

# Machine Learning – Systemy uczące się



Jerzy Stefanowski  
Instytut Informatyki  
Politechniki Poznańskiej

Fragmenty wykładu studentów Politechniki  
Poznańskiej Poznań 2020

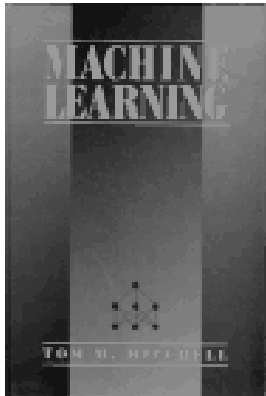
# Co będzie ?

- Wprowadzenie do problematyki maszynowego uczenia się
- Studium przypadku -- sterowanie pojazdem z wykorzystaniem sieci neuronowych
- Klasyfikacja podstawowych metod
- Reprezentacja przykładów uczących oraz wiedzy
- Algorytm eliminacji kandydatów
- Podsumowanie

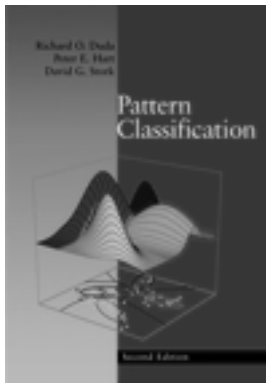
# Co to jest maszynowe uczenie się ?

- **Maszynowe uczenie się** (ang. *Machine Learning*) jest analizą procesów uczenia się oraz tworzeniem systemów, które doskonalą swoje działanie na podstawie doświadczeń z przeszłości.
- Wykład dotyczy dwóch grup metod :
  - ✦ symbolicznego/indukcyjnego uczenia się na przykładach (ang. *symbolic/inductive learning*)
  - ✦ sztucznych sieci neuronowych (ang. *artificial neural networks*)
- Inspiracje: informatyka, statystyka, rozpoznawanie struktur (ang. *pattern recognition*), psychologia i kognitywistyka

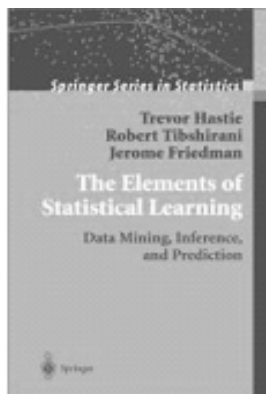
# Literatura - przykłady



**Mitchell (1989). Machine Learning.**  
**<http://www.cs.cmu.edu/~tom/mlbook.html>**

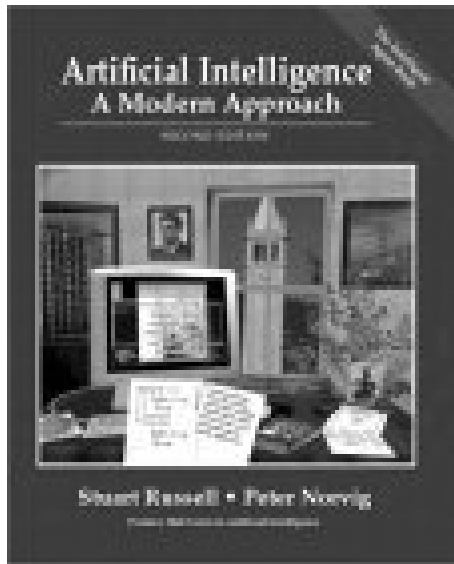


**Duda, Hart, & Stork (2000). Pattern Classification.**  
**<http://rii.ricoh.com/~stork/DHS.html>**



**Hastie, Tibshirani, & Friedman (2001). The Elements of Statistical Learning.** **<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>**

# Literature (cd.)



**Russell & Norvig (2004). Artificial Intelligence.**  
**<http://aima.cs.berkeley.edu/>**

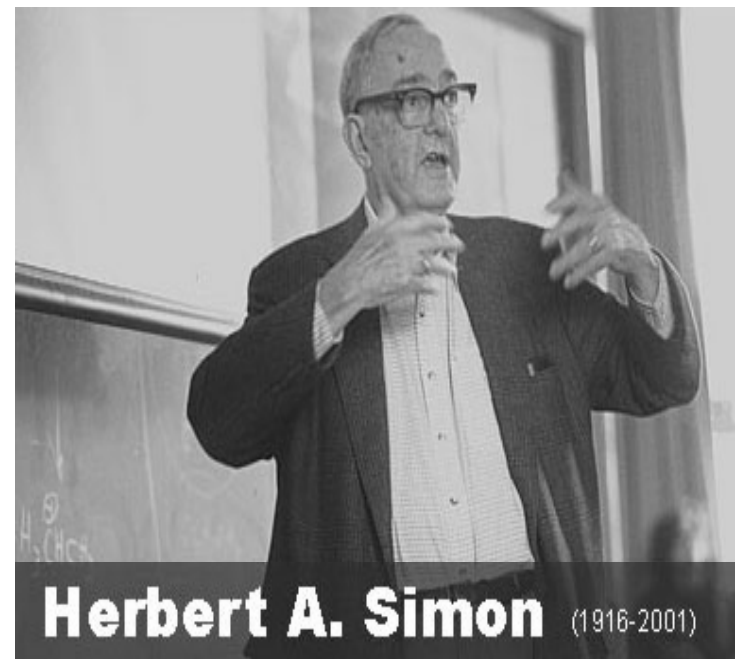
*A must-have for AI Master students. Contains good chapters on various machine learning techniques*

# Uczenie maszyn

- Machine Learning (ML) – część sztucznej inteligencji (AI) lub inteligencji obliczeniowej (Computational Intelligence - CI)
- Systemy adaptujące się, zmieniające swoje wewnętrzne parametry tak, aby rozpoznać charakter danych
- ML umożliwia pozyskiwanie wiedzy na podstawie analizy zachowań ekspertów lub danych doświadczalnych

# Systemy uczące się

Uczenie – zmiany w systemie adaptującym się pozwalające mu w przyszłości działać bardziej efektywnie na takich samych zadaniach lub zadaniach o podobnych charakterze (Simon 1983)



Nagroda Nobla w  
Ekonomii 1978

# Inne definicje

## Definition (#1/2)

A learning system uses  
sample data

to generate an updated basis  
for improved [performance]

