Minimalizacja Ryzyka Empirycznego

Systemy uczące się

Mateusz Lango
+ Marek Wydmuch
Zakład Uczenia Maszynowego
Wydział Informatyki i Telekomunikacji
Politechnika Poznańska

„Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”,
projekt finansowany ze środków Programu Operacyjnego Polska Cyfrowa POPC.03.02.00-00-0001/20

Fundusze Europejskie
Polska Cyfrowa

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

Ankieta na start zajęć

https://tinyurl.com/su2024start

Informacje organizacyjne

- Przedmiot *rozszerza* kompetencje dot. systemów uczących się nabytych na studiach I stopnia.
- Dwa pierwsze laboratoria są przewidziane na powtórkę pewnych informacji.
- Wszystkie materiały dydaktyczne na platformie eKursy (również materiały z wykładu).
- Prowadzący lab.: dr hab. inż Maciej Komosiński, prof. PP i mgr inż. Marek Wydmuch
- Obecność rozliczana według standardowych zasad – max. 2 nieuspraw. nieobecności.
- Każdy prowadzący przydziela studentowi do 50% sumarycznej liczby punktów.
- Ocena końcowa według standardowej tabeli:

Punkty	Ocena
$[90\%, 100\%]$	5.0
$[80\%, 90\%)$	4.5
$[70\%, 80\%)$	4.0
$[60\%, 70\%)$	3.5
$[50\%, 60\%)$	3.0
$[0\%, 50\%)$	2.0

Informacje organizacyjne – przebieg lab. i ocena z części M. Wydmucha

- Zajęcia w formie mini-wykładów i rozwiązywania zadań na tablicy.
- Zadania rozwiązywane przez studentów. Możliwość zgłaszania się do rozwiązania zadania, w innym wypadku rozwiązującego wybiera prowadzący. Nie ma konsekwencji za nieumiejętność rozwiązania, ale rozwiązanie zadania jako wolontariusz nagradza +1pp (punkt procentowy) do ostatecznej liczby punktów
- Ocena na podstawie zadań domowych po każdy laboratoriach. Wszystkie zadania są w formie Jupyter Notebooków, które należy wypełnić i przesłać w ciągu 1 tygodnia.
- Każdy rozpoczęty tydzień spóźnienia, obniża maksymalną liczbę punktów, jaką można uzyskać o 10% (pół oceny), ale nie niżej niż 50% (czyli oddając wszystkie zadania w czerwcu, wciąż można otrzymać 3.0).
- Brak możliwości poprawy zadań (lepiej się spóźnić, niż oddać źle zrobione).
- Wykrycie plagiatu będzie skutkować wpisanie 0 punktów za zadanie.
- Prowadzący może odbyć rozmowę ze studentem o zadaniu domowym w celu zweryfikowania samodzielności rozwiązania.

Informacje organizacyjne – przebieg lab. i ocena z części M. Komosińskiego

- Ocena na podstawie mini raportów z każdy zajęć.
- Podobna zasada rozliczania spóźnień i oszustów.
- Więcej szczegółów na zajęciach z Maciejem Komosińskim.

Informacje organizacyjne – konsultacje/pytania z M. Wydmuch

- Kontakt: mwydmuch@cs.put.poznan.pl
- Konsultacje: większość wtorków między 10:15 a 11:15 (w przypadku potrzeby konsultacji w innym terminie proszę pisać maile)

Zaczynamy!

Uczenie maszynowe

Uczenie maszynowe (ang. machine learning) – dział sztucznej inteligencji (AI) poświęcony algorytmom automatycznie poprawiającym swoje działanie poprzez doświadczenie (dane).

Wiele różnych problemów i zastosowań (patrz: dyskusja na wykładzie):

- klasyfikacja SPAM
- automatyczna diagnoza/interpretacja wyników
- wykrywanie nowych zagrożeń w cyberprzestrzeni
- agenty dialogowe
- ...

Uczenie nadzorowane

Uczenie nadzorowane

Zadanie polegające na nauczeniu się funkcji $y = f(x)$ na podstawie przykładowych par wejścia-wyjścia. [za: Wikipedia]

W zależności od typu zmiennej y mówimy o:

- regresji jeśli y jest zmienną ciągłą
- klasyfikacji jeśli y jest ono zmienną dyskretną

Problem

Dlaczego po prostu nie zaimplementować funkcji $f(x)$ np. w Python tylko uruchamiać „uczenie maszynowe”? Dla jakich problemów/sytuacji ma to sens, a dla jakich nie?

Problem

Jakie są wady i zalety rozwiązań korzystających z uczenia maszynowego?

Problemy uczenia nadzorowanego - przykłady

- Regresja *jednowymiarowa*: $f(x) = y \in \mathbb{R}$, np. y $= 4.3$
- Regresja *wielowymiarowa*: $f(x) = \boldsymbol{y} \in \mathbb{R}^m$, np. $\boldsymbol{y} = [1.2, 5.6, \ldots, -0.6]$
- Klasyfikacja *binarna*: $f(x) = y \in \{0, 1\}$ albo $y \in \{-1, 1\}$, np. $y = 1$
- Klasyfikacja *wielo-klasowa*: $f(x) = y \in [m]$[1], np. $y = 5$
- Klasyfikacja *wielo-etykietowa*: $f(x) = \boldsymbol{y} \in \{0, 1\}^m$, np. $\boldsymbol{y} = [1, 0, \ldots, 1]$
- Rangowanie *punktowe*: $f(x + x_j) = y_j \in \mathbb{R}$
- Rangowanie *parami*: $f(x + x_i, x + x_j) = y_{i,j} \in \{0, 1\}$
- $\ldots$

[1] $[m] := (\{1, 2, \ldots, m\})$

Uczenie nadzorowane

Uczenie nadzorowane

Zadanie polegające na nauczeniu się funkcji $y = f(x)$ na podstawie przykładowych par wejścia-wyjścia. [za: Wikipedia]

W zależności od typu zmiennej y mówimy o:

- regresji jeśli y jest zmienną ciągłą
- klasyfikacji jeśli y jest ono zmienną dyskretną

Problem

Dlaczego po prostu nie zaimplementować funkcji $f(x)$ np. w Python tylko uruchamiać „uczenie maszynowe"? Dla jakich problemów/sytuacji ma to sens, a dla jakich nie?

Problem

Jakie są wady i zalety rozwiązań korzystających z uczenia maszynowego?

