Po co maszynowe uczenie?

Można powiedzieć, że geneza maszynowego uczenia jest dwojaka:

sztuczna inteligencja

Ludzka inteligencja jest nierozerwalnie związana z uczeniem się. Każde działanie człowieka jest związane z analizowaniem co można ulepszyć, jak uprościć sobie życie, skrócić czas wykonywania danej pracy, ułatwić sobie zadanie. Ponowne wykonywanie tej samej pracy zawsze bierze pod uwagę doświadczenia z przeszłości.

Algorytmy sztucznej inteligencji nie mają tej właściwości. Większość stosowanych w sztucznej inteligencji modeli nie ulepsza się samoistnie w trakcie pracy. Ten mechanizm trzeba zaprogramować oddzielnie. Naturalne jest więc poszukiwanie algorytmów uczenia się. Maszynowe uczenie wywodzi się, i oryginalnie było częścią sztucznej inteligencji.

big data

Żyjemy w świecie nadmiaru informacji. Ta informacja jest potrzebna, można pożytecznie ją wykorzystać do wielu celów, ale nie jest łatwo jej użyć, właśnie przez jej nadmiar. Metody automatycznej analizy danych są potrzebne.

Istnieje szereg podejść w maszynowym uczeniu. Starsze metody są oparte na wiedzy. Nowsze metody oparte są głównie na prawdopodobieństwie i statystyce. Najlepsze metody łączą jedno i drugie.

Rodzaje maszynowego uczenia

Uczenie maszynowe ma kilka odmian, z którymi związane są różne modele i algorytmy.

uczenie nadzorowane (supervised, predictive)

Uczymy się odwzorowania wejścia x do wyjścia y na podstawie serii uczącej $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ gdzie N jest liczbą przykładów uczących.

W najprostszym przypadku każde wejście x_i jest wektorem liczb reprezentujących próbkę, zwanych atrybutami lub cechami (*features*). W ogólnym przypadku może to jednak być jakiś obiekt, np. obraz, dokument tekstowy, graf, szereg czasowy, itp.

Wartość y może być wartością dyskretną (inaczej: **kategoryczną** lub **nominalną**), czyli przyjmującą wartości z pewnego skończonego zbioru, np. $\{$kobieta, mężczyzna$\}$, albo może być wartością liczbową typu real. W pierwszym przypadku problem uczenia nazywamy problemem **klasyfikacji** albo **rozpoznawania wzorców**, a w drugim problemem **regresji**.

uczenie nienadzorowane (unsupervised, descriptive)

W tym przypadku uczymy się na podstawie zbioru wejść $\mathcal{D} = \{x_i\}_{i=1}^N$ gdzie celem jest znalezienie „interesujących wzorców" w danych. Zadanie to jest gorzej określone niż uczenie nadzorowane, ponieważ np. nie ma jasnego kryterium oceny jakości uczenia.

uczenie ze wzmocnieniem (reinforcement)

Jeszcze trudniejszy scenariusz, gdzie celem jest nauczenie się działania na podstawie okresowo otrzymywanych sygnałów zwanych nagrodami, albo wzmocnieniami. Agent uczy się przez działanie; wzmocnienia otrzymuje on w odpowiedzi na swoje konkretne akcje.

