

Systemy Wspomagania Decyzji

Uczenie Maszynowe: wybór i ocena modelu. Drzewa Decyzyjne

Marcin Sydow

Plan

- reprezentacja wiedzy
- reguły decyzyjne
- drzewa decyzyjne i algorytm ID3
- złożoność modelu
- wybór i ocena modelu
- przetrenowanie i sposoby ominięcia
- walidacja krzyżowa

Podejścia do uczenia maszynowego

Systemy
Wspomaga-
nia
Decyzji


Marcin
Sydow

- sztuczne sieci neuronowe
- drzewa decyzyjne
- reguły decyzyjne
- support vector machines
- wiele innych...

Sieci neuronowe jako “black box”

Sieci neuronowe (zwłaszcza wielowarstwowe z regułą uczenia opartą na propagacji wstecznej) stanowią potężny i uniwersalny model uczenia maszynowego.

Jednak, mimo że taka sieć może “nauczyć się *teoretycznie* wszystkiego”¹ to wiedza w tym modelu reprezentowana jest w sposób zupełnie **nieczytelny** dla człowieka: w postaci wag połączeń i wartości progów poszczególnych neuronów. Taki model nazywamy “black box”, jest skuteczny ale nie nadaje się do analizy przez człowieka.

¹przy odpowiednio dużym zbiorze treningowym 

Reprezentacja wiedzy, cd

Istnieją modele uczenia maszynowego, gdzie automatycznie “nauczona” wiedza jest reprezentowana w sposób przejrzysty dla człowieka, np.:

- Reguły decyzyjne
- Drzewa decyzyjne

Przykład - diagnostyka okulistyczna

Systemy
Wspomaga-
nia
Decyzji

Marcin
Sydow

Wiedza w formie “surowej” tabeli decyzyjnej:

wiek	presc.	astygmatyzm	łzawienie	OKULARY
młody	myope	nie	niskie	zbędne
młody	myope	nie	normalne	lekkie
młody	myope	yes	niskie	zbędne
młody	myope	tak	normalne	mocne
młody	hypermetrope	nie	niskie	zbędne
młody	hypermetrope	nie	normalne	lekkie
młody	hypermetrope	tak	niskie	zbędne
młody	hypermetrope	tak	normalne	mocne
pre-presbyopic	myope	nie	niskie	zbędne
pre-presbyopic	myope	nie	normalne	lekkie
pre-presbyopic	myope	tak	niskie	zbędne
pre-presbyopic	myope	tak	normalne	mocne
pre-presbyopic	hypermetrope	nie	niskie	zbędne
pre-presbyopic	hypermetrope	nie	normalne	lekkie
pre-presbyopic	hypermetrope	tak	niskie	zbędne
pre-presbyopic	hypermetrope	tak	normalne	zbędne
presbyopic	myope	nie	niskie	zbędne
presbyopic	myope	nie	normalne	zbędne
presbyopic	myope	tak	niskie	zbędne
presbyopic	myope	tak	normalne	mocne
presbyopic	hypermetrope	nie	niskie	zbędne
presbyopic	hypermetrope	nie	normalne	lekkie
presbyopic	hypermetrope	tak	niskie	zbędne
presbyopic	hypermetrope	tak	normalne	zbędne

(Taka forma reprezentacji jest mało “skompresowana”: każdy wiersz to oddzielny przypadek.

Wiedza w formie reguł decyzyjnych

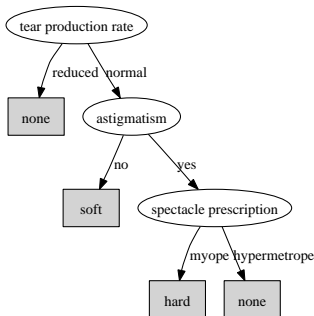
przykład kilku pierwszych **automatycznie wygenerowanych** reguł decyzyjnych (dla problemu diagnostyki okulistycznej):

- IF tear production rate = reduced THEN recommendation = NONE
- IF age = young AND astigmatic = no AND tear production rate = normal THEN recommendation = SOFT
- IF age = presbyopic AND astigmatic = no AND tear production rate = normal THEN recommendation = SOFT
- IF age = presbyopic AND spectacle prescription = myope AND astigmatic = no THEN recommendation = NONE

Reguły mogą stanowić dużo bardziej zwartą formę reprezentacji wiedzy niż tabela decyzyjna.