Uczenie nadzorowane

Uczenie nadzorowane

Zadanie polegające na nauczeniu się funkcji $y = f(x)$ na podstawie przykładowych par wejścia-wyjścia. [za: Wikipedia]

W zależności od typu zmiennej y mówimy o:

- regresji jeśli y jest zmienną ciągłą
- klasyfikacji jeśli y jest ono zmienną dyskretną

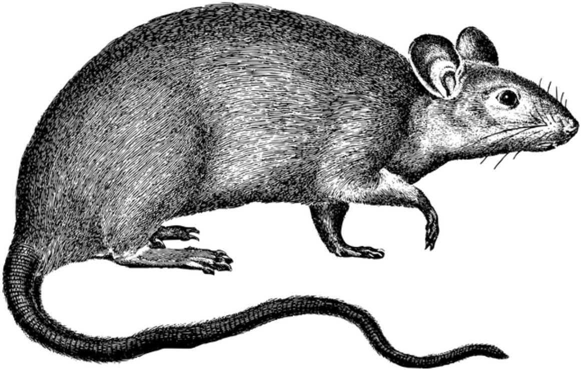
Problem

Dlaczego po prostu nie zaimplementować funkcji $f(x)$ np. w Python tylko uruchamiać „uczenie maszynowe”? Dla jakich problemów/sytuacji ma to sens, a dla jakich nie?

Problem

Jakie są wady i zalety rozwiązań korzystających z uczenia maszynowego?

Uczenie się: jak szczury uczą się unikać trucizn?[1]

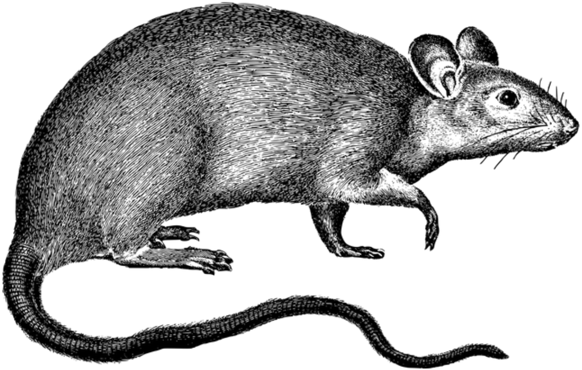


- Szczur odnajdujący nowy rodzaj jedzenia zjada tylko jego małą porcję
- Jeśli po nim zachoruje – unika go w przyszłości
- Proste uczenie się przez zapamiętywanie (jeden rodzaj jedzenia = jedna obserwacja)

To podejście, choć działające w tym przypadku ma jednak wadę: brak *uogólniania* na nowe przykłady

[1] za: *Understanding Machine Learning from Theory to Practice*, S. Shalev-Shwartz & S. Ben-David

Uczenie się: jak szczury uczą się unikać trucizn?[1]



- Szczur odnajdujący nowy rodzaj jedzenia zjada tylko jego małą porcję
- Jeśli po nim zachoruje – unika go w przyszłości
- Proste uczenie się przez zapamiętywanie (jeden rodzaj jedzenia = jedna obserwacja)

To podejście, choć działające w tym przypadku ma jednak wadę: brak *uogólniania* na nowe przykłady

[1] za: *Understanding Machine Learning from Theory to Practice*, S. Shalev-Shwartz & S. Ben-David

Uczenie się: przesądny gołąb[2]

Po 20 sekundach ptak robi przypadkową czynność np. macha skrzydłami

Koajrzy machanie skrzydłami z jedzeniem

Troszkę cześciej macha skrzydłami

Po 20 sekundach z większym prawdopodobieństwem ptak będzie akurat machał skrzydłami

Foto: Tim Bradshaw

https://youtu.be/TtfQlkGwE2U?t=25

https://youtu.be/Qv4H81gEGDQ

Uczenie się: przesądny gołąb^2

Foto: Tim Bradshaw

https://youtu.be/TtfQlkGwE2U?t=25

https://youtu.be/Qv4H81gEGDQ

Po 20 sekundach ptak robi przypadkową czynność np. macha skrzydłami

↓

Koajrzy machanie skrzydłami z jedzeniem

↓

Troszkę cześciej macha skrzydłami

↓

Po 20 sekundach z większym prawdopodobieństwem ptak będzie akurat machał skrzydłami

Uczenie się: przesądny gołąb[2]

Foto: Tim Bradshaw

https://youtu.be/TtfQlkGwE2U?t=25

https://youtu.be/Qv4H81gEGDQ

Po 20 sekundach ptak robi przypadkową czynność np. macha skrzydłami

↓

Koajrzy machanie skrzydłami z jedzeniem

↓

Troszkę cześciej macha skrzydłami

↓

Po 20 sekundach z większym prawdopodobieństwem ptak będzie akurat machał skrzydłami

Uczenie się: przesądny gołąb[2]

Foto: Tim Bradshaw

https://youtu.be/TtfQlkGwE2U?t=25

https://youtu.be/Qv4H81gEGDQ

Po 20 sekundach ptak robi przypadkową czynność np. macha skrzydłami

↓

Koajrzy machanie skrzydłami z jedzeniem

↓

Troszkę cześciej macha skrzydłami

↓

Po 20 sekundach z większym prawdopodobieństwem ptak będzie akurat machał skrzydłami

Uczenie się: przesądny gołąb[2]

Foto: Tim Bradshaw

https://youtu.be/TtfQlkGwE2U?t=25

https://youtu.be/Qv4H81gEGDQ

Po 20 sekundach ptak robi przypadkową czynność np. macha skrzydłami

↓

Koajrzy machanie skrzydłami z jedzeniem

↓

Troszkę cześciej macha skrzydłami

↓

Po 20 sekundach z większym prawdopodobieństwem ptak będzie akurat machał skrzydłami

Uczenie się: przesądny gołąb[3]

Foto: Tim Bradshaw

https://youtu.be/TtfQlkGwE2U?t=25

https://youtu.be/Qv4H81gEGDQ

- Co poszło nie tak?
- Wcześniej: zapamiętywanie prowadziło do unikania trucizny przez szczura
- Podobny eksperyment na szczurach: jedzenie „powodowało” wstrząs elektryczny
- Szczur nie był w stanie skojarzyć, że dany rodzaj jedzenia powoduje późniejszy ból, choć był w stanie nauczyć się że jedzenie powoduje (też z opóźnieniem) problemy z trawieniem