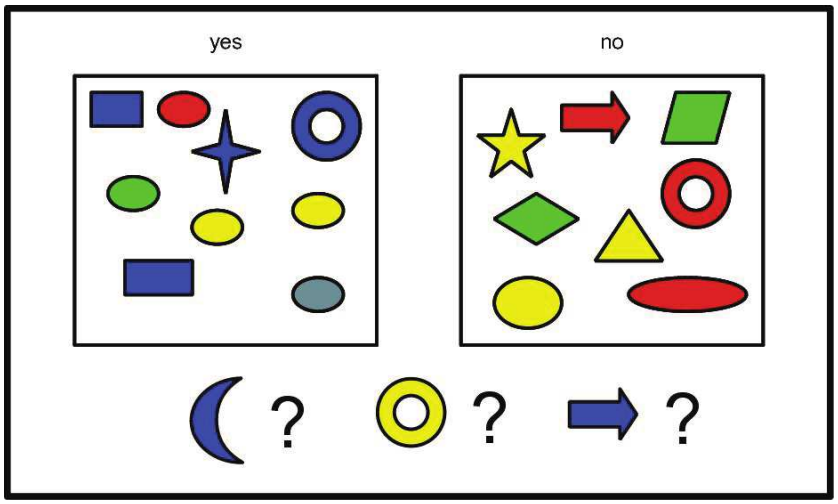
Celem jest takie działanie, aby maksymalizować pozytywne wzmocnienia (nagrody) a minimalizować negatywne (kary). Dobrą metaforą uczenia ze wzmocnieniem jest uczenie się przez dziecko: od podstawowego poruszania się w świecie, takiego jak nauka chodzenia, do „poprawnego" (skutecznego) zachowania się w trudnym i niezrozumiałym świecie ludzi dorosłych.

Uczenie nadzorowane — klasyfikacja

Celem jest nauczenie się odwzorowania z wejścia x na wyjście y gdzie $y \in \{1, ..., C\}$ a C oznacza liczbę klas. Gdy $C = 2$ to jest to **klasyfikacja binarna**, a gdy $C > 2$ to **klasyfikacja wieloklasowa**. Przypadek gdy klasy wyjściowe nie są rozłączne (np. ktoś może być zaklasyfikowany jednocześnie jako *tall* i *strong*) jest nazywany **klasyfikacją wieloetykietową**. W tym przypadku często lepiej zbudować **model wielowyjściowy** określający wiele powiązanych klas binarnych (lub wieloklasowych). Domyślnie przez klasyfikację będziemy rozumieli klasyfikację wieloklasową z jednym wyjściem.

Można uogólnić problem klasyfikacji jako zadanie **aproksymacji funkcji**. Zakładamy $y = f(x)$ dla pewnej nieznanej funkcji f i należy estymować funkcję f na podstawie danego zbioru uczącego przykładów etykietowanych, a następnie wykonywać predykcje $\hat{y} = \hat{f}(x)$. Głównym celem jest dokonywanie predykcji na nieznanych wcześniej wektorach wejściowych, ponieważ dla wektorów z serii uczącej wystarczyłoby zapamiętać i potem odszukać wartość wyjścia. Zatem uczenie się wymaga **generalizacji**.

Uczenie nadzorowane — klasyfikacja: przykład



Ponieważ próba dokonania klasyfikacji tak na podstawie samego kształtu jak i samego koloru nie wydaje się sensowna, próbujemy opisać próbki za pomocą zestawu D cech (atrybutów) reprezentując problem klasyfikacji za pomocą macierzy $N \times D$.

D features (attributes)

N cases

Color	Shape	Size (cm)
Blue	Square	10
Red	Ellipse	2.4
Red	Ellipse	20.7

Label
1
1
0

Uczenie nadzorowane — predykcja probabilistyczna

Aby rozwiązać przypadki dwuznaczne, takie jak żółte kółko w powyższym przykładzie, dobrym rozwiązaniem jest wyznaczenie rozwiązania w postaci prawdopodobieństwa. Oznaczymy rozkład prawdopodobieństwa na zbiorze wszystkich etykiet dla wektora wejściowego x i zbioru uczącego $\mathcal{D}$ jako $p(y|x, \mathcal{D})$. W ogólnym przypadku reprezentuje ono wektor o wymiarze C (w przypadku binarnym wystarcza pojedyncze prawdopodobieństwo). W przypadku gdy będziemy rozważać różne modele M uczenia, ten rozkład prawdopodobieństwa będzie również zależny od modelu, i poprawne będzie podkreślenie tego w oznaczeniu $p(y|x, \mathcal{D}, M)$.