Przykładem algorytmu automatycznie generującego reguły decyzyjne jest **algorytm pokrywania** (ang. covering)

Wiedza w formie drzewa decyzyjnego



Dużo bardziej zwarta forma reprezentacji wiedzy (uwaga: te reguły pokrywają wszystkie poza 2 przypadki!)

Automatyczne generowanie drzew decyzyjnych:

Metoda ID3

W skrócie:

- 1 Wybieramy atrybut
- 2 tworzymy rozgałęzienia dla poszczególnych wartości atrybutu
- 3 powtarzamy 1 i 2 aż do momentu, gdy zostaną tylko elementy jednej kategorii we wszystkich rozgałęzieniach.

Uwaga: Im dłużej budujemy drzewo tym większe ryzyko “przetrenowania”.

Atrybut do podziału wybieramy ze względu na pewne kryterium - ogólnie dążymy do tego, żeby drzewo:

- jak najdokładniej klasyfikowało
- było jak najprostsze

(zauważmy: są to wzajemnie **przeciwstawne** postulaty)

Automatyczne generowanie drzew decyzyjnych - przykład

Przypomnijmy dane dotyczące pogody i pewnej gry:

outlook	temperature	humidity	windy	PLAY?
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

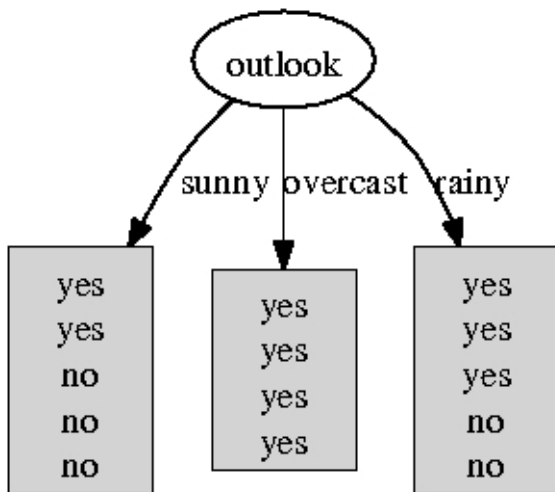
Budowanie drzewa decyzyjnego - Metoda ID3