[3] za: *Understanding Machine Learning from Theory to Practice*, S. Shalev-Shwartz & S. Ben-David

Skuteczne uczenie się - obserwacje

Skuteczne uczenie się powinno się składać z następujących komponentów:

1. zapamiętanie / *zbudowanie reprezentacji wiedzy* z doświadczeń (danych)
2. *uogólnienie* tej wiedzy na inne sytuacje/przykłady
3. ignorowanie przypadkowych korelacji poprzez eliminację niektórych hipotez, najczęściej z wiedzy wstępnej o problemie (ang. inductive bias)

Algorytm uczący się – schemat

Nieznana funkcja docelowa
$f : X \to Y$

Dane uczące
$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

Zwykle $y_i \neq f(x_i)$, a jedynie
$y_i = f(x_i) + \epsilon$

Zbiór hipotez $\mathcal{H}$

Algortym uczący

Wybrana hipoteza
$g \approx f$

Algorytm uczący się – schemat

Nieznana funkcja docelowa
$f : X \to Y$

Dane uczące
$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

Zwykle $y_i \neq f(x_i)$, a jedynie
$y_i = f(x_i) + \epsilon$

Zbiór hipotez $\mathcal{H}$

Algortym uczący

Wybrana hipoteza
$g \approx f$

Algorytm uczący się – schemat

Nieznana funkcja docelowa
$f : X \to Y$

Dane uczące
$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

Zwykle $y_i \neq f(x_i)$, a jedynie
$y_i = f(x_i) + \epsilon$

Zbiór hipotez $\mathcal{H}$

Algortym uczący

Wybrana hipoteza
$g \approx f$

Algorytm uczący się – schemat

Nieznana funkcja docelowa
$f : X \to Y$

Dane uczące
$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

Zwykle $y_i \neq f(x_i)$, a jedynie
$y_i = f(x_i) + \epsilon$

Zbiór hipotez $\mathcal{H}$

Algortym uczący

Wybrana hipoteza
$g \approx f$

Algorytm uczący się – schemat

Nieznana funkcja docelowa
$f : X \to Y$

Dane uczące
$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

Zwykle $y_i \neq f(x_i)$, a jedynie
$y_i = f(x_i) + \epsilon$

Zbiór hipotez $\mathcal{H}$

Algortym uczący

Wybrana hipoteza
$g \approx f$

Inżynieria cech – budowa reprezentacji problemu

- dobra reprezentacja problemu x_i umożliwia lepszy wynik uczenia np. poprzez częściową eliminację czynnika ϵ

Problem

Zaproponuj co najmniej 5 cech dla modelu filtrującego wiadomości e-mail (SPAM/$\neg$SPAM)

Problem

Jaka jest charakterystyka/własności dobrej cechy? Spróbuj odnieść się do poniższych przykładów cech:

- *student_id = 299616*
- *specjalizacja = SI*
- *wiek studenta zakodowany jako ciąg 8 zmiennych binarnych, systemem dwójkowym*
- *wynik z rozmowy kwalifikacyjnej = -1 (jeśli do niej nie przystąpiono)*
- *L.p. na liście kandydatów = 12*

Inżynieria cech – budowa reprezentacji problemu

- dobra reprezentacja problemu x_i umożliwia lepszy wynik uczenia np. poprzez częściową eliminację czynnika ϵ

Problem

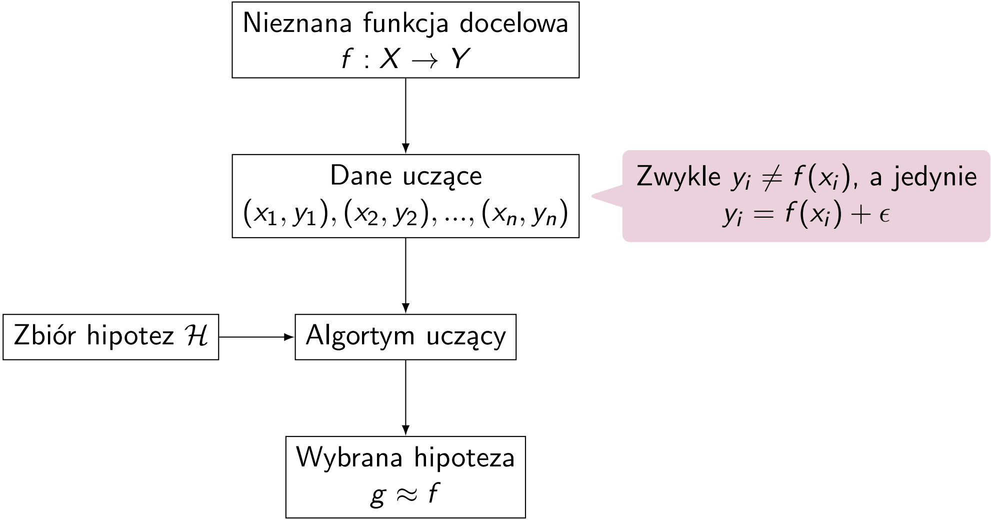
Zaproponuj co najmniej 5 cech dla modelu filtrującego wiadomości e-mail (SPAM/$\neg$SPAM)

Problem

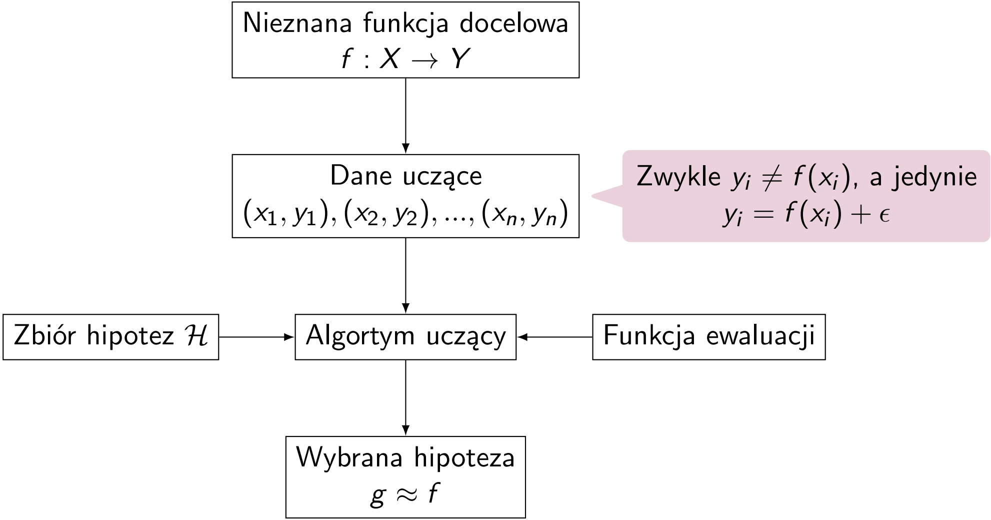
Jaka jest charakterystyka/własności dobrej cechy? Spróbuj odnieść się do poniższych przykładów cech:

