

1 Uczenie maszynowe

1.1 Potrzeba uczenia maszynowego

Człowiek nabywa umiejętności ucząc się. Większość dotychczas omawianych systemów polega na przeszukiwaniu z góry zaprogramowanych możliwości i korzysta z gotowej wiedzy zakodowanej w odpowiedni sposób przez eksperta. Prawdziwie inteligentne systemy muszą uczyć się na podstawie swoich błędów, wyciągać z nich wnioski i modyfikować swoje reguły lub wewnętrzne reprezentacje. Machine Learning (ML), czyli uczenie maszynowe jest jedną z najważniejszych poddziedzin sztucznej inteligencji. Pokrywa się też z wieloma innymi dziedzinami, czerpiąc idee z informatyki, statystyki, teorii rozpoznawania struktur (pattern recognition) i kognitywistyki. dotychczas rozważane programy służyły jedynie do zapisywania i wykorzystania wiedzy znalezionej przez człowieka. Chociaż ze względu na stopień ich skomplikowania nie można z góry (bez przeprowadzenia obliczeń) przewidzieć, do jakich wniosków dojdą czy jakie decyzje podejmą, to jednak korzystają one z wiedzy w jawnej postaci i można się upierać, że wiedzą tyle, ile wpisał w program jego twórca. systemy ML mają za zadanie uczyć się na przykładach, a więc po pewnym czasie nawet ich twórca nie wie, jaką wiedzę dysponuje taki system. Uczenia Maszynowe umożliwia więc mechanizację akwizycji wiedzy i jest alternatywą dla tworzenia reguł przez ekspertów lub inżynierów wiedzy.

Często dane są w postaci przykładów a rozwiązanie problemu redukuje się do klasyfikacji. Eksperci opisują sytuacje podając przykłady znacznie częściej niż podając reguły, jakie stosują. Czasami reguły otrzymane metodami ML są znacznie lepsze niż reguły wydedukowane przez ludzi. ML oferuje więc szybszą drogę do tworzenia systemów eksperckich. Porównanie czasów rozwoju systemów eksperckich i systemów wykorzystujących metody uczenia maszynowego, przedstawione w tabeli poniżej, pokazuje, że szybkość tworzenia systemów eksperckich wykorzystujących możliwości uczenia maszynowego może być nawet o dwa rzędy wielkości większa. Dokładność czy jakość działania takich systemów jest często wyższa, np. system AIOCS do klasyfikacji ankiet opracowanych w ramach spisu powszechnego w USA, którego koszt opracowania szacowano na 16 osobolat, działał średnio o 20% mniej dokładnie niż system PACE oparty na technikach ML, opracowany w ciągu 4 miesięcy.

Nazwa	Typ	L. reguł	Czas (lat)	Poprawki (lat)
MYCIN	ES	1000	100	?
XCON	ES	8000	180	30
GASOIL	ML	2800	1	0,1
BMT	ML	30.000	9	2

Do ML można zaliczyć wszystkie metody adaptacji, łącznie z sieciami neuronowymi i algorytmami genetycznymi.

1.2 Klasy metod ML

Powszechnie stosowane :

- Drzewa decyzji/indukcja reguł
- Rozumowanie oparte na analogiach (case-based)
- i oparte na pamięci (memory-based)
- Logika rozmyta
- Sieci neuronowe

Pojawiają się:

- version spaces
- algorytmy genetyczne

sieci probabilistyczne (Baysowskie)
zbiory i logika przybliżona
induktywne programowanie logiczne (ILP)
uczenie ze wzmocnieniem (reinforcement learning)

Zintegrowane pakiety programów z elementami ML, np. MLC++

Dużo zastosowań w przemyśle maszynowym, naftowym (BP), chemicznym, finansach (American Express), komputerowym (Siemens), określanie jakości oprogramowania (NEC), eksploracja baz danych (data mining)
Oszczędności rzędu M\$/rok !

Zastosowania w nauce: katalogowanie obiektów

Nowe zastosowania: komputerowy czeladnik - obserwacja zachowania eksperta, interakcja

1.3 Version spaces

Opis - książka Nillsona, podobne do logiki przybliżonej.

S= Najbardziej specyficzny zbiór hipotez zgadzający się z danymi;

G = Najbardziej ogólny zbiór hipotez

H = Przestrzeń hipotez

D = Dane

V(H,D) - version space, przestrzeń hipotez zgodnych z danymi; zbiory hipotez T przedstawia się w postaci węzłów siatki (lattice);

Można w niej wprowadzić relację częściowego porządku: teoria $T_1 > T_2$ (jest bardziej ogólna niż) jeśli wszystkie przykłady, które może sklasyfikować T_2 może i T_1 .

1.4 Ogólna teoria uczenia maszynowego

Problem: dla danego ciągu treningowego sklasyfikowanych przykładów $\{\mathbf{X}^k, C_k\}$ i nieznanego wektora \mathbf{X} użyć miary podobieństwa (odległości) $d(\mathbf{X}, \mathbf{X}_k)$ do oceny prawdopodobieństwa klasyfikacji $p(C_i | \mathbf{X}; M)$, gdzie M jest opisem modelu.

Prawdopodobieństwo przypisania klasy C_i danym \mathbf{X} przedstawić można jako

$$p(C_i | \mathbf{X}, k, G(d(\|\mathbf{X} - \mathbf{X}_r\|; r)), \mathbf{R}_n) \quad (0.1)$$

gdzie k =liczba sąsiadów, $G(d(\cdot, r))$ funkcja ważąca wpływ odległych sąsiadów; $d(\cdot, r)$ funkcja odległości sparametryzowana przez promień r i inne nie wymienione tu parametry, a \mathbf{R}_n to wektory referencyjne. Takie modele mogą się różnić wieloma cechami, w zależności od wyboru procedur i parametrów.

1.1.4. k-NN

W najprostszym przypadku mamy metodę najmniejszych sąsiadów, w której ustalony jest jakiś sposób obliczania odległości - np. Euklidesowe odległości dla standaryzowanych danych (przesuniętych do średniej każdego z atrybutów i podzielonych przez standardową wariancję). Jedynym parametrem jest tu liczba wektorów referencyjnych k uwzględnianych w sąsiedztwie \mathbf{X} . Wektorami referencyjnymi są wszystkie wektory treningowe. Parametr k należy zoptymalizować na danych należących do zbioru pośredniego (validation set); optymalizacja na zbiorze testowym jest pewnym oszustwem, gdyż klasyfikator nie ma prawa optymalizować parametrów na zbiorze testowym. Z drugiej strony dla małych zbiorów danych trudno będzie trudno podzielić je na trzy części: treningową, testową i potrzebną do optymalizacji. Dla małych zbiorów danych nie należy się też spodziewać, by metoda k-NN dawała dobre rezultaty, dopiero przy dużej liczbie próbek, gdy granice decyzji pomiędzy poszczególnymi klasami są „gęsto obsadzone” przez wektory referencyjne, rezultaty bywają bardzo dobre.