

> Uczenie maszynowe (ang. machine learning) – dziedzina wchodząca w skład nauk zajmujących się problematyką sztucznej inteligencji. Jest to dziedzina interdyscyplinarna ze szczególnym uwzględnieniem takich dziedzin jak informatyka, robotyka i statystyka.

Głównym celem jest praktyczne zastosowanie dokonań w dziedzinie sztucznej inteligencji do stworzenia automatycznego systemu potrafiącego doskonalić się przy pomocy zgromadzonego doświadczenia (czyli danych) i nabywania na tej podstawie nowej wiedzy.

http://pl.wikipedia.org/wiki/Uczenie_maszynowe

Uczenie drzew decyzyjnych (ang. Decision Tree Learning) – drzewo decyzyjne to graficzna metoda wspomagania procesu decyzyjnego, stosowana w teorii decyzji. Algorytm drzew decyzyjnych jest również stosowany w uczeniu maszynowym do pozyskiwania wiedzy na podstawie przykładów. Jest to schemat o strukturze drzewa decyzji i ich możliwych konsekwencji. Zadaniem drzew decyzyjnych może być zarówno stworzenie planu, jak i rozwiązanie problemu decyzyjnego. Metoda drzew decyzyjnych jest szczególnie przydatna w problemach decyzyjnych z licznymi, rozgałęziającymi się wariantami

http://en.wikipedia.org/wiki/Decision_tree_learning

Uczenie z przykładów (ang. Instance-based Learning) – w odróżnieniu od metod uczenia, które konstruują ogólny, tzw. jawny opis funkcji docelowej, kiedy dostarczane są dane uczące, uczenie tego typu po prostu zapamiętuje przykłady. Uogólnianie nad tymi przykładami jest odwołane do czasu, aż nowy przykład (zadanie) ma być klasyfikowane. Za każdym razem, kiedy przychodzi nowe zapytanie

(przykład), badane są jego powiązania z zapamiętanymi przykładami aby ustalić wartość docelowej funkcji nowego przykładu.

http://en.wikipedia.org/wiki/Instance-based_learning

Uczenie się zbioru reguł (ang. Learning Sets of Rules) – zbiór reguł w postaci klauzul Hornowskich może być interpretowany jako program w np. języku Prolog

http://pl.wikipedia.org/wiki/Klauzula_Horna

Uczenie indukcyjne i analityczne (ang. Inductive and Analytical Learning) – metody uczenia indukcyjnego (wykorzystują sieci neuronowe, drzewa decyzyjne), wymagają pewnej liczby przykładów aby osiągnąć pewien poziom uogólnienia. Analityczne uczenie stosuje wiedzę aprioryczną i wnioskowanie dedukcyjne do powiększania informacji dostarczanej przez przykłady uczące. Czyste indukcyjne uczenie formułuje ogólne hipotezy poprzez znalezienie empirycznych regularności w przykładach uczących. Natomiast czyste analityczne uczenie stosuje aprioryczną wiedzę do otrzymania ogólnych hipotez dedukcyjnie. Połączenie obu podejść daje korzyści: lepszą poprawność i trafność uogólniania gdy dostępna jest wiedza aprioryczna oraz szukanie zależności w obserwowanych danych uczących do wypracowania szybkiej wiedzy apriorycznej.

http://cse-wiki.unl.edu/wiki/index.php/Combining_Inductive_and_Analytical_Learning

Uczenie przez wzmacnianie (ang. Reinforcement Learning) – uczenie przez wzmacnianie to metoda wyznaczania optymalnej polityki sterowania przez agenta w nieznanym mu środowisku, na podstawie interakcji z tym środowiskiem. Jedyną informacją, na której agent się opiera jest sygnał wzmocnienia (poprzez wzorowanie się na pojęciu wzmocnienia z nauk behawioralnych w psychologii), który osiąga wysoką wartość (nagrodę), gdy agent podejmuje poprawne decyzje lub niską (karę) gdy podejmuje decyzje błędnie.

http://en.wikipedia.org/wiki/Reinforcement_learning

> Uczenie/Metody bayesowskie (Bayesian Learning), Statistical learning

Na uczenie bayesowskie składają się metody oparte na twierdzeniu sformułowanym przez XVIII-wiecznego matematyka Thomasa Bayesa. Odgrywają znaczną i ostatnio rosnącą rolę w dziedzinie sztucznej inteligencji, zwłaszcza w uczeniu się maszyn. Można ogólnie powiedzieć, że wzór Bayesa stał się podstawą do rozwoju teorii i algorytmów różnych form wnioskowania probabilistycznego.

<http://www.mimuw.edu.pl/~szczuka/mme/wyklad8>

Dysponując zbiorem etykietowanych przykładów i nowym przykładem chcielibyśmy umieć powiedzieć jaka jest **najbardziej prawdopodobna** wartość decyzji dla tego przykładu. Jakie jest prawdopodobieństwo przypisania nowego przykładu do jakiejś klasy w świetle posiadanych danych treningowych.

Notacja

T - zbiór danych treningowych.

T^d - zbiór danych treningowych o decyzji d .

$T_{a_i=v}^d$ - zbiór danych treningowych o wartości atrybutu a_i równej v i decyzji d .

\mathbb{H} - przestrzeń hipotez.

$0 \leq \Pr(A) \leq 1$ - prawdopodobieństwo A .

$\Pr(A|B)$ - prawdopodobieństwo zdarzenia A pod warunkiem B .