- *student_id = 299616*
- *specjalizacja = SI*
- *wiek studenta zakodowany jako ciąg 8 zmiennych binarnych, systemem dwójkowym*
- *wynik z rozmowy kwalifikacyjnej = -1 (jeśli do niej nie przystąpiono)*
- *L.p. na liście kandydatów = 12*

Jak projektować algorytmy uczące?



Jak projektować algorytmy uczące?



Funkcje ewaluacji (oceny)

- problem regresji

$$L(\hat{y}, y) = (\hat{y} - y)^2$$

- problem klasyfikacji

$$L(\hat{y}, y) = \begin{cases} 0 & \hat{y} = y \\ 1 & \hat{y} \neq y \end{cases}$$

Jak projektować algorytmy uczące?

- Naszym celem jest osiągnięcie najlepszego uogólniania wiedzy tj. chcielibyśmy aby

$$\hat{f} = \arg\min \mathbb{E}[L(g(X), Y)]$$

- Niestety nie jest to możliwe, gdyż jest to średnia po wszystkich *możliwych* danych
- Zasada minimalizacji ryzyka empirycznego (ang. *empirical risk minimization, ERM*):

$$\hat{f} = \arg\min_{g \in \mathcal{H}} \frac{1}{n} \sum_{\text{dane uczace}} [L(g(X), Y)]$$

- Algorytm uczący się ma trzy części: reprezentację wiedzy, funkcję ewaluacji i algorytm optymalizacyjny[4]

[4] Pedro Domingos, *A few useful things to know about Machine Learning*, Communications of the ACM, 2012

Jak projektować algorytmy uczące?

- Naszym celem jest osiągnięcie najlepszego uogólniania wiedzy tj. chcielibyśmy aby

$$\hat{f} = \arg\min \mathbb{E}[L(g(X), Y)]$$

- Niestety nie jest to możliwe, gdyż jest to średnia po wszystkich *możliwych* danych
- Zasada minimalizacji ryzyka empirycznego (ang. *empirical risk minimization, ERM*):

$$\hat{f} = \arg\min_{g \in \mathcal{H}} \frac{1}{n} \sum_{\textit{dane uczace}} [L(g(X), Y)]$$

- Algorytm uczący się ma trzy części: reprezentację wiedzy, funkcję ewaluacji i algorytm optymalizacyjny[4]

[4] Pedro Domingos, *A few useful things to know about Machine Learning*, Communications of the ACM, 2012

Regresja liniowa – 3 części

1. Reprezentacja – hipotezami są zestawy wektorów o długości $d + 1$ (liczba cech $+ 1$) zawierające wagi oraz wyrazy wolne

$$\mathcal{H} = \{w : w \in \mathbb{R}^{d+1}\}$$

2. Funkcja celu – błąd kwadratowy

$$L(\hat{y}, y) = (\hat{y} - y)^2$$

3. Algorytm optymalizacyjny: rozwiązanie równania wynikającego z przyrównania pochodnej do zera

Problem

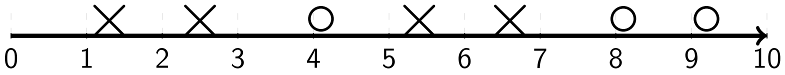
Zastanów się w jaki sposób regresja liniowa osiąga omawiane 3 cele uczenia się: (zapamiętanie) konstrukcja reprezentacji wiedzy z danych, uogólnianie i ignorowanie przypadkowych korelacji.

Zadanie

Problem

Zakładając poniższy zbiór do klasyfikacji binarnej $(\times, \circ)$ *z jedną cechą* x *oraz następującą klasę hipotez* $\mathcal{H} = \{\times \text{ if } x < t \text{ else } \circ : t \in \mathbb{R}\} \cup \{\times \text{ if } x > t \text{ else } \circ : t \in \mathbb{R}\}$

1. *Który klasyfikator zostanie wybrany poprzez ERM?*
2. *W jaki sposób można zaimplementować algorytm wybierający klasyfikator zgodny z ERM dla tej klasy hipotez?*
3. *Jak wyglądałoby rozszerzenie tego problemu i algorytmu dla zbioru z dwoma cechami?*
4. *Oszacuj złożoność obliczeniową zaproponowanego algorytmu.*



Zadanie *

Problem

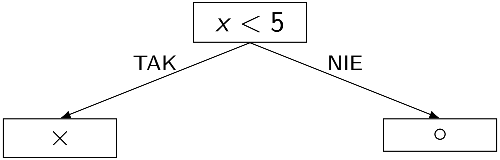
Rozważmy problem klasyfikacji binarnej ze skończonym $\mathcal{H}$ i założeniem że $f \in \mathcal{H}$. W takim wypadku jest możliwe użycie następującego algorytmu uczącego:

1. *Odczytaj przykład uczący (x_i, y_i)*
2. *Wyeliminuj z $\mathcal{H}$ wszystkie g takie że $g(x_i) \neq y_i$*
3. *Powtarzaj, a na końcu wybierz dowolną hipotezę która pozostała w $\mathcal{H}$*

a) Zakładając n-elementowy zbiór niezależnych danych uczących, podaj wzór na prawdopodobieństwo że w ostatecznym zbiorze $\mathcal{H}$ będzie hipoteza z błędem klasyfikacji większym niż $\epsilon \in [0, 1]$. (Innymi słowy: jak jest szansa że ten algorytm dostarczy klasyfikator z błędem większym niż ϵ.
b) Co najmniej ilu przykładów uczących n potrzebujesz, abyś miał 90% pewność, że algorytm nie zwróci klasyfikatora z błędem większym niż ϵ?