Gdy wynik klasyfikacji jest wektorem prawdopodobieństw, sensowne jest wybrać jako wynik klasyfikacji wartość:

$$\hat{y} = \hat{f}(x) = \underset{c=1}{\overset{C}{\operatorname{argmax}}}\ p(y = c|x, \mathcal{D})$$

Odpowiada ona etykiecie najbardziej prawdopodobnej klasy, i zwane jest **estymatą MAP** (MAP - *maximum a posteriori*). Ze względu na probabilistyczny charakter wyniku klasyfikacji, mówimy w tym wypadku o **predykcji probabilistycznej**.

Uczenie nadzorowane — predykcja probabilistyczna (2)

Rozważmy ponownie przypadek klasyfikacji żółtego kółka z poprzedniego przykładu. Ten przypadek budzi poważne wątpliwości, więc wyznaczony rozkład prawdopodobieństwa $p(\hat{y}|x, \mathcal{D})$ będzie daleki od zerojedynkowego. Tzn. wybrana klasa MAP będzie miała prawdopodobieństwo raczej bliżej 0.5 niż 1.0. Na przykład, rozkład prawdopodobieństw dla istniejących kategorii może być: $\langle 0.55; 0.48; 0.36; \ldots \rangle$.

W wielu takich przypadkach lepiej odpowiedzieć „nie wiem" niż dawać odpowiedź o małej wiarygodności. Ma to szczególne znaczenie w dziedzinach takich jak medycyna lub finanse, gdzie w podejmowaniu decyzji wolimy unikać ryzyka. Dotyczy to nie tylko zastosowań „poważnych" gdzie od decyzji zależy czyjeś życie lub powodzenie ważnego przedsięwzięcia. Również w zastosowaniach takich jak gry komputerowe, teleturnieje lub testy akademickie, „strzelanie" często się nie opłaca. Na udzieleniu złej odpowiedzi możemy stracić (punkty lub pieniądze), podczas gdy odpowiedź „nie wiem" jest bezpieczniejsza.

Uczenie nienadzorowane

W uczeniu nienadzorowanym mamy do czynienia z surowymi danymi, które możemy traktować jako wejściowe, bez określenia wyjścia. Celem uczenia jest wykrycie ciekawych lub istotnych wzorców w danych.

Tej enigmatycznej definicji nie sposób przecenić. Uczenie nienadzorowane jest znacznie bardziej zbliżone do typowego uczenia ludzi i zwierząt. Ma również o wiele większe zastosowania niż uczenie nadzorowane.

Można to uzasadnić argumentem ilościowym. Aby zrealizować uczenie nadzorowane trzeba stworzyć dane uczące, co jest kosztowne, i dalece ograniczone. Na przykład (Geoff Hinton), ludzie uczą się widzieć po prostu patrząc. Co jakiś czas otrzymujemy wskazówki takie jak: „to jest pies", albo „to jest autobus." Jednak ilość takich wskazówek jest ograniczona, np. do 1 bita na sekundę. Biorąc pod uwagę, że system widzenia w mózgu ma 10^{14} połączeń, a człowiek żyje $\approx 10^9$ sekund, nadzorowane uczenie widzenia jest silnie limitowane. Nienadzorowane uczenie może przetworzyć 10^5 bitów na sekundę, i ma zdecydowaną przewagę.

Uczenie nienadzorowane (2)

Rodzaje uczenia nienadzorowanego:

wykrywanie skupień (*clustering*)
chcemy pogrupować dane opisane różnymi parametrami w naturalne skupiska

wykrywanie istotnych parametrów (*discovering latent factors*)
dla zbioru danych opisanych wieloma parametrami, chcemy wybrać te najistotniejsze

określanie struktury grafu
określanie które ze zbioru zmiennych są najbardziej skorelowane z którymi innymi

uzupełnianie danych
uzupełnianie brakujących danych w bazach, statystykach

filtrowanie grupowe (*collaborative filtering*)
określenie co może spodobać się danemu konsumentowi, biorąc pod uwagę jego i innych konsumentów wcześniejsze oceny różnych produktów (np. Netflix)

analiza tendencji zakupowych (*market basket analysis*)
mając dane o zakupach realizowanych w ramach wcześniejszych transakcji, chcemy przewidzieć co jeszcze dany klient mógłby chcieć dorzucić do swojego koszyka