on subsequent data  
from the same source

and expresses the new basis  
in intelligible symbolic form



- **Donald Michie (1991)**

- *Computer Journal*



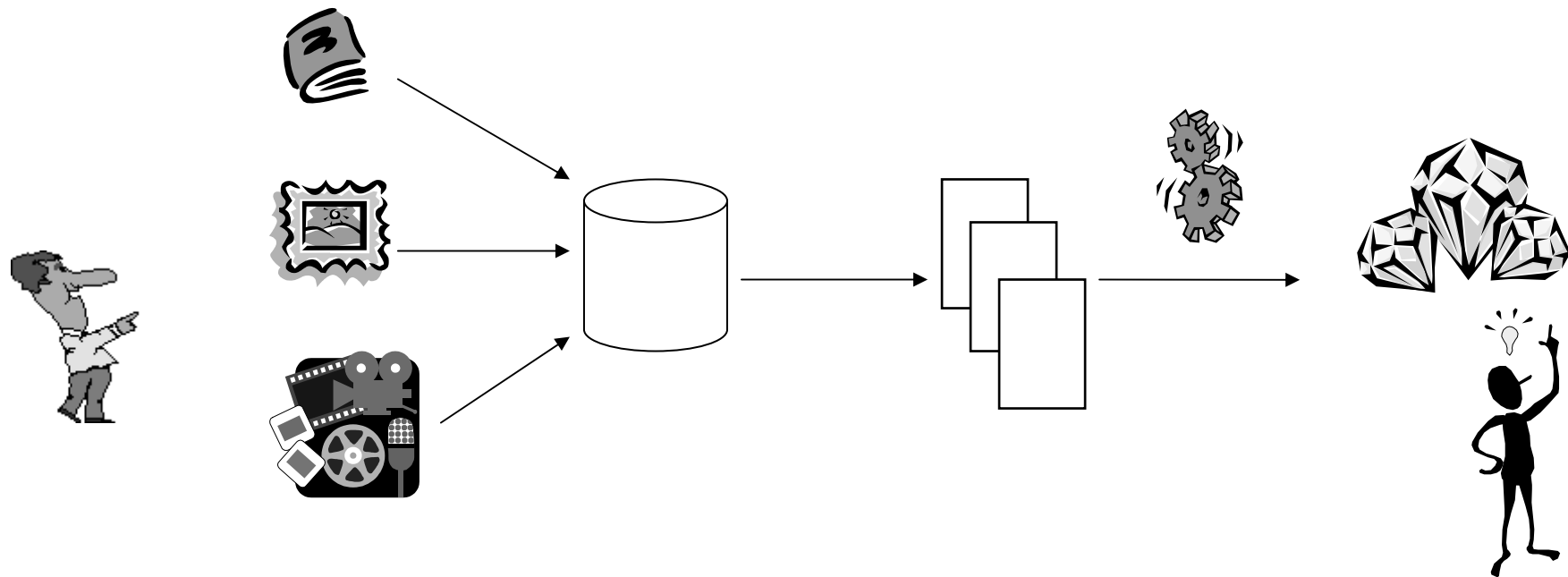
# Definicja R. Michalskiego

- "Uczenie się to konstruowanie i zmiana reprezentacji doświadczanych faktów. W ocenie konstruowanych reprezentacji bierze się pod uwagę:
  1. **wiarygodność** - określa stopień w jakim reprezentacja odpowiada rzeczywistości,
  2. **efektywność** - charakteryzuje przydatność reprezentacji do osiągnięcia danego celu.
  3. **poziom abstrakcji** - odpowiada zakresowi szczegółowości i precyzji pojęć używanych w reprezentacji; określa on tzw. moc opisową reprezentacji.
- Reprezentacja jest rozumiana jako np. opisy symboliczne, algorytmy, modele symulacyjne, plany obrazu."

# Systemy uczące się

Dane często w postaci przykładów

Eksperti opisując sytuację podają przykłady



# Definicja problemu uczenia

**Uczenie = doskonalenie działania dla pewnych zadań na podstawie doświadczenia**

- ✱ doskonalenie działania dla zadań  $T$
- ✱ z względu na miarę oceny  $P$
- ✱ oparte na doświadczeniu  $E$

## **Przykład 1: Gra w szachy**

$T$ : nauczyć się rozgrywać końcówki szachowe

$P$ : % wygranych gier

$E$ : zapis rozgrywek bardzo dobrych graczy

## **Przykład 2: Rozpoznawanie adresu (kodu) pocztowego**

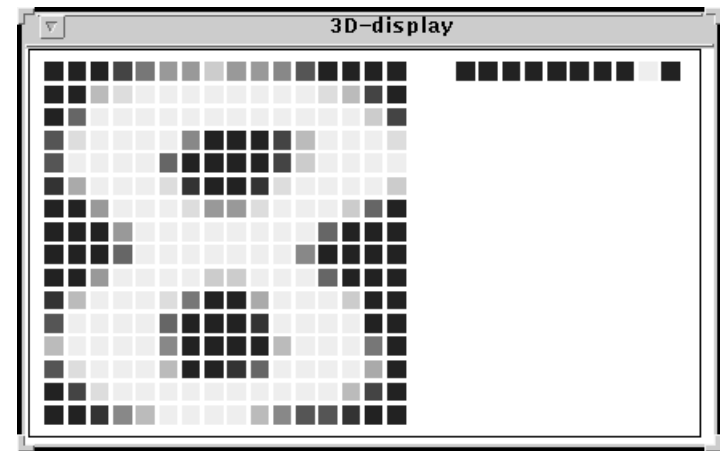
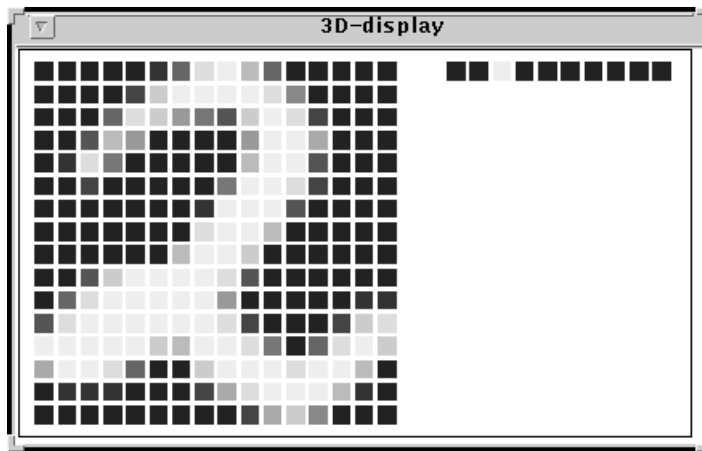
$T$ : rozpoznanie adresu (kodu) odbiorcy na liście

$P$ : % sortowanych listów poprawnie przypisanych do miasta odbiorcy

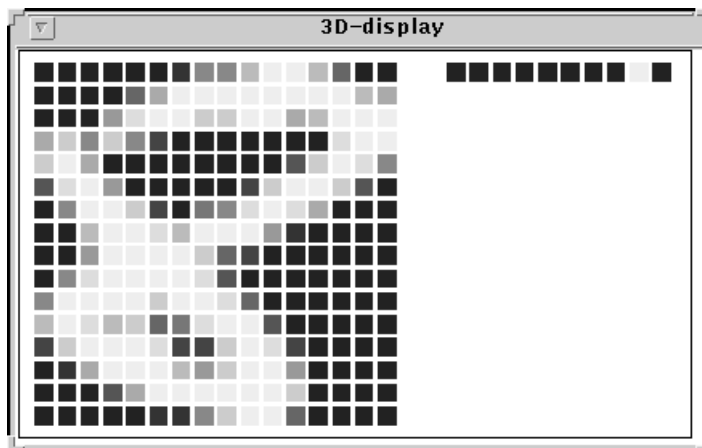
$E$ : odpowiednia liczba przykładów ręcznie napisanych adresów ze znanym adresem przeznaczenia

# Rozpoznawanie pisma - SSN

Uczenie:



Rozpoznawanie:



# Dlaczego systemy uczące się

- Zadania eksploracji i analizy danych, gdzie duże rozmiary zbiorów danych uniemożliwiają ich analizę w sposób nieautomatyczny (np. ekonomiczne lub medyczne bazy danych)
- Środowiska, gdzie system musi się dynamicznie dostosowywać do zmieniających się warunków (np. systemy sterowania)
- Problemy, które są złożone, trudne do opisu i często nie posiadają wystarczających modeli teoretycznych albo ich uzyskanie jest bardzo kosztowne lub mało wiarygodne
- Rozwój zastosowań praktycznych

# Projekt ALVINN – sterowanie pojazdem z wykorzystaniem sieci neuronowych

*ALVINN (Autonomous Land Vehicle In a Neural Network)*

system uczący się sterowania pojazdu na podstawie obserwacji zachowania kierowcy

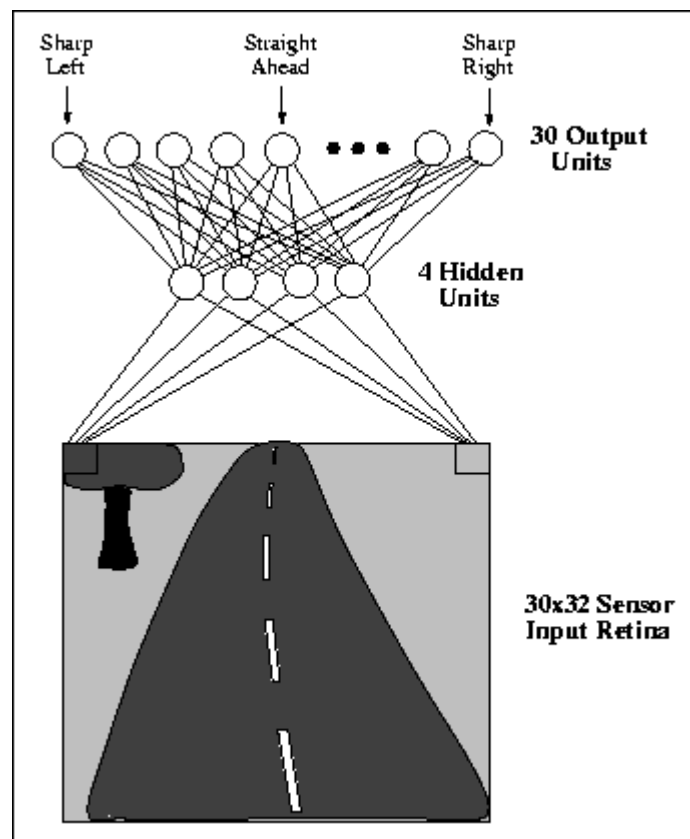
(Pomerleau & Jochem)