Decision stump a drzewo decyzyne

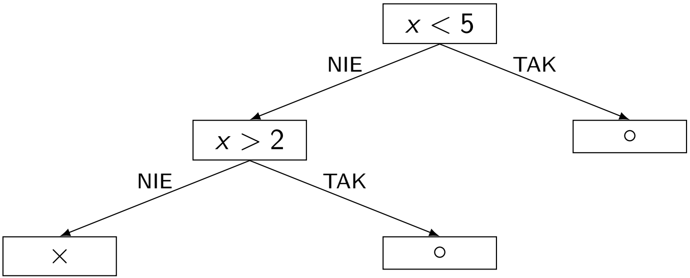
Klasyfikatory z klasy hipotez rozważanych w poprzednim zadaniu możemy zwizualizować jako:



Tego typu klasyfikator dzieli nam zbiór danych na dwie części: jedna zawierająca wszystkie elementy $x < 5$ i druga $x \geq 5$. Stosując taki klasyfikator ponownie do uzyskanych części otrzymujemy *drzewo decyzyjne*.

Decision stump a drzewo decyzyne

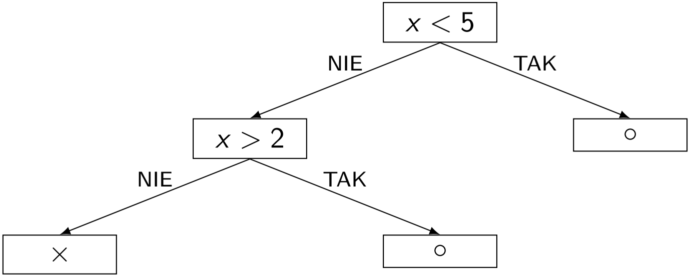
Klasyfikatory z klasy hipotez rozważanych w poprzednim zadaniu możemy zwizualizować jako:



Tego typu klasyfikator dzieli nam zbiór danych na dwie części: jedna zawierająca wszystkie elementy $x < 5$ i druga $x \geq 5$. Stosując taki klasyfikator ponownie do uzyskanych części otrzymujemy *drzewo decyzyjne*.

Decision stump a drzewo decyzyne

Klasyfikatory z klasy hipotez rozważanych w poprzednim zadaniu możemy zwizualizować jako:



Tego typu klasyfikator dzieli nam zbiór danych na dwie części: jedna zawierająca wszystkie elementy $x < 5$ i druga $x \geq 5$. Stosując taki klasyfikator ponownie do uzyskanych części otrzymujemy *drzewo decyzyjne*.

Problem

Jak mogłyby wyglądać „decision stump" dla cech nominalnych?

Drzewa decyzyjne – jak się uczyć?

- Zasada minimalizacji ryzyka empirycznego (ang. *empirical risk minimization, ERM*):

$$\hat{f} = \arg\min_{g \in \mathcal{H}} \frac{1}{n} \sum_{\textit{dane uczace}} [L(g(X), Y)]$$

  gdzie L to błąd zero-jedynkowy.
- wspomniane 3 cele uczenia się:
  - (zapamiętanie) reprezentacja wiedzy z danych
  - uogólnianie
  - ignorowanie przypadkowych korelacji
- Niech $\mathcal{H}$ zawiera tylko najmniejsze drzewo dla każdej (możliwej do zareprezentowania) funkcji
- ERM w tej sytuacji jest NP-zupełny, wersja decyzyjna NP-trudna, przybliżenie $(1+\epsilon)$ NP-trudne, $(4-\epsilon)$ NP-trudne, ... $\Rightarrow$ w praktyce nie stosujemy optymalnych algorytmów[5]

[5] Z wyjątkiem małych zbiorów: Hu et al. *Optimal Sparse Decision Trees*, NeurIPS 2019

Drzewa decyzyjne – jak się uczyć?

- Zasada minimalizacji ryzyka empirycznego (ang. *empirical risk minimization, ERM*):

$$\hat{f} = \arg\min_{g \in \mathcal{H}} \frac{1}{n} \sum_{\textit{dane uczace}} [L(g(X), Y)]$$

  gdzie L to błąd zero-jedynkowy.
- wspomniane 3 cele uczenia się:
  - (zapamiętanie) reprezentacja wiedzy z danych
  - uogólnianie
  - ignorowanie przypadkowych korelacji
- Niech $\mathcal{H}$ zawiera tylko najmniejsze drzewo dla każdej (możliwej do zareprezentowania) funkcji
- ERM w tej sytuacji jest NP-zupełny, wersja decyzyjna NP-trudna, przybliżenie $(1+\epsilon)$ NP-trudne, $(4-\epsilon)$ NP-trudne, ... $\Rightarrow$ w praktyce nie stosujemy optymalnych algorytmów[5]

[5] Z wyjątkiem małych zbiorów: Hu et al. *Optimal Sparse Decision Trees*, NeurIPS 2019

Drzewo decyzyjne - prosty algorytm uczący

1. Reprezentacja – drzewa z podziałami binarnymi dla cech ciągłych, dla cech nominalnych podziały równościowe
2. Funkcja celu – ERM z błędem zero-jedynkowym

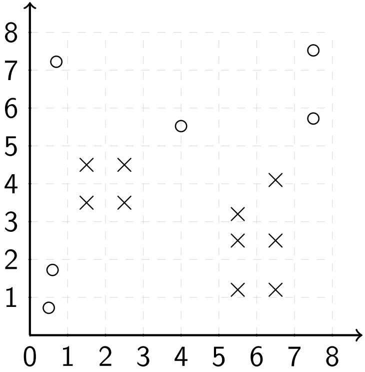
$$L(\hat{y}, y) = \begin{cases} 0 & \hat{y} = y \\ 1 & \hat{y} \neq y \end{cases}$$

3. Algorytm optymalizacyjny: algorytm zachłanny
   - Sprawdź wszystkie możliwe podziały i wybierz ten, który najbardziej optymalizuje funkcję celu tj. błąd klasyfikacji
   - Jeśli nie jest możliwy podział poprawiający funkcję celu – stwórz liść
   - Utwórz podział i rekurencyjnie wywołaj procedurę w każdym z liści

Zadanie

Problem

Zbuduj drzewo decyzyjne dla poniższego zbioru danych.