Prawdopodobieństwo

$$\Pr(A \wedge B) = \Pr(A) \Pr(A|B) = \Pr(B) \Pr(B|A).$$

$$\Pr(A \vee B) = \Pr(A) + \Pr(B) - \Pr(A \wedge B).$$

Jeśli A_1, \dots, A_n są wzajemnie wykluczające:

$$\sum_{i=1}^n \Pr(A_i) = 1 \Rightarrow \Pr(B) = \sum_{i=1}^n \Pr(B|A_i) \Pr(A_i)$$

Twierdzenie Bayesa

Sformułowanie, które wykorzystamy dalej.

Dla dowolnej hipotezy $h \in \mathbb{H}$ i zbioru danych $T \subset X$ zachodzi:

$$\Pr(h|T) = \frac{\Pr(T|h) \Pr(h)}{\Pr(T)}$$

Dowód: http://pl.wikipedia.org/wiki/Twierdzenie_Bayesa#Dow.C3.B3d

Przykład: http://en.wikipedia.org/wiki/Bayes%27_theorem#Introductory_example

$\Pr(h|T)$ - prawdopodobieństwo *a posteriori* hipotezy h przy posiadaniu danych T - tego szukamy.

$\Pr(T)$ - prawdopodobieństwo danych. Nie musimy go znać (na szczęście), żeby porównywać prawdopodobieństwa *a posteriori* hipotez.

Potrzebujemy wyznaczyć $\Pr(h)$ i $\Pr(T|h)$. Na razie zakładamy, że potrafimy je wyznaczyć, a także, że mamy ustalone \mathbb{H} .

Maximum A Posteriori - MAP

Mając dany zbiór T , klasyfikujemy nowy przykład $x \in X$ wykorzystując hipotezę $h_{MAP} \in \mathbb{H}$ czyli przypisujemy obiektowi x wartość decyzji zwróconą przez $h_{MAP}(x)$, gdzie:

$$h_{MAP} = \arg \max_{h \in \mathbb{H}} \Pr(h|T) = \arg \max_{h \in \mathbb{H}} \Pr(T|h) \cdot \Pr(h)$$

Maximum Likelihood - ML

Mając dany zbiór T , klasyfikujemy nowy przykład $x \in X$ wykorzystując hipotezę $h_{ML} \in \mathbb{H}$ czyli przypisujemy obiektowi x wartość decyzji zwróconą przez $h_{ML}(x)$, gdzie:

$$h_{ML} = \arg \max_{h \in \mathbb{H}} \Pr(T|h).$$

Przykład:

http://en.wikipedia.org/wiki/Maximum_likelihood#Discrete_distribution.2C_finite_parameter_space

Optymalny klasyfikator bayerowski

(Bayesian Optimal Classifier – BOC)

zawsze zwraca najbardziej prawdopodobną wartość decyzji dla danego przykładu i próbki uczącej. Nie może zatem być pokonany przez żaden algorytm uczący, jeśli porównujemy błędy rzeczywiste (globalne).

Niech $c(\cdot)$ będzie przybliżonym pojęciem, T próbką treningową.

$$h_{BOC} = \arg \max_{d \in C} \Pr(c(x) = d|T)$$

gdzie:

$$\Pr(c(x) = d|T) = \sum_{h \in \mathbb{H}} \Pr(c(x) = d|h) \Pr(h|T)$$

$$\Pr(c(x) = d|h) = \begin{cases} 1 & \text{if } h(x) = d \\ 0 & \text{if } h(x) \neq d \end{cases}$$

Szczegółowy fragment wykładu z przykładem: <https://class.coursera.org/machlearning-001/lecture/239>

Niech x_* będzie nowym przykładem, który mamy sklasyfikować. Powinniśmy wybrać taką hipotezę (decyzję) h , że:

$$h(x_*) = \arg \max_{d \in C} \Pr(c(x) = d | \bigwedge_{i=1}^n a_i(x) = a_i(x_*))$$

czyli, ze wzoru Bayesa

$$\arg \max_{d \in C} \Pr(c(x) = d) \cdot \Pr(\bigwedge_{i=1}^n a_i(x) = a_i(x_*) | c(x) = d)$$

Naiwny klasyfikator bayesowski

(Naïve Bayes classifier - NBC)

$$\arg \max_{d \in C} \Pr(c(x) = d) \cdot \prod_{i=1}^n \Pr(a_i(x) = a_i(x_*) | c(x) = d)$$

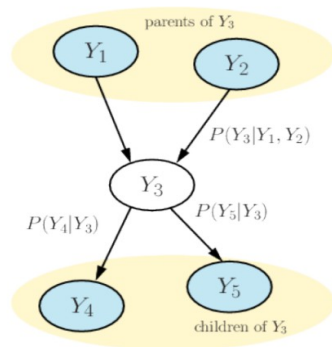
Przykład – klasyfikacja spamu:

http://pl.wikipedia.org/w/index.php?title=Naiwny_klasyfikator_bayesowski&veaction=5#Przyk.C5.82ad_.E2.80.93_klasyfikacja_dokumentu

Sieć Bayesowska

- skierowany graf acykliczny, w którym wierzchołki reprezentują zdarzenia, a krawędzie związki przyczynowe pomiędzy tymi zdarzeniami. Jeśli od wierzchołka A prowadzi ścieżka do wierzchołka B to B jest potomkiem A
- niezależność danego zdarzenia od wszystkich innych, które nie są jego potomkami.

http://en.wikipedia.org/wiki/Bayesian_network



Inny przykład: http://en.wikipedia.org/wiki/Bayesian_network#Example

Niezależne, całościowe wykłady:

- <http://www.di.unipi.it/~bacciu/teaching/ML/lect1-bayesLearn-2014-hand.pdf>
- <https://class.coursera.org/machlearning-001/lecture> [Week Five: Statistical Learning]

Narzędzia:

- <http://scikit-learn.org> – biblioteka open source do Machine learningu dla języka Python