# ALVINN

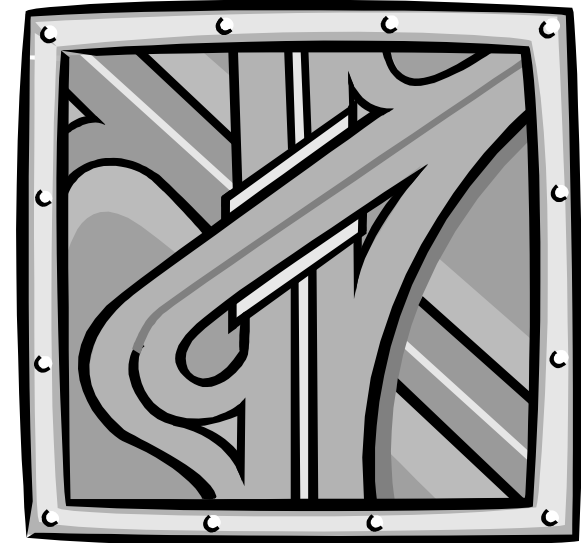


Architektura systemu:  
wielowarstwowa sieć  
neuronowa uczona  
algorytmem wstecznej  
propagacji błędów



# ALVINN - uczenie

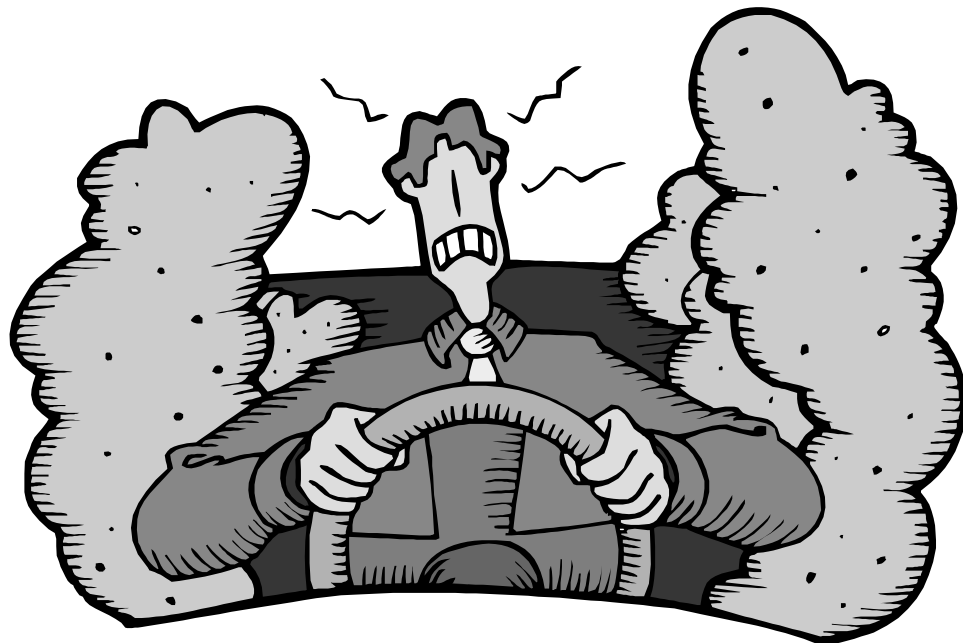
- Prezentacja zapisu video z kierowania pojazdem skojarzonego z informacją o sposobie kierowania
- ALVINN nauczył się kierować pojazdem na kilku rodzajach dróg; w tym z prędkością 70 mph na autostradzie k. Pittsburga
- Eksperymenty z bieżącą obserwacją zachowań kierowcy i kontynuowaniem jego stylu jazdy





# ALVINN ;-)

- Sieć neuronowa „śledzi” zachowanie kierowcy, więc...
- ... uważaj kto jest Twoim nauczycielem !



# Realizacja procesu uczenia się

- Dobrze zdefiniowane zadanie.
- Wybrane miary oceny (jedno, dobrze zdefiniowane kryterium).
- Co jest doświadczeniem (przykładami)?
- Czego należy się nauczyć?
- Jak reprezentować przykłady uczące?
- Jak reprezentować hipotezy/wiedzę?
- Jaki algorytm uczenia należy wybrać ?

# Learning (wiedza a umiejętności)

- Rote learning (implementacja wiedzy / uczenie się na pamięć)
  - ✱ Learning foreign words (and their translations)
  - ✱ Learning telephone numbers, bank accounts,  
..
- Inductive learning (uogólnienie)
  - ✱ Learning from examples
  - ✱ Learning by observing and discovering

# Early work on automatic learning

- Rosenblatt (1957): Perceptrons
- Samuel (1963): Rote learning
- Winston (1970): Concept learning
- Mitchell (1977): Version spaces

(See Russel and Norvig for more details)

# Klasyfikacja metod maszynowego uczenia się

- W ogólności system zmieniać się może w dwojaki sposób:
  - ✱ system nabywa "nową wiedzę" z zewnętrznych źródeł,
  - ✱ system modyfikuje się samodzielnie poprzez intensywne wykorzystywanie posiadanej już wiedzy w sposób bardziej efektywny.
- **Wnioskowanie Indukcyjne** opiera się na zbiorze faktów (obserwacji) oraz ewentualnym zbiorze hipotez a priori dotyczących tych faktów, a w wyniku daje uogólnienie wyjaśniające te fakty.
- **Dedukcja** - wyprowadzanie wniosków, które są logiczną konsekwencją posiadanej informacji.

# Symboliczne metody ML w odkrywaniu wiedzy z danych

- Wiele problemów sprowadza się do tzw. uczenia nadzorowanego
- Nadzór: przykłady uczące (obiekty) przydzielone są do pewnych znanych wstępnie kategorii (klas decyzyjnych)
- Cel: nauczyć się zasad przydziału obiektów ze zbioru uczącego do klas na podstawie opisu obiektów - problem klasyfikacji

# Uczenie z nadzorem

Uczenie z nadzorem: podział na znane klasy, przekazywanie znanej wiedzy, heteroasocjacja - kojarzenie obiektów i ich własności.