Uczenie się indukcyjne — model ogólny

Najprostsza forma: uczenie się nieznanej funkcji $f(x)$ na podstawie serii par $\langle x, f(x)\rangle$.
Przykład zbioru par uczących: $\{\langle$  $, 0\rangle, \langle$  $, 1\rangle\}$.

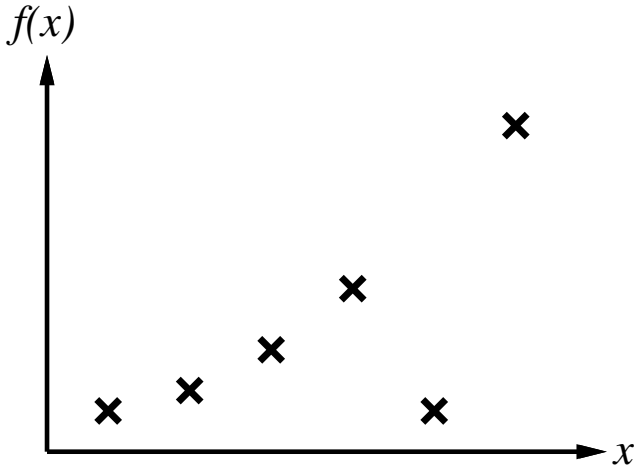
Dokładniej: na podstawie serii przykładów uczących należy znaleźć hipotezę $h \in \mathcal{H}$ ($\mathcal{H}$ nazywamy **przestrzenią hipotez**) taką, że $h \approx f$, tzn. hipoteza h przybliża nieznaną funkcję f w sensie jakiegoś kryterium. W najprostszym przypadku jako kryterium możemy przyjąć minimalizację liczby elementów zbioru uczącego, dla których $h(x) \neq f(x)$.

W rzeczywistości jednak celem uczenia jest uogólnianie. Zatem zależy nam nie tyle na poprawnym sklasyfikowaniu elementów zbioru uczącego, co na uchwyceniu ogólnej zasady, według której zostały one sklasyfikowane. Zdolność skutecznego uczenia się w tym sensie zależy od:

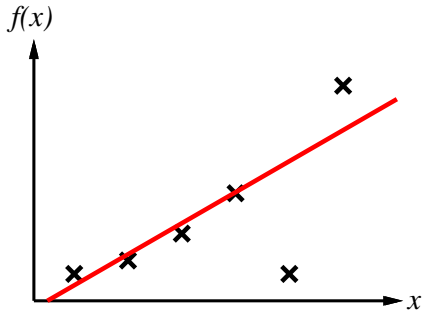
- zbioru uczącego, zarówno jego liczności jak i konkretnego doboru elementów,
- stosowanego algorytmu uczenia,
- rozważanej przestrzeni hipotez $\mathcal{H}$ (która jest związana z algorytmem uczenia); jeśli będzie ona zbyt uboga to skuteczne uczenie może okazać się niemożliwe.