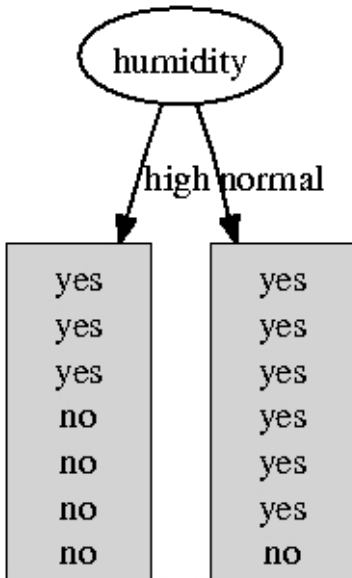
Systemy
Wspomaga-
nia
Decyzji

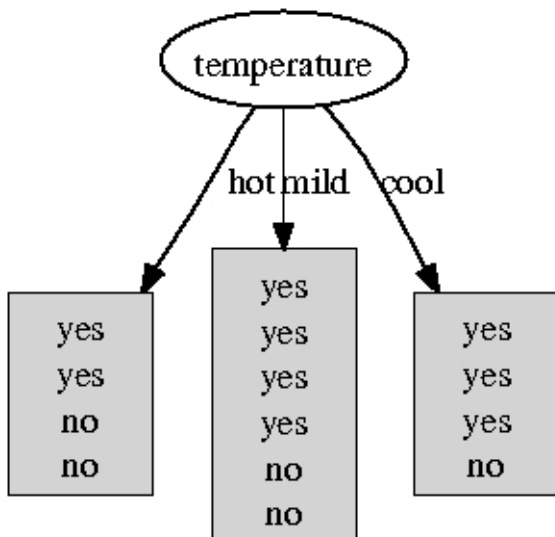
Marcin
Sydow

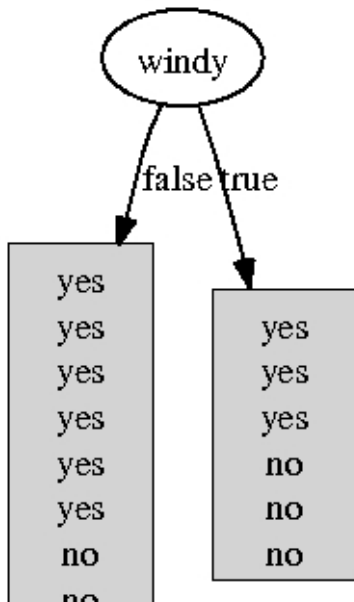
Mamy do wyboru 4 atrybuty: outlook, temperature, humidity oraz windy.

Czy widać który jest najlepszy?









Kryterium wyboru atrybutu do podziału

Metoda ID3

Intuicyjnie - atrybut jest tym lepszy im lepiej “rozdziela kategorie”.

Ściślej - z każdym możliwym podziałem można związać pewną miarę “jakości podziału” i wybrać ten atrybut, dla którego wartość tej miary jest najlepsza.

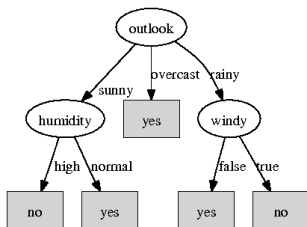
Na przykład, taką miarą jest “zysk informacyjny” (ang. **information gain**), pojęcie wprowadzone w teorii informacji i związane z pojęciem **entropii**, służące do mierzenia ilości informacji (rozwinętej w latach 40. XX. wieku m.in. przez wybitnego uczonego: **Claude Shannon'a**).

Wybieramy taki podział, że będzie trzeba **najmniej informacji**, żeby następnie wyspecyfikować kategorię.

Czy widać, który to atrybut?

Wynikowe Drzewo

Po kilku krokach, przy opisanej powyżej procedurze, otrzymujemy następujące wynikowe drzewo decyzyjne:



outlook	temp.	hum.	win.	?
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Udoskonalone Algorytmy Budowy Drzew

Najczęściej stosowanym w praktyce algorytmem budowy drzew decyzyjnych jest ogólnie dostępny **algorytm C4.5**.

Algorytm ten jest znacznym rozbudowaniem idei pokazanej przed chwilą (ID3).

Zawiera też znaczną ilość dodatkowych ulepszeń, do których należą m.in.: dostosowanie do atrybutów numerycznych, brakujących wartości, zanieczyszczonych danych oraz tzw. “oczyszczanie” drzewa (ang. **pruning**), które automatycznie upraszcza to drzewo i zapobiega **przetrenowaniu**.

Algorytm C4.5 ma też komercyjną (zastrzeżoną) wersję: **C5.2**, która jest jeszcze bardziej rozbudowana, i cechuje się nieznacznie wyższymi osiągnięciami.

Złożoność Modelu

Jest to bardzo ważne pojęcie. Im bardziej złożony (zawierający więcej “detali”) jest model, tym ma teoretycznie większe możliwości w odwzorowaniu niuansów uczonego pojęcia, ale niesie to też ryzyko tzw. **przetrenowania** czyli dostosowania się modelu “na sztywno” do danych trenujących, bez “uogólnienia” wiedzy na nieznane przypadki.