Opis obiektów

System adaptujący się

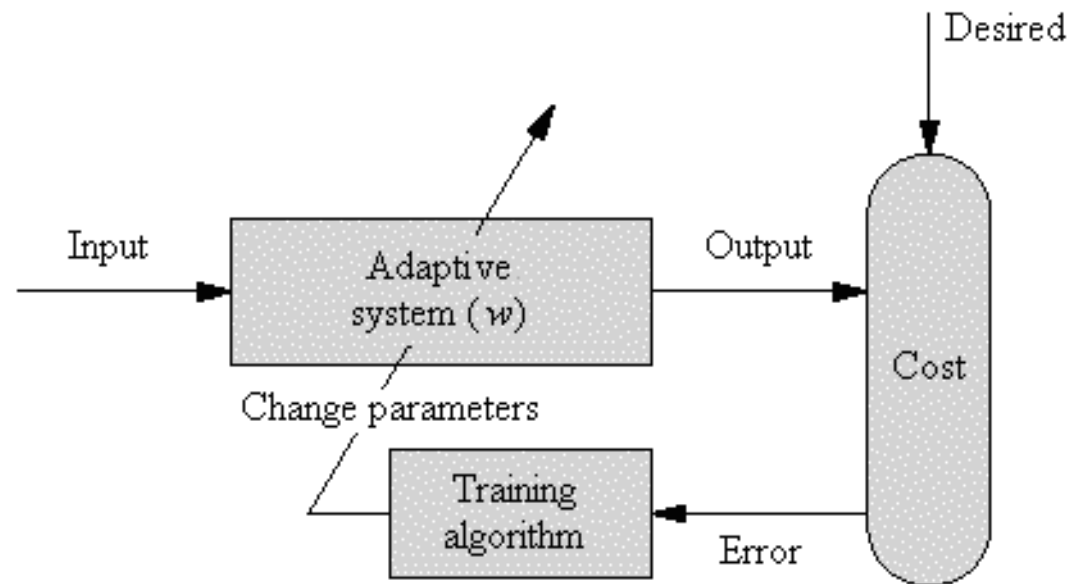
Wyniki działania

Pożądane odpowiedzi

Funkcja kosztu

Algorytm uczący

Cel uczenia: generalizacja



# Zadanie uczenia nadzorowanego

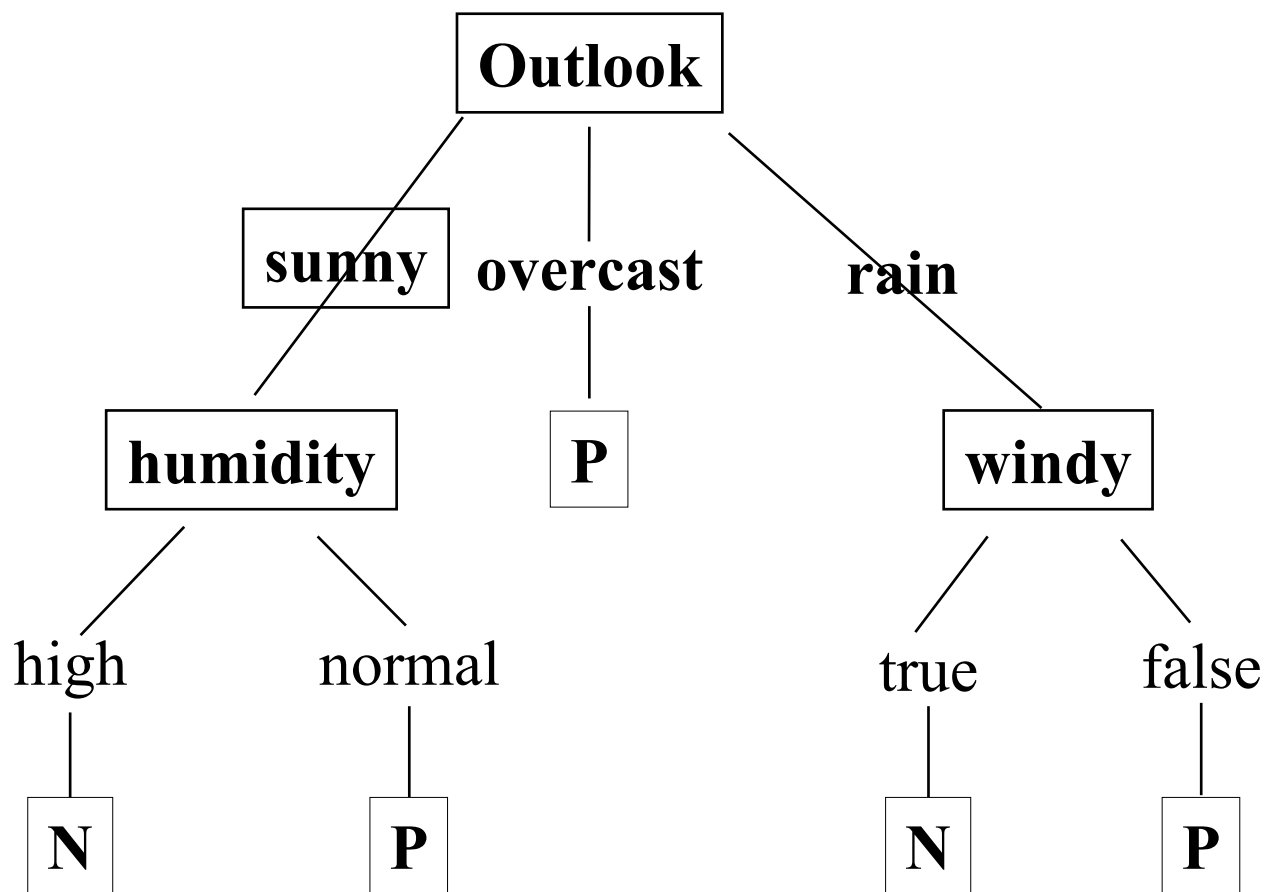


Przykład	$x_1$	$x_2$	$x_3$	$x_4$	$y$
0	0	1	1	0	0
1	0	0	0	0	0
2	0	0	1	1	1
3	1	0	0	1	1
4	0	1	1	0	0
5	1	1	0	0	0
6	0	1	0	1	0

- Nauczyciel ( $c$  - target function)  $\rightarrow$  funkcja  $c(x)$  nieznana uczniowi; poszukuje hipotezy  $h$ , która ją najlepiej przybliża.

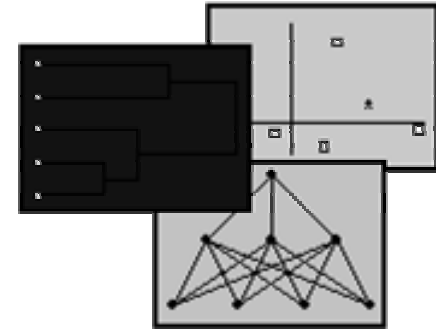


# Drzewo decyzyjne



# Klasy metod

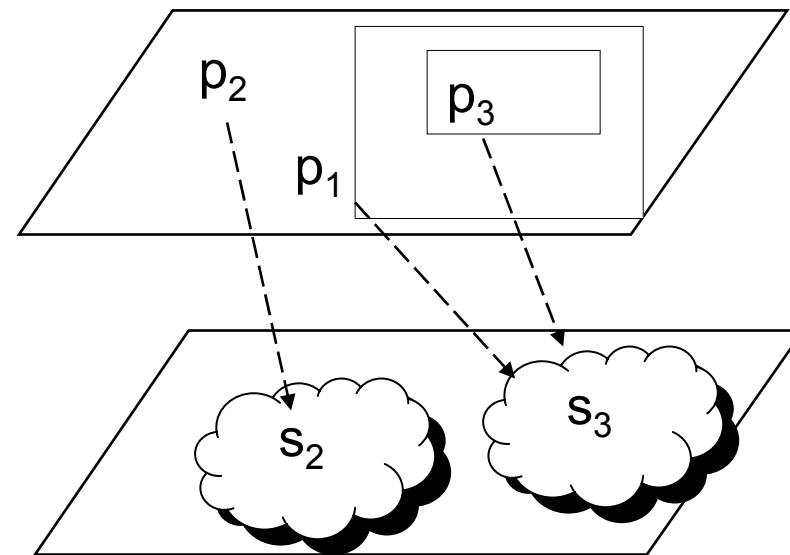
- Często stosowane:
  - ✱ drzewa decyzyjne/ reguły decyzyjne
  - ✱ uczenie się z przypadków (IBL, KNN, CBR..)
  - ✱ klasyfikatory probabilistyczne
  - ✱ sieci neuronowe
  - ✱ zbiory i logika rozmyta
- Inne metody:
  - ✱ algorytmy genetyczne
  - ✱ induktywne programowanie logiczne (ILP)
  - ✱ zbiory przybliżone
  - ✱ ...