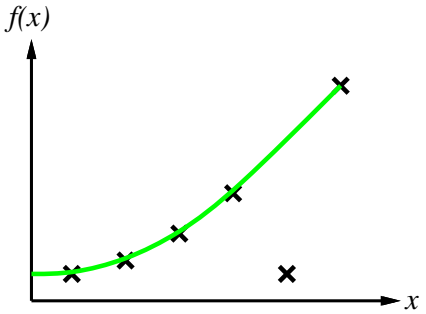
Uczenie się indukcyjne — przykład

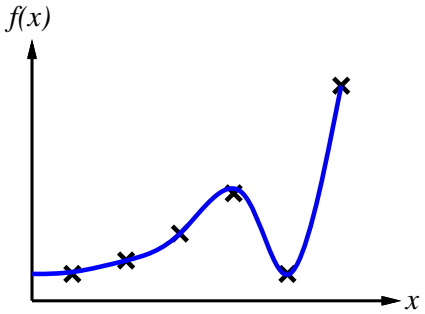
Na przykład, dla funkcji zadanej następującym zbiorem wartości:

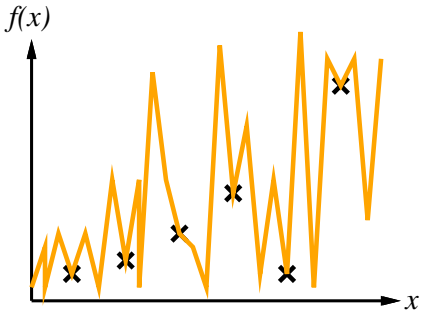


Możemy rozważać następujące hipotezy:





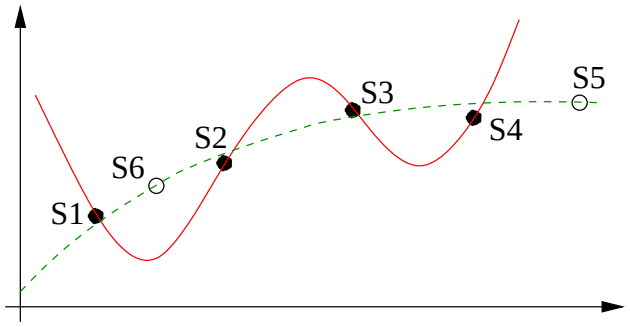




Zasada brzytwy Ockhama: spośród hipotez, które jednakowo dobrze wyjaśniają dane zjawisko, wybierz najprostszą — tę która wymaga dokonania najmniejszej liczby założeń.

Przeuczenie

Częstym zjawiskiem w uczeniu maszynowym jest **przeuczenie** (*overfitting*). Objawia się **wykrywaniem pozornych prawidłowości** w dużej ilości danych, gdzie prawdziwe prawidłowości są prostsze lub słabsze, lub są maskowane przez błędy, lub są całkowicie nieistniejące. Jest to możliwe gdy bogata przestrzeń hipotez $\mathcal{H}$ zawiera — między innymi — hipotezy h dużo bardziej złożone niż poszukiwana funkcja f.



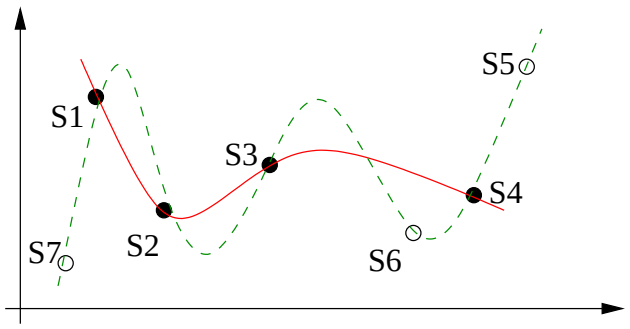
Punkty S1,...,S4 reprezentują krzywą przerywaną, z drobnymi błędami. Jeśli proces uczący, dążąc do **nadmiernego dopasowania**, zastosuje zbyt wysokiego rzędu wielomian, to może wygenerować krzywą ciągłą. Objawem będą duże błędy dla punktów walidacyjnych/testowych S5, S6.

Inny przykład: uczymy się prawidłowości wyników rzucania kostką, i pośród atrybutów wejściowych mamy: numer rzutu, imię rzucającego, dzień tygodnia rzutu, kolor kości, itp. Pożądanym rezultatem byłoby wykrycie niezależności wyniku od tych czynników, ale w niektórych przypadkach algorytm może wykryć pewne marginalne zależności.