Przykład obliczeń: uproszczony „Play golf?” [Quinlan ’86]

Outlook	Windy	Play?
słonecznie	false	○
słonecznie	true	○
pochmurnie	false	×
deszcz	false	×
deszcz	false	×
deszcz	true	○
pochmurnie	true	×
słonecznie	false	○
słonecznie	false	×
deszcz	false	×
słonecznie	true	×
pochmurnie	true	×
pochmurnie	false	×
deszcz	true	○

Zadanie

Rozważmy problem klasyfikacji binarnej ze skończoną klasą hipotez $\mathcal{H}$ i założeniem że funkcja f, której chcemy się nauczyć należy do tej klasy hipotez $f \in \mathcal{H}$. W takim wypadku jest możliwe użycie następującego algorytmu uczącego:

1. Odczytaj kolejny przykład uczący (x_i, y_i)
2. Wyeliminuj z $\mathcal{H}$ wszystkie hipotezy h takie że $h(x_i) \neq y_i$
3. Powtarzaj 1 i 2, a na końcu wybierz dowolną hipotezę która pozostała w $\mathcal{H}$.

Pytanie

Zakładając n-elementowy zbiór niezależnych danych uczących, podaj wzór na prawdopodobieństwo że w ostatecznym zbiorze $\mathcal{H}$ będzie hipoteza z błędem klasyfikacji równym ϵ.

Pytanie

Zakładając n-elementowy zbiór niezależnych danych uczących, podaj wzór na prawdopodobieństwo że w ostatecznym zbiorze $\mathcal{H}$ będzie hipoteza z błędem klasyfikacji równym ϵ.

Zadanie

Rozważmy tworzenie podziału zbioru do klasyfikacji binarnej w którym prawdopodobieństwo klasy $+$ wynosi $p = 0.4$, a po podziale uzyskano podzbiory z $p_1 = 0.1$ i $p_2 = 0.7$.
Wskazówka: odpowiedz na pytania używając oznaczeń:

$$p = \frac{n_+}{n} \qquad p_1 = \frac{n_{1,+}}{n_1} \qquad p_2 = \frac{n_{2,+}}{n_2} \qquad x = \frac{n_1}{n}$$

- Ile procent przykładów znalazło się w pierwszym podzbiorze (x)?
- Ile wynosi błąd po wykonaniu podziału?

Zadanie

Rozważmy tworzenie podziału zbioru do klasyfikacji binarnej w którym prawdopodobieństwo klasy $+$ wynosi $p = 0.4$, a po podziale uzyskano podzbiory z $p_1 = 0.1$ i $p_2 = 0.7$.
Wskazówka: odpowiedz na pytania używając oznaczeń:

$$p = \frac{n_+}{n} \qquad p_1 = \frac{n_{1,+}}{n_1} \qquad p_2 = \frac{n_{2,+}}{n_2} \qquad x = \frac{n_1}{n}$$

- Ile procent przykładów znalazło się w pierwszym podzbiorze (x)?

$$p = \frac{n_+}{n} = \frac{n_{1,+} + n_{2,+}}{n_1 + n_2}$$

- Ile wynosi błąd po wykonaniu podziału?

Zadanie

Rozważmy tworzenie podziału zbioru do klasyfikacji binarnej w którym prawdopodobieństwo klasy + wynosi $p = 0.4$, a po podziale uzyskano podzbiory z $p_1 = 0.1$ i $p_2 = 0.7$.
Wskazówka: odpowiedz na pytania używając oznaczeń:

$$p = \frac{n_+}{n} \qquad p_1 = \frac{n_{1,+}}{n_1} \qquad p_2 = \frac{n_{2,+}}{n_2} \qquad x = \frac{n_1}{n}$$

- Ile procent przykładów znalazło się w pierwszym podzbiorze (x)?

$$p = \frac{n_+}{n} = \frac{n_{1,+} + n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1 + n_2} + \frac{n_{2,+}}{n_1 + n_2}$$

- Ile wynosi błąd po wykonaniu podziału?

Zadanie

Rozważmy tworzenie podziału zbioru do klasyfikacji binarnej w którym prawdopodobieństwo klasy $+$ wynosi $p = 0.4$, a po podziale uzyskano podzbiory z $p_1 = 0.1$ i $p_2 = 0.7$.
Wskazówka: odpowiedz na pytania używając oznaczeń:

$$p = \frac{n_+}{n} \qquad p_1 = \frac{n_{1,+}}{n_1} \qquad p_2 = \frac{n_{2,+}}{n_2} \qquad x = \frac{n_1}{n}$$

- Ile procent przykładów znalazło się w pierwszym podzbiorze (x)?

$$p = \frac{n_+}{n} = \frac{n_{1,+} + n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1 + n_2} + \frac{n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1}\frac{n_1}{n_1 + n_2} + \frac{n_{2,+}}{n_2}\frac{n_2}{n_1 + n_2}$$

- Ile wynosi błąd po wykonaniu podziału?

Zadanie

Rozważmy tworzenie podziału zbioru do klasyfikacji binarnej w którym prawdopodobieństwo klasy $+$ wynosi $p = 0.4$, a po podziale uzyskano podzbiory z $p_1 = 0.1$ i $p_2 = 0.7$.
Wskazówka: odpowiedz na pytania używając oznaczeń:

$$p = \frac{n_+}{n} \qquad p_1 = \frac{n_{1,+}}{n_1} \qquad p_2 = \frac{n_{2,+}}{n_2} \qquad x = \frac{n_1}{n}$$

- Ile procent przykładów znalazło się w pierwszym podzbiorze (x)?

$$p = \frac{n_+}{n} = \frac{n_{1,+} + n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1 + n_2} + \frac{n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1}\frac{n_1}{n_1 + n_2} + \frac{n_{2,+}}{n_2}\frac{n_2}{n_1 + n_2}$$
$$= p_1 x + p_2(1 - x)$$

- Ile wynosi błąd po wykonaniu podziału?

Zadanie

Rozważmy tworzenie podziału zbioru do klasyfikacji binarnej w którym prawdopodobieństwo klasy + wynosi $p = 0.4$, a po podziale uzyskano podzbiory z $p_1 = 0.1$ i $p_2 = 0.7$.
Wskazówka: odpowiedz na pytania używając oznaczeń:

$$p = \frac{n_+}{n} \qquad p_1 = \frac{n_{1,+}}{n_1} \qquad p_2 = \frac{n_{2,+}}{n_2} \qquad x = \frac{n_1}{n}$$

- Ile procent przykładów znalazło się w pierwszym podzbiorze (x)?

$$p = \frac{n_+}{n} = \frac{n_{1,+} + n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1 + n_2} + \frac{n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1} \frac{n_1}{n_1 + n_2} + \frac{n_{2,+}}{n_2} \frac{n_2}{n_1 + n_2}$$
$$= p_1 x + p_2(1 - x)$$

$$0.4 = 0.1x + 0.7(1 - x)$$

- Ile wynosi błąd po wykonaniu podziału?