# Uczenie bez nadzoru

Wykrywanie nieznanymi klas, uczenie spontaniczne,  
odkrywanie ciekawych struktur w przestrzeni danych,  
korelacja zachowań systemu ze zmianą tych struktur.  
Dominuje w okresie niemowlęcym.

Opis obiektów  $p_i$   
System adaptujący się  
Tworzenie klas  
Miary podobieństwa  
Funkcja jakości  
grupowania  
Algorytm uczący



# Uczenie zachowań

Uczenie z krytykiem lub „z wzmocnieniem” pożądaných zachowań po dłuższym okresie.

Optymalizacja zysków na dłuższą metę.

Gry z przeciwnikiem: przegrana lub wygrana na końcu partii.

Uczenie dojrzałe (nabieranie „mądrości”), behawioralne.

Opis obiektów/stanów  $p_i$

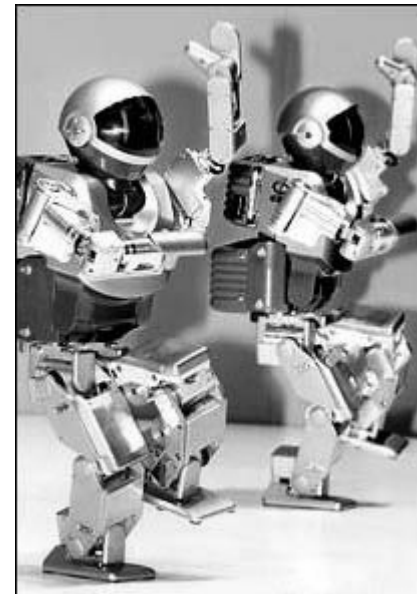
Opis reguł działania

System adaptujący się

Tworzenie strategii/planów

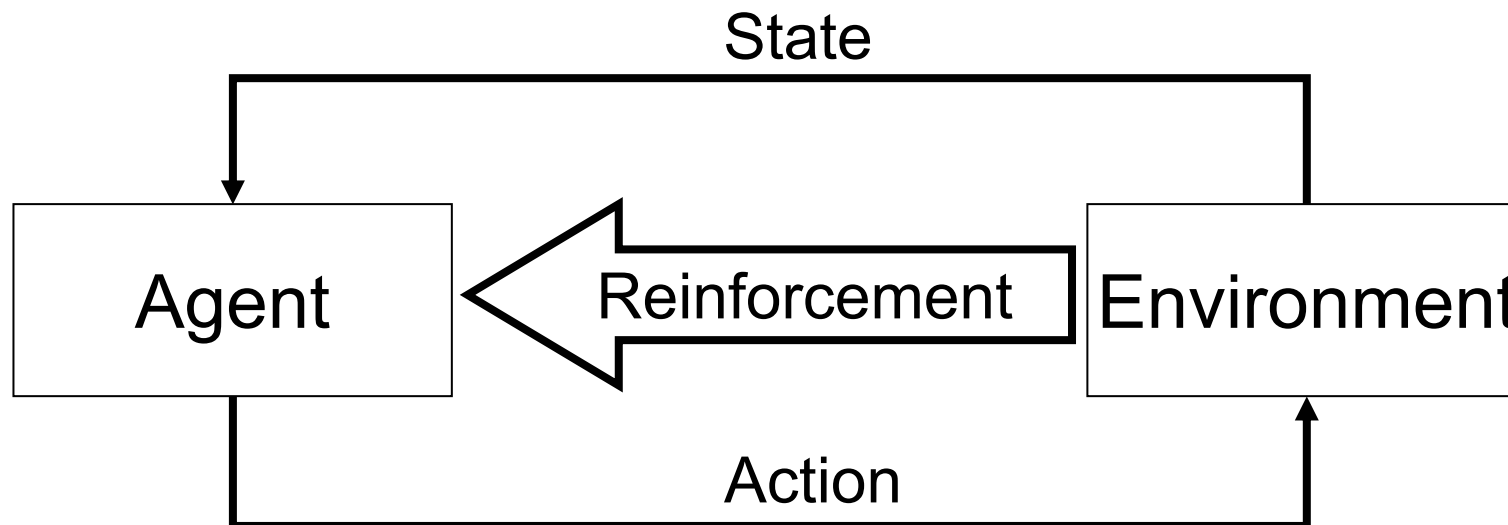
Miara sukcesu

Algorytm uczący



# Uczenie się ze wzmocnieniem

- Programming agents by reward and punishment
- No need to specifying HOW the task is to be achieved
- Agent learns behavior through trial-and-error interactions with a dynamic environment



# Draft RL Algorithm

For each time step  $t$  do:

1. Perceive current state  $\mathbf{x}_t$ ,
2. Choose action  $\mathbf{a}_t$  to be performed,
3. Perform selected action  $\mathbf{a}_t$ ,
4. Watch for current state  $\mathbf{x}_{t+1}$  and reinforcement  $r_t$ ,
5. Learn basing on new experience:  
 $\langle \mathbf{x}_t, \mathbf{a}_t, r_t, \mathbf{x}_{t+1} \rangle$ .

					£
☺					€

## **Dimensions: Constraints**

(on selecting a learning algorithm)

- o Task/objective
  - Learning task
  - Performance task
- o Availability of background knowledge
  - Encoded
  - Interactive
- o Availability of data
  - Incremental vs. batch
  - Passive vs. active
- o Characteristics of data
  - Static vs. drifting
  - Propositional or first-order

## **Learning Tasks**

Supervised learning (from examples)

- Knowledge-poor
- Knowledge-intensive
  - *Improve predictive accuracy*
  - *Improve explanatory capability*
  - *Improve efficiency (speed, space)*

Problem solving

- Learning operator application conditions
- Learning to solve additional problems

Unsupervised learning (clustering)

- No teacher available; unlabeled data

Scientific discovery

- Often involves aspects of both supervised & unsupervised learning

- Parę uwag o klasyfikacji metod

# Uczenie się z wykorzystaniem wiedzy

## Knowledge-Intensive Supervised Learning

Given:

1. A set of training instances
2. A hypothesis of the target concept
3. Background knowledge (BK)

Induce:

A modified hypothesis (concept description) that is consistent with the domain theory & training instances

Typical performance measures:

- Predictive accuracy (on test set)
- Comprehensibility
- Speed (training and testing)
- Storage requirements
- Hypothesis language
- Instance representation language

Infinite number of approaches. Three are:

- Explanation-based learning
  - Learning macros, etc
- Inductive logic programming (ILP)
- Case-based learning

## Explanation-Based Learning (EBL)

General idea: "Speedup learning"

- Explain how a given instance instantiates the BK
- Explanation = Proof
- Induced hypotheses must be consistent with both training instances and BK
- Yields macros (i.e., caching, chunking, compiling)

Definition:

*Explanation-based learning* is the process of forming generalizations by constructing explanations.

Strengths:

- Can work with very few instances (none?)
- Allows for explicit representation of BK
- Exploits BK, usually in the form of a set of rules

Limitations:

- Requires background knowledge
- Ongoing work to handle imperfect BK
- Utility issues: When does learning/caching pay off?