Złożoność nie powinna być więc za wysoka.

Zwykle możemy kontrolować złożoność modelu.

Na przykład:

- w sieciach neuronowych, złożoność modelu rośnie wraz z liczbą neuronów.
- w drzewach decyzyjnych: wraz z liczbą węzłów drzewa
- w regułach decyzyjnych: wraz z liczbą reguł

Model powinien być jak najprostszy.

Przykłady zbyt złożonych modeli

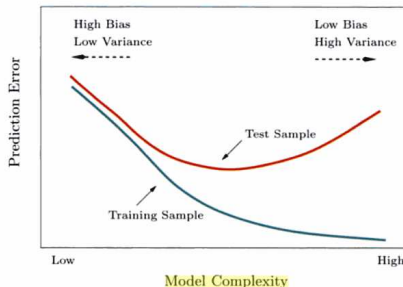
- 100-węzłowe drzewo decyzyjne do problemu “iris”
- 100 neuronów w sieci modelującej problem Xor

Oczywiście **za mało złożony** model nie jest w stanie skutecznie nauczyć się pojęcia (np. pojedynczy neuron dla problemu Xor)
Zbyt złożony model powoduje jednak następujące problemy:

- długi i kosztowny obliczeniowo proces uczenia
- zbyt sztywne dostosowanie do konkretnych przykładów uczących (tzw. **przetrenowanie**) bez możliwości “uogólniania” na nowe nieznane przypadki. W takim przypadku model osiąga b. dobre wyniki tylko na danych trenujących ale na nieznanach przypadkach (poza zbiorem uczącym) model radzi sobie bardzo słabo. (przypomina to uczenie się “na pamięć” przez niektórych studentów)

Złożoność modelu, cd

Zależność pomiędzy złożonością modelu a błędem na danych trenujących i testujących, odpowiednio:



Przetrenowanie jest widoczne w prawej części wykresu (zbyt skomplikowany model). Jak widać, najlepsza złożoność modelu, z punktu widzenia jego skuteczności, jest w środkowej części modelu (obrazek wg: Hastie, Tibshirani "Elements of Statistical Learning", p. 194)

Wybór i ocena modelu

Dwa istotne problemy:

- wybór odpowiedniego modelu i stopnia jego złożoności
- ocena jakości modelu (przewidzenie jak dobrze model będzie działał na faktycznie nieznanach przypadkach)

Jakość oceniana na danych uczących, będzie zawsze *zawyżona*

Jak ocenić jakość modelu?

Jeśli danych treningowych jest wystarczająco dużo:
podzielić dane na **trzy oddzielne zbiory**:

- 1 treningowy (do uczenia się)
- 2 walidacyjny (wybór modelu i kontrola stopnia złożoności)
- 3 testowy (zachowany do momentu ostatecznej oceny modelu)

Nie ma ogólnej reguły na proporcje wielkości, może być np.:
50%, 25%, 25%, respectively

Za mało danych uczących

Wtedy stosuje się inne metody, np:

- **walidacja krzyżowa** (cross-validation)
- leave-one-out
- bootstrap

walidacja krzyżowa jest najbardziej popularna

Walidacja krzyżowa

Pozwala jednocześnie osiągnąć 2 pozornie sprzeczne cele:

- użyć całego zbioru treningowego
- nie oceniać systemu na przykładach ze zbioru treningowego

Dzielimy zbiór treningowy na N rozłącznych części (w sposób losowy). Bierzymy jedną część jako zbiór ewaluacyjny a pozostałe $N-1$ jako treningowe. Powtarzamy N razy (dla każdej części). Łączna Proporcja błędu to uśrednione proporcje ze wszystkich N .

Najczęściej bierze się $N=10$ (ang. 10-fold cross-validation).