Zadanie

Rozważmy tworzenie podziału zbioru do klasyfikacji binarnej w którym prawdopodobieństwo klasy $+$ wynosi $p = 0.4$, a po podziale uzyskano podzbiory z $p_1 = 0.1$ i $p_2 = 0.7$.
Wskazówka: odpowiedz na pytania używając oznaczeń:

$$p = \frac{n_+}{n} \qquad p_1 = \frac{n_{1,+}}{n_1} \qquad p_2 = \frac{n_{2,+}}{n_2} \qquad x = \frac{n_1}{n}$$

- Ile procent przykładów znalazło się w pierwszym podzbiorze (x)?

$$p = \frac{n_+}{n} = \frac{n_{1,+} + n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1 + n_2} + \frac{n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1}\frac{n_1}{n_1 + n_2} + \frac{n_{2,+}}{n_2}\frac{n_2}{n_1 + n_2}$$
$$= p_1 x + p_2(1 - x)$$

$$0.4 = 0.1x + 0.7 - 0.7x$$

- Ile wynosi błąd po wykonaniu podziału?

Zadanie

Rozważmy tworzenie podziału zbioru do klasyfikacji binarnej w którym prawdopodobieństwo klasy + wynosi $p = 0.4$, a po podziale uzyskano podzbiory z $p_1 = 0.1$ i $p_2 = 0.7$.
Wskazówka: odpowiedz na pytania używając oznaczeń:

$$p = \frac{n_+}{n} \qquad p_1 = \frac{n_{1,+}}{n_1} \qquad p_2 = \frac{n_{2,+}}{n_2} \qquad x = \frac{n_1}{n}$$

- Ile procent przykładów znalazło się w pierwszym podzbiorze (x)?

$$\begin{aligned} p &= \frac{n_+}{n} = \frac{n_{1,+} + n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1 + n_2} + \frac{n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1} \frac{n_1}{n_1 + n_2} + \frac{n_{2,+}}{n_2} \frac{n_2}{n_1 + n_2} \\ &= p_1 x + p_2(1 - x) \end{aligned}$$

$$0.4 = 0.1x + 0.7 - 0.7x$$

$$-0.3 = -0.6x$$

- Ile wynosi błąd po wykonaniu podziału?

Zadanie

Rozważmy tworzenie podziału zbioru do klasyfikacji binarnej w którym prawdopodobieństwo klasy $+$ wynosi $p = 0.4$, a po podziale uzyskano podzbiory z $p_1 = 0.1$ i $p_2 = 0.7$.
Wskazówka: odpowiedz na pytania używając oznaczeń:

$$p = \frac{n_+}{n} \qquad p_1 = \frac{n_{1,+}}{n_1} \qquad p_2 = \frac{n_{2,+}}{n_2} \qquad x = \frac{n_1}{n}$$

- Ile procent przykładów znalazło się w pierwszym podzbiorze (x)?

$$p = \frac{n_+}{n} = \frac{n_{1,+} + n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1 + n_2} + \frac{n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1}\frac{n_1}{n_1 + n_2} + \frac{n_{2,+}}{n_2}\frac{n_2}{n_1 + n_2}$$
$$= p_1 x + p_2(1 - x)$$

$$0.4 = 0.1x + 0.7 - 0.7x$$
$$-0.3 = -0.6x \qquad \Rightarrow x = 0.5$$

- Ile wynosi błąd po wykonaniu podziału?

Zadanie

Rozważmy tworzenie podziału zbioru do klasyfikacji binarnej w którym prawdopodobieństwo klasy $+$ wynosi $p = 0.4$, a po podziale uzyskano podzbiory z $p_1 = 0.1$ i $p_2 = 0.7$.
Wskazówka: odpowiedz na pytania używając oznaczeń:

$$p = \frac{n_+}{n} \qquad p_1 = \frac{n_{1,+}}{n_1} \qquad p_2 = \frac{n_{2,+}}{n_2} \qquad x = \frac{n_1}{n}$$

- Ile procent przykładów znalazło się w pierwszym podzbiorze (x)?

$$p = \frac{n_+}{n} = \frac{n_{1,+} + n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1 + n_2} + \frac{n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1}\frac{n_1}{n_1 + n_2} + \frac{n_{2,+}}{n_2}\frac{n_2}{n_1 + n_2}$$
$$= p_1 x + p_2(1 - x)$$

$$0.4 = 0.1x + 0.7 - 0.7x$$
$$-0.3 = -0.6x \qquad \Rightarrow x = 0.5$$

- Ile wynosi błąd po wykonaniu podziału?

$$\epsilon_{new} = x\epsilon_1 + (1 - x)\epsilon_2$$

Zadanie

Rozważmy tworzenie podziału zbioru do klasyfikacji binarnej w którym prawdopodobieństwo klasy $+$ wynosi $p = 0.4$, a po podziale uzyskano podzbiory z $p_1 = 0.1$ i $p_2 = 0.7$.
Wskazówka: odpowiedz na pytania używając oznaczeń:

$$p = \frac{n_+}{n} \qquad p_1 = \frac{n_{1,+}}{n_1} \qquad p_2 = \frac{n_{2,+}}{n_2} \qquad x = \frac{n_1}{n}$$

- Ile procent przykładów znalazło się w pierwszym podzbiorze (x)?