# Terminologia związana z indukcyjnym uczeniem się

- Dziedzina  $X$
- Przykład uczący  $x \in X$
- Atrybut  $a_i$
- Klasa, pojęcie docelowe  $C_i \subseteq X$   $c: X \rightarrow \{0,1\}$
- $C_i = \{x \in X : c(x)=i\}$
- Przykład etykietowany dla uczenia z nadzorem  $\langle x, c(x) \rangle$
- Przykład pozytywny dla pojęcia  $c$  - taki  $x \in X$  że  $c(x)=1$
- Przykład negatywny dla pojęcia  $c$  - taki  $x \in X$  że  $c(x)=0$
- Hipoteza  $h$  - funkcja  $f: X \rightarrow \{0,1\}$
- Przestrzeń hipotez  $H$
- Zbiór uczący
- Błąd stosowania hipotezy  $h$  dla zbioru treningowego  $D$
- Indukcyjne ukierunkowanie (*inductive bias*)

# Reprezentacja przykładów uczących

- **Przykłady** – zbiór obiektów, których dotyczy nabywana wiedza; mogą to być przedmioty, osoby, obserwacje, ...
- **Atrybuty** to pewne cechy charakteryzujące właściwości rozważanych obiektów.
- Zbiór przykładów (obiektów)  $U$  jest opisany za pomocą skończonego, niepustego zbioru atrybutów  $A = \{a_1, a_2, \dots, a_m\}$  i reprezentowany jest w postaci **tabeli informacyjnej**  $IT = (U, A)$

# Przykład dydaktyczny

- **Tabela 1.** Zbiór przykładów opisujących samochody za pomocą zbioru 4 atrybutów {*Typ samochodu, Cena, Prędkość, Zużycie paliwa*}; Pierwsza kolumna zawiera numer identyfikujący samochód

<i>lp.</i>	<i>Typ</i>	<i>Cena</i>	<i>Prędkość</i>	<i>Zużycie</i>
1	mały	akceptowalna	przeciętny	małe
2	duży	akceptowalna	szybki	średnie
3	kompakt	drogi	szybki	średnie
4	duży	drogi	szybki	wysokie
5	mały	drogi	wolny	średnie
6	kompakt	akceptowalna	przeciętny	średnie
7	kompakt	drogi	wolny	średnie
8	duży	drogi	przeciętny	wysokie
9	duży	akceptowalna	przeciętny	średnie

# Uczenie się pojęć

- **Pojęcie** (ang. *concept*) – ogólne określenie oznaczające zbiór obiektów posiadających wspólne właściwości, które odróżniają je od innych pojęć.
- Pojęcie to jedna z postaci wiedzy o świecie, używana do opisywania systemów interpretowania obserwacji oraz abstrakcyjnych idei.
- Pojęcia – pozwalają na klasyfikowanie pewnych obiektów na zbiory (kategorie / klasy).
- Pojedyncze pojęcie wyznacza podział zbioru obiektów na **przykłady pozytywne** oraz **przykłady negatywne** (kontrprzykłady) tego pojęcia.
- Pojęcia wielokrotne – klasyfikacja

# Przykład ilustracyjny opisu samochodów

- Rozważane pojęcie - „*samochód, którym chciałbym jeździć*”
- Do przykładów tego pojęcia zaliczmy samochody {2,3,6,9}.
- Kontrprzykładami, czyli przykładami negatywnymi tego pojęcia, są samochody {1,4,5,7,8}.
- Wspólne właściwości definiujące pojęcie:  
 $(\text{zużycie}=\text{średni}) \wedge ((\text{cena}=\text{akceptowalna}) \vee (\text{prędkość}=\text{szybki}))$
- Definicja znana „nauczycielowi” – niedostępna jawnie systemowi uczącemu się.
- **Cel:** system w trakcie procesu uczenia się ma odkryć / znaleźć nieznaną mu definicję pojęcia na podstawie dostarczonych przykładów problemów kontrprzykładów tego pojęcia

- Przestrzeń wersji (CEA) – Algorytm eliminacji kandydatów

# Przykład pojęcia „*EnjoySport*”

- Specyfikacja problemu
  - \* Przykłady  $X$  opisane 6 atrybutami: Sky, Temp, Humidity, Wind, Water, Forecast
- Pojęcia binarne  $c$ :  $\text{EnjoySport} : X \rightarrow \{0,1\}$  // no, yes
- Hipoteza ucznia: Koniunkcja wyrażeń elementarnych
  - \* Sky            AirTemp   Humidity   Wind        Water        Forecast  
    <Sunny ?            ?            Strong ?            Same>
- Ograniczenia: konkretna wartość ( $\text{Water} = \text{Warm}$ ); don't care (" $\text{Water} = ?$ ") oraz wartość niedozwolona „ $\emptyset$ ".

Example	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy Sport
0	Sunny	Warm	Normal	Strong	Warm	Same	Yes
1	Sunny	Warm	High	Strong	Warm	Same	Yes
2	Rainy	Cold	High	Strong	Warm	Change	No
3	Sunny	Warm	High	Strong	Cool	Change	Yes

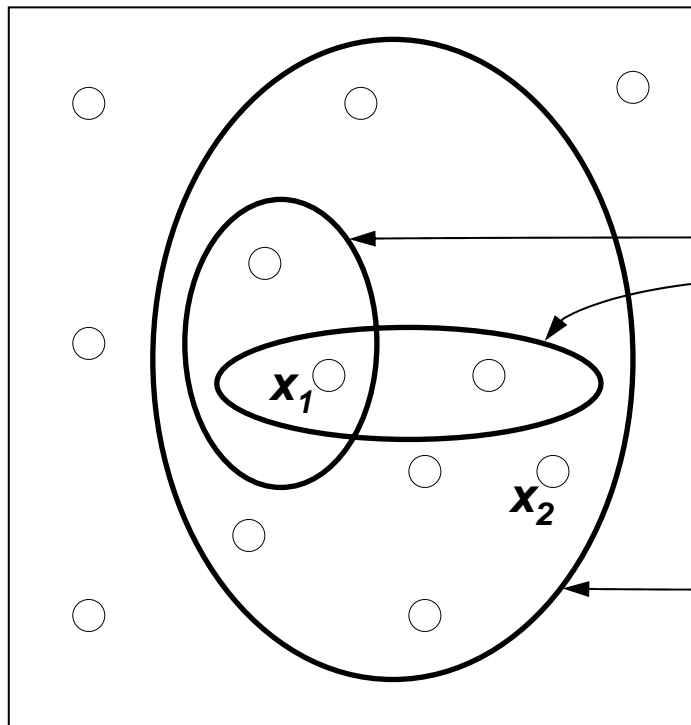
# Concept Learning and Version Spaces

- Concept Learning as Search through  $H$ 
  - ✱ Hypothesis space  $H$  as a state space
  - ✱ Learning: finding the correct hypothesis
- General-to-Specific Ordering over  $H$ 
  - ✱ Partially-ordered set: Less-Specific-Than (More-General-Than) relation
  - ✱ Upper and lower bounds in  $H$
- Version Space Candidate Elimination Algorithm
  - ✱  $S$  and  $G$  boundaries characterize learner's uncertainty
  - ✱ Version space can be used to make predictions over unseen cases
- Learner Can Generate Useful Queries
- Inne zagadnienia?



# Przykłady, hipotezy i ich uporządkowanie *Less-Specific-Than*

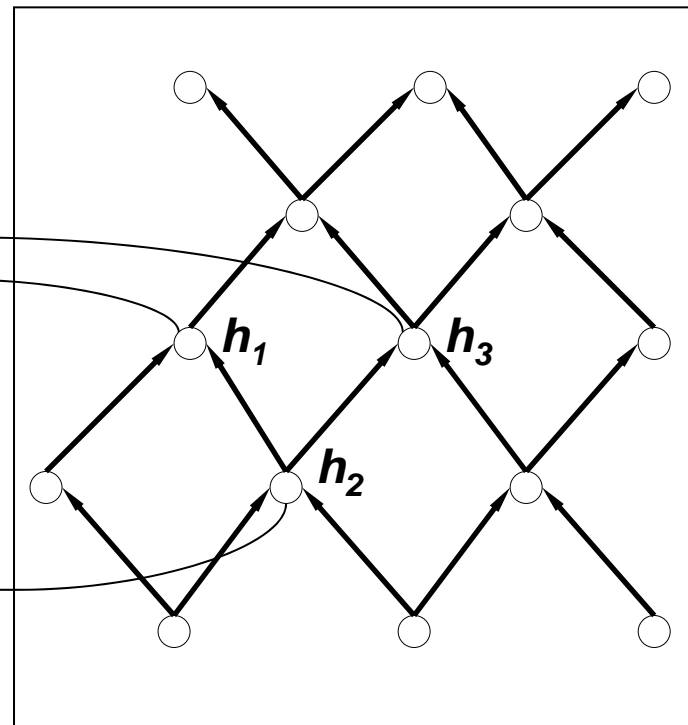
Przykłady  $X$



$x_1 = \langle \text{Sunny, Warm, High, Strong, Cool, Same} \rangle$   
 $x_2 = \langle \text{Sunny, Warm, High, Light, Warm, Same} \rangle$