Przeuczenie można wykryć stosując pewien rodzaj testowania zwany walidacją. Jednak trudno jest je zwalczyć, ponieważ ogólnie chcemy mieć silne algorytmy uczenia i duże zbiory danych, które w rzeczywistych warunkach niemal zawsze zawierają błędy.

Niedouczenie

Jak zauważyliśmy wyżej, duże wartości błędów walidacji, następujące po skutecznym uczeniu, mogą być objawem przeuczenia. Jednak nie zawsze. Mogą one również być objawem **niedouczenia** (*underfitting*) wynikającego z niewystarczającej liczby próbek uczących, lub ze zbyt uproszczonego modelu zastosowanego w uczeniu (zbyt ubogiej przestrzeni hipotez $\mathcal{H}$).



Uczenie punktów S1,...,S4 może wygenerować krzywą ciągłą. Punkty walidacyjne/testowe S5, S6, S7 dają duży błąd, ale w tym przypadku jest to wynik niedouczenia. Ponowne uczenie z tymi punktami prowadzi do otrzymania krzywej przerywanej.

Określanie, czy błędy otrzymane w testowaniu klasyfikatora wynikają z przeuczenia, niedouczenia, lub może czegoś jeszcze innego, jest trudnym problemem w uczeniu maszynowym i często wymaga przeprowadzenia wielu eksperymentów.

Modele parametryczne i nieparametryczne

Inne rozróżnienie pomiędzy metodami maszynowego uczenia wynika z tego, czy liczba parametrów budowanego modelu jest stała, czy rośnie wraz z liczbą próbek serii uczącej. Jeśli stała to mówimy o **modelu parametrycznym**, w przeciwnym wypadku o **modelu nieparametrycznym**. Ten podział pojawia się zarówno w uczeniu nadzorowanym (klasyfikacja, regresja), jak i nienadzorowanym.

Modele parametryczne są typowo szybsze, lecz wymagają przyjęcia silniejszych założeń co do dystrybucji danych w rozpatrywanej dziedzinie. Nieparametryczne modele są bardziej elastyczne, ale przy wielkich zbiorach danych mogą być obliczeniowo niepraktyczne.

Przykładem modelu nieparametrycznego może być algorytm klasyfikacji kNN (Najbliższych Sąsiadów). Polega on na dokonaniu klasyfikacji nowej próbki przez analizę najbliższych jej sąsiadów zbioru uczącego w przestrzeni cech.

Przykładem modelu parametrycznego może być algorytm regresji liniowej przybliżający wartość nieznanej funkcji. Metoda oblicza współczynniki wzoru przy założeniu, że próbki zbioru uczącego zostały wygenerowane z rozkładem normalnym (gausowskim), co ma sens dla wielu zjawisk koncentrujących się wokół pewnych wartości średnich.

Modele klasyfikacji

- klasyfikacja oparta na zbiorze próbek, brak modelu

  np. k-najbliższych sąsiadów

- budowa modelu generatywnego, tzn. łącznej dystrybucji $p(x, y)$, i klasyfikacja z użyciem reguły Bayesa w celu obliczenia prawdopodobieństwa warunkowego $p(y|x)$ w celu wybrania etykiety y

  np. naiwny klasyfikator bayesowski, sieci bayesowskie

- budowa modelu dyskryminatywnego, tzn. rozkładu warunkowego $p(y|x)$, co pozwala na bezpośrednią klasyfikację próbek