Stratyfikacja (ang. stratification)

Polega na tym, że w zbiorze walidującym proporcje przykładów należących do wszystkich kategorii (w zagadnieniu klasyfikacji) są bardzo zbliżone do tych zaobserwowanych w całym pierwotnym zbiorze treningowym.

Technika “leave-one-out” jest szczególnym przypadkiem “cross-validation”. N wynosi tutaj tyle ile jest przypadków w zbiorze treningowym.

- Zbiory walidujące są więc jedno-elementowe.
- Technika ta jest, oczywiście, kosztowna obliczeniowo.
- Zauważmy też, że jej wynik jest deterministyczny (w przeciwieństwie do innych wariantów cross-validation, gdzie podział jest losowy).
- W sposób oczywisty, zbiory walidujące nie są stratyfikowane.

- Macierz Pomyłek (ang. Confusion Matrix)
- Precyzja i Pełność (dla 2-klasowych)
- miara F

Macierz Pomyłek

Założmy, że w problemie klasyfikacji mamy K kategorii. Wtedy macierz pomyłek (ang. confusion matrix) jest narzędziem związanym z ewaluacją klasyfikatora. Jest to macierz kwadratowa wymiaru $K \times K$. Wiersz i odpowiada faktycznej kategorii danego przypadku. Kolumna j odpowiada kategorii, do jakiej system zakwalifikował (być może błędnie) dany przypadek. Komórka (i,j) macierzy zawiera liczbę przypadków ze zbioru ewaluacyjnego, które należą do klasy i , oraz zostały zaklasyfikowane przez system do kategorii j .

zaklasyfikowano jako ->	a	b	c
a = Iris-setosa	50	0	0
b = Iris-versicolor	0	44	6
c = Iris-virginica	0	5	45

Uwaga: idealny klasyfikator miałby niezerowe liczby tylko na przekątnej macierzy pomyłek.

Precyzja i Pełność

W przypadku 2 kategorii (nazwijmy je: “pozytywną” i “negatywną”) można rozważać inne ważne miary ewaluacji: **precyzję** P (ang. precision) i **pełność** R (ang. recall).

Definition

Precyzja P to proporcja przypadków faktycznie pozytywnych wśród wszystkich zaklasyfikowanych przez system jako pozytywne.

Definition

Pełność R to proporcja przypadków faktycznie pozytywnych i zaklasyfikowanych przez system jako pozytywne wśród wszystkich faktycznie pozytywnych.

Oczywiście im wartości miar P i R są wyższe (maks. 1), tym klasyfikator lepszy.

W praktyce jednak, zwykle powiększenie jednej pogarsza drugą (są w pewnym sensie przeciwstawne).

Ponieważ trudno jest równocześnie maksymalizować równocześnie P i R istnieje też inna popularna miara, będąca funkcją obu powyższych. Miarą tą jest **F-miara** (ang. F-measure), zdefiniowana w sposób następujący:

Definition

$$F = \frac{2 \cdot P \cdot R}{P + R}$$

Przykład

Rozważmy następującą macierz pomyłek:

zaklasyfikowano jako ->	pozytywny	negatywny
pozytywny	40	5
negatywny	10	45

$$\text{Precyzja: } P = \frac{40}{(40+10)} = \frac{4}{5}$$

$$\text{Pełność: } R = \frac{40}{(40+5)} = \frac{8}{9}$$

$$\text{F-miara: } F = \frac{2 \cdot \frac{4}{5} \cdot \frac{8}{9}}{\frac{4}{5} + \frac{8}{9}} = \frac{64}{76} = \frac{16}{19}$$

Problemy kontrolne

- model typu “black box”
- reprezentacja wiedzy
- reguły decyzyjne i algorytmy (idea)
- drzewa decyzyjne
- złożoność modelu
- wybór i ocena modelu
- przetrenowanie i jego ominięcie
- walidacja krzyżowa
- Ewaluacja: Macierz Pomyłek, Precyzja, Pełność, F-miara

Dziękuję za uwagę.