$\leq_p \equiv \text{Less-Specific-Than} \equiv \text{More-General-Than}$

Hipotezy  $H$



$h_1 = \langle \text{Sunny, ?, ?, Strong, ?, ?} \rangle$   
 $h_2 = \langle \text{Sunny, ?, ?, ?, ?, ?} \rangle$   
 $h_3 = \langle \text{Sunny, ?, ?, ?, Cool, ?} \rangle$

$h_2 \leq_p h_1$   
 $h_2 \leq_p h_3$

Specific

General

# Version Spaces – przestrzeń wersji

- Definicja: Spójna hipoteza
  - ✱ Hipoteza  $h$  jest spójna ze zbiorem uczącym  $D$  pewnego pojęcia  $c$  iff  $h(x) = c(x)$  dla każdego przykładu uczącego  $\langle x, c(x) \rangle$  w  $D$ .
  - ✱  $Consistent(h, D) \equiv \forall \langle x, c(x) \rangle \in D . h(x) = c(x)$
- Definicja: Przestrzeń wersji
  - ✱ Przestrzeń wersji  $VS_{H,D}$ , ze względu na przestrzeń hipotez  $H$  i zbiór uczący  $D$ , jest podzbiorem przestrzeni hipotez  $H$  spójnych ze wszystkim przykładami uczącymi  $D$ .
  - ✱  $VS_{H,D} \equiv \{ h \in H \mid Consistent(h, D) \}$
- Def: General Boundary
  - ✱ General boundary  $G$  of version space  $VS_{H,D}$ : set of most general members
  - ✱ Most general  $\equiv$  *minimal* elements of  $VS_{H,D} \equiv$  “set of necessary conditions”
- Def: Specific Boundary
  - ✱ Specific boundary  $S$  of version space  $VS_{H,D}$ : set of most specific members
  - ✱ Most specific  $\equiv$  *maximal* elements of  $VS_{H,D} \equiv$  “set of sufficient conditions”