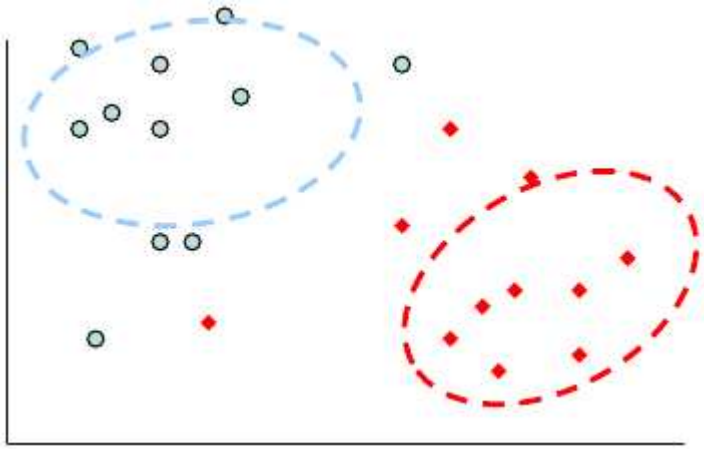
  np. drzewa decyzyjne, regresja logistyczna, maszyny wektorów nośnych (SVM), sieci neuronowe

Modele generatywne i dyskryminatywne

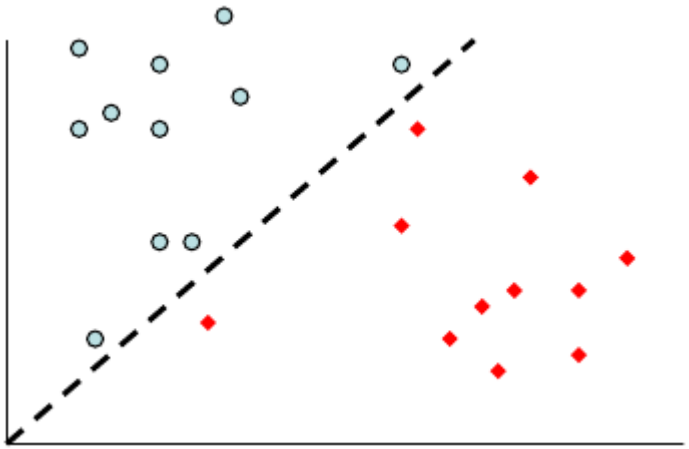
Budowa modelu generatywnego wykorzystuje wszystkie próbki.
Zbudowany model $p(x, y)$ może być wykorzystany na różne sposoby, np. do generowania nowych próbek zgodnych z oryginalną dystrybucją.

W budowie modelu dyskryminatywnego interesuje nas głównie granica między klasami.

generatywny



dyskryminatywny



Można znaleźć wiele dyskusji filozoficznych na temat wyższości jednego podejścia nad drugim. Częstym jest pogląd, że modele dyskryminatywne ogólnie górują nad generatywnymi, lecz przy małej ilości danych treningowych to modele generatywne mogą być korzystniejsze.

Uczenie maszynowe jest trudne

Nie ma jednego podejścia, które gwarantuje sukces w uczeniu maszynowym. Wynika to z konieczności generalizacji, która jest zawsze pewną heurystyką. Założenia, które działają poprawnie dla jednej dziedziny problemowej, mogą zawodzić w innej.

To zjawisko bywa formułowane żartobliwie jako „*no free lunch theorem*."

Aby je przybliżyć, wyobraźmy sobie zadanie klasyfikacji, w którym po przeanalizowaniu pewnej liczby próbek zbioru uczącego, został wygenerowany klasyfikator. Przyjmijmy, że jest to klasyfikacja binarna True/False.

Rozpatrzmy teraz nową próbkę, spoza zestawu uczącego. Jej klasą może być wartość True lub False. Przyjmijmy, że wygenerowany klasyfikator określił ją jako True. Jednak w rzeczywistości klasą próbki może być False. Gdyby algorytm uczenia maszynowego to wiedział, to wygenerowałby nieco inny model, który być może określiłby wartość tej próbki odmiennie.

Materiały

W tej prezentacji wykorzystane zostały materiały z następujących prac:

1. Stuart J. Russell, Peter Norvig: Artificial Intelligence A Modern Approach (Third Edition), Prentice-Hall, 2010

2. Kevin P. Murphy: Machine Learning A Probabilistic Perspective, MIT Press, 2012