolnie etykietować dowolnie wiele przykładów, co oznacza, że wymiar VC jest nieskończony.

Przykład: koniunkcje

$n = 3$: $\mathrm{VC}(\mathbb{H}) \geq 3$ (wszystkie etykietowania możliwe dla 100, 010, 001).

$n > 3$: $\mathrm{VC}(\mathbb{H}) = n$?

5.2 Zastosowanie

Wymiar VC może być użyty jako miara złożoności przestrzeni modeli do wyprowadzenia ograniczeń na wymaganą liczbę przykładów dla scenariuszy spójnego uczenia się i agnostycznego uczenia się, zastępując w tej roli rozmiar przestrzeni modeli i eliminując założenie o jej skończoności. Wyprowadzenie tych ograniczeń jest bardziej złożone i nie będziemy go przedstawiać, podając gotowe wyniki. Jednak ich interpretacja jest taka sama, jak analogicznych wyników przedstawionych poprzednio.

Spójne uczenie się: do uzyskania przez spójny algorytm uczenia się z prawdopodobieństwem co najmniej $1 - \delta$ model o błędzie rzeczywistym nieprzekraczającym ϵ wystarczy:

$$m \geq \frac{1}{\epsilon}\left(4 \log_2 \frac{2}{\delta} + 8\mathrm{VC}(\mathbb{H}) \log_2 \frac{13}{\epsilon}\right)$$

przykładów trenujących.

Agnostyczne uczenie się: z prawdopodobieństwem $1 - \delta$:

$$e_{\Omega,c}(h) \leq e_{T,c}(h) + \sqrt{\frac{1}{m}\left(\mathrm{VC}(\mathbb{H})\left(\ln \frac{2m}{\mathrm{VC}(\mathbb{H})} + 1\right) + \ln \frac{4}{\delta}\right)}$$

6 Podsumowanie wniosków z teorii

- Zbyt bogata przestrzeń modeli – duże $|\mathbb{H}|$ lub duże $\mathrm{VC}(\mathbb{H})$ – zwiększa ryzyko nadmiernego dopasowania (potrzeba więcej przykładów trenujących, aby uzyskać z wystarczającą pewnością wystarczająco mały błąd).

- Zbyt uboga przestrzeń modeli zwiększa ryzyko niedopasowania (mniejsza szansa, że istnieje w model o wystarczająco małym błędzie).
- Konieczna równowaga.
- W praktyce spójne uczenie się zwykle nie jest pożądane, gdyż nie ma dostępu do wystarczająco wielu przykładów trenujących, a często nie jest także możliwe, gdyż używana przestrzeń modeli nie zawiera modelu o zerowym błędzie na zbiorze trenującym (np. dlatego, że zestaw dostępnych atrybutów nie wystarcza do odróżniania przykładów różnych klas).
- Praktyczne algorytmy mogą używać pojemnych przestrzeni modeli, lecz zmniejszać efektywny wymiar VC przez dodatkowe mechanizmy ograniczające nadmierne dopasowanie.
- Brzytwa Ockhama – preferencja dla prostych modeli.

Teoria indukcyjnego uczenia się pojęć, której wybrane podstawowe elementy przedstawiliśmy, dostarcza pewnych istotnych gwarancji dotyczących możliwości uzyskania modeli o małym błędzie. Dzięki niej oczekiwanie, że indukcyjne uczenie się „działa", nie jest oparte tylko na nadziei i intuicji, lecz również na pewnych twardszych podstawach. Jednak intuicja także może korzystać z tej teorii, pozwalając nam lepiej zrozumieć, co czyni zadanie uczenia się „trudnym". Okazuje się, że „pojemne" przestrzenie modeli, które mogłyby się wydawać korzystne jako zwiększające szansę, że zawierają odpowiednio dobre modele, zwiększają złożoność uczenia się, wymagania na liczbę przykładów i ryzyko nadmiernego dopasowania.

Nasza dyskusja dotyczyła dwóch scenariuszy uczenia się, spójnego i agnostycznego. Spośród nich ten drugi jest bliższy potrzeb praktycznych – spójne uczenie się dla rzeczywistych zadań jest często niemożliwe (jeśli zbiór trenujący zawiera przykłady o jednakowych wartościach atrybutów należące do różnych klas), a także często niepożądane ze względu na ryzyko nadmiernego dopasowania (jeśli liczba przykładów trenujących nie jest na tyle duża, aby błąd rzeczywisty mógł być wystarczająco mały).

Praktyczne algorytmy niekiedy godzą używanie bogatych przestrzeni modeli (o dużym wymiarze VC) z ograniczeniem ryzyka nadmiernego dopasowania przez stosowanie mechanizmów zabezpieczających, które duży „teoretyczny" wymiar VC redukują do mniejszego „efektywnego" wymiaru. Takim mechanizmem może być np. preferencja dla prostszych modeli, określana jako zasada brzytwy Ockhama – co jest nawiązaniem do średniowiecznego filozofa Wilhelma Ockhama, który wzywał do preferowania prostszych wyjaśnień pytań metafizycznych.