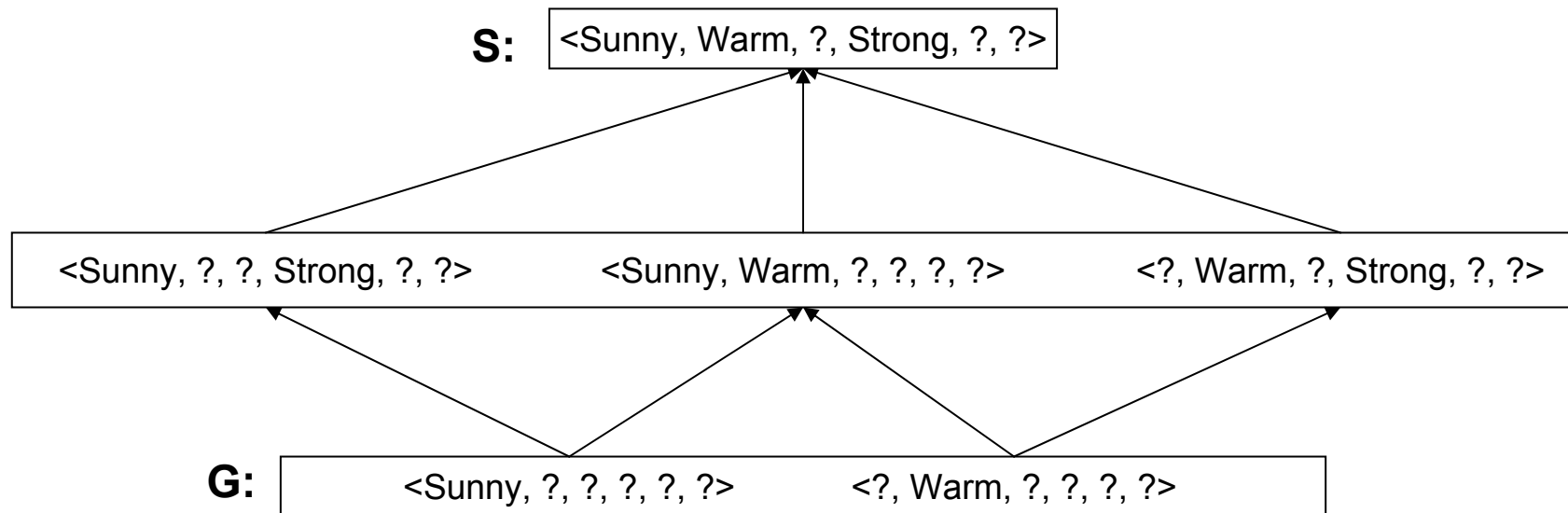
# The *List-Then-Eliminate* Algorithm

1. Initialization: *VersionSpace*  $\leftarrow$  a list containing every hypothesis in  $H$

2. For each training example  $\langle x, c(x) \rangle$

Remove from *VersionSpace* any hypothesis  $h$  for which  $h(x) \neq c(x)$

3. Output the list of hypotheses in *VersionSpace*

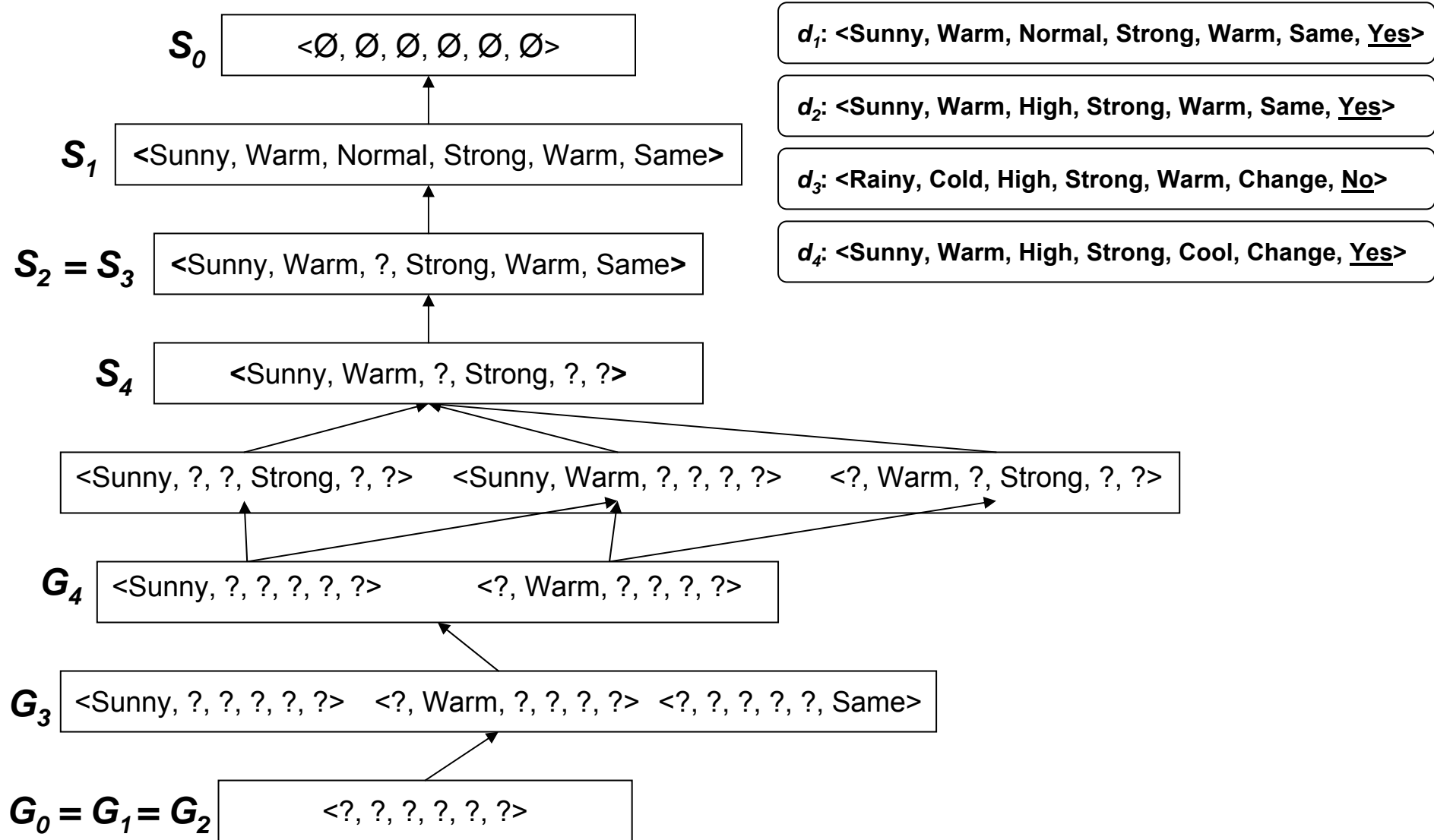


**Example Version Space**

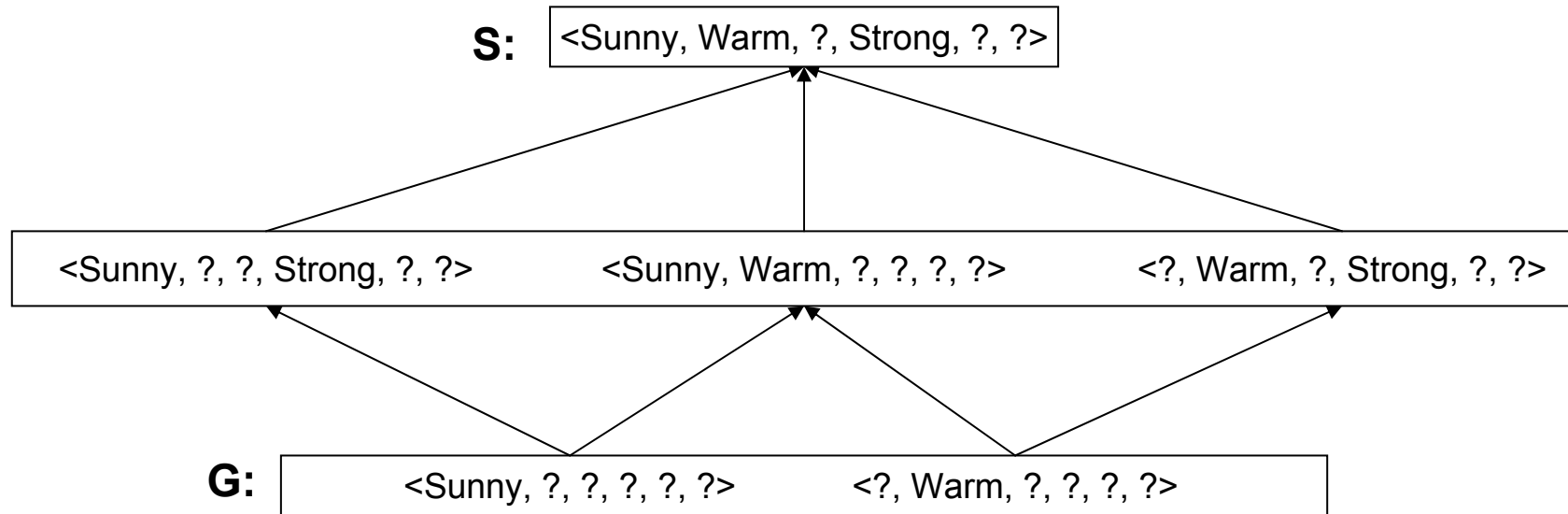
# Algorytm CEA

- inicjalizuj  $S$  i  $G$  odpowiednio jako zbiory maksymalnie szczegółowych i maksymalnie ogólnych hipotez w  $H$ ;
- dla każdego przykładu  $x$  wykonaj
  - ✱ jeśli  $x$  jest przykładem negatywnym ( $c(x)=0$ ) to
    - pozostaw w  $S$  tylko hipotezy, które nie pokrywają  $x$ ;
    - każdą hipotezę  $g$  w  $G$ , która pokrywa  $x$ , zastąp hipotezami bardziej szczegółowymi które nie pokrywają  $x$  tak, aby każda z nich pozostała tak samo lub bardziej ogólna niż pewna hipoteza z  $S$ ;
    - usuń z  $G$  każdą hipotezę bardziej szczegółową niż jakakolwiek inna hipoteza z  $G$ ;
  - ✱ jeśli  $x$  jest przykładem pozytywnym ( $c(x)=1$ ) to
    - pozostaw w  $G$  tylko hipotezy, które pokrywają  $x$ ;
    - każdą hipotezę  $s$  w  $S$ , która nie pokrywa  $x$ , zastąp hipotezami bardziej ogólnymi które pokrywają  $x$  tak, aby każda z nich pozostała tak samo lub bardziej szczegółowa niż pewna hipoteza z  $G$ ;
    - usuń z  $S$  każdą hipotezę bardziej ogólną niż jakakolwiek inna hipoteza z  $S$ .

# Przykład ilustracyjny



# Jaki powinien być kolejny przykład?



- Jakie pytanie może zadać nauczycielowi system uczący się?
- Jak sklasyfikować następujące przykłady?
  - <Sunny, Warm, Normal, Strong, Cool, Change>
  - <Rainy, Cold, Normal, Light, Warm, Same>
  - <Sunny, Warm, Normal, Light, Warm, Same>

# Wybrane zastosowania ML

- Liczne zastosowania medyczne (diagnozowanie chorób, dobór terapii, analiza danych laboratoryjnych i diagnostycznych), farmacja, biologia, genetyka (DNA)
- Dużo zastosowań w przemyśle maszynowym, chemicznym, telekomunikacyjnym i finansach
- Systemy sterowania, automatyka, robotyka
- Eksploracja baz danych, e-commerce, Web-mining
- Zastosowania naukowe, np. katalogowanie obiektów astronomicznych, odkrywanie praw
- Analiza i rozpoznawanie obrazów, mowy i pisma

# Systemy uczące się

## Podsumowanie

- Aktualny stan
  - ✱ wiele różnych metod
  - ✱ stosowane do dobrze zdefiniowanych baz danych
  - ✱ większość zastosowań dotyczy problemów uczenia nadzorowanego
  - ✱ „proste” strategie uczenia nie wykorzystujące wiedzy „wyższego poziomu”



# Systemy uczące się - Podsumowanie

## Przyszłość ?

- ✱ odkrywanie wiedzy z różnego rodzaju danych
- ✱ złożone systemy uczące się
- ✱ statystyczna interpretacja teorii uczenia się.
- ✱ uczenie „dojrzałe”; intensywniejsze wykorzystanie wiedzy dziedzinowej
- ✱ zastosowania do złożonych problemów
- ✱ oprogramowanie z wbudowanymi zdolnościami samoadaptacji ?

# Gdzie szukać dalej ?

## Książki

P.Cichosz: Systemy uczące się.

K.Krawiec, J.Stefanowski: Uczenia maszynowe i sieci neuronowe.

W. Duch i in. (red.): Sieci neuronowe.

T.Mitchell: Machine learning.

## Internet

[www.cs.put.poznan.pl/jstefanowski/mlteaching.html](http://www.cs.put.poznan.pl/jstefanowski/mlteaching.html)

[www.phys.uni.torun.pl/~duch/ai-ml.html](http://www.phys.uni.torun.pl/~duch/ai-ml.html)

[www.aic.nrl.navy.mil/~aha/research/machine-learning.html](http://www.aic.nrl.navy.mil/~aha/research/machine-learning.html)

**Dziękuję za uwagę !**

e-mail: [Jerzy.Stefanowski@cs.put.poznan.pl](mailto:Jerzy.Stefanowski@cs.put.poznan.pl)