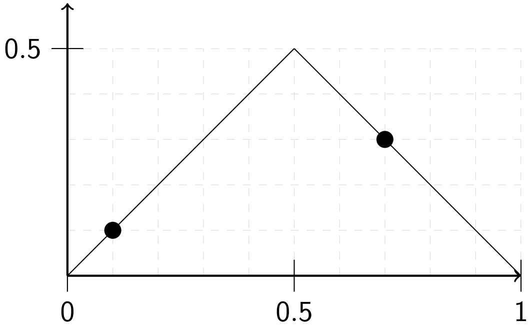
$$p = \frac{n_+}{n} = \frac{n_{1,+} + n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1 + n_2} + \frac{n_{2,+}}{n_1 + n_2} = \frac{n_{1,+}}{n_1} \frac{n_1}{n_1 + n_2} + \frac{n_{2,+}}{n_2} \frac{n_2}{n_1 + n_2}$$
$$= p_1 x + p_2(1 - x)$$

$$0.4 = 0.1x + 0.7 - 0.7x$$
$$-0.3 = -0.6x \qquad \Rightarrow x = 0.5$$

- Ile wynosi błąd po wykonaniu podziału?

$$\epsilon_{new} = x\epsilon_1 + (1 - x)\epsilon_2 = 0.5 \cdot 0.1 + 0.5 \cdot 0.3 = 0.2$$

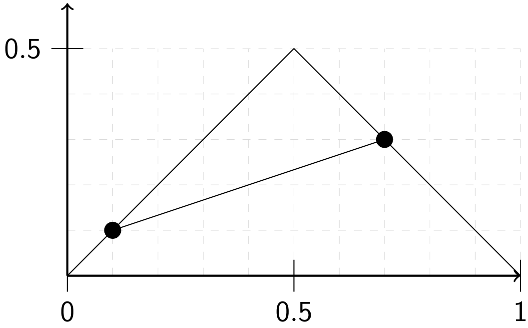
Interpretacja geometryczna



Rozważmy tworzenie podziału zbioru którego początkowe $p = 0.4$, a po podziale uzyskano podzbiory o $p_1 = 0.1$ i $p_2 = 0.7$.

- Błąd po podziale to
$$\epsilon_{new} = x\epsilon_1 + (1 - x)\epsilon_2$$
czyli leży na linii łączącej ϵ_1 i ϵ_2.
- Ponieważ wzór na p jest analogiczny (z tymi samymi wagami) i dotyczy osi x – wyznacza on miejsce odczytu ϵ_{new}
$$p = p_1 x + p_2(1 - x)$$
- Zysk z wykonania podziału to zaznaczona odległość.

Interpretacja geometryczna



Rozważmy tworzenie podziału zbioru którego początkowe $p = 0.4$, a po podziale uzyskano podzbiory o $p_1 = 0.1$ i $p_2 = 0.7$.

- Błąd po podziale to

$$\epsilon_{new} = x\epsilon_1 + (1 - x)\epsilon_2$$

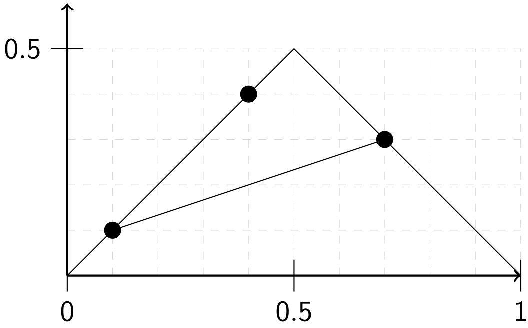
  czyli leży na linii łączącej ϵ_1 i ϵ_2.

- Ponieważ wzór na p jest analogiczny (z tymi samymi wagami) i dotyczy osi x – wyznacza on miejsce odczytu ϵ_{new}

$$p = p_1 x + p_2(1 - x)$$

- Zysk z wykonania podziału to zaznaczona odległość.

Interpretacja geometryczna



Rozważmy tworzenie podziału zbioru którego początkowe $p = 0.4$, a po podziale uzyskano podzbiory o $p_1 = 0.1$ i $p_2 = 0.7$.

- Błąd po podziale to

  $$\epsilon_{new} = x\epsilon_1 + (1 - x)\epsilon_2$$

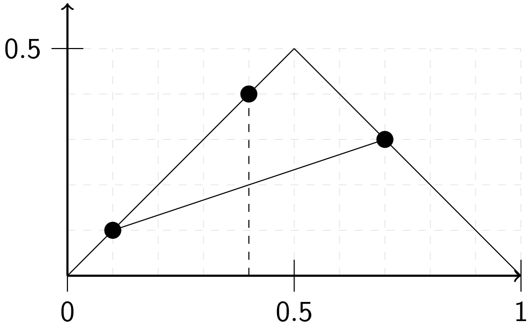
  czyli leży na linii łączącej ϵ_1 i ϵ_2.

- Ponieważ wzór na p jest analogiczny (z tymi samymi wagami) i dotyczy osi x – wyznacza on miejsce odczytu ϵ_{new}

  $$p = p_1 x + p_2(1 - x)$$

- Zysk z wykonania podziału to zaznaczona odległość.

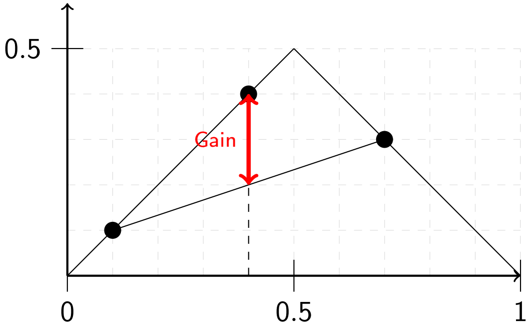
Interpretacja geometryczna



Rozważmy tworzenie podziału zbioru którego początkowe $p = 0.4$, a po podziale uzyskano podzbiory o $p_1 = 0.1$ i $p_2 = 0.7$.

- Błąd po podziale to
$$\epsilon_{new} = x\epsilon_1 + (1 - x)\epsilon_2$$
czyli leży na linii łączącej ϵ_1 i ϵ_2.
- Ponieważ wzór na p jest analogiczny (z tymi samymi wagami) i dotyczy osi x – wyznacza on miejsce odczytu ϵ_{new}
$$p = p_1 x + p_2(1 - x)$$
- Zysk z wykonania podziału to zaznaczona odległość.

Interpretacja geometryczna



Rozważmy tworzenie podziału zbioru którego początkowe $p = 0.4$, a po podziale uzyskano podzbiory o $p_1 = 0.1$ i $p_2 = 0.7$.

- Błąd po podziale to

$$\epsilon_{new} = x\epsilon_1 + (1-x)\epsilon_2$$

  czyli leży na linii łączącej ϵ_1 i ϵ_2.

- Ponieważ wzór na p jest analogiczny (z tymi samymi wagami) i dotyczy osi x – wyznacza on miejsce odczytu ϵ_{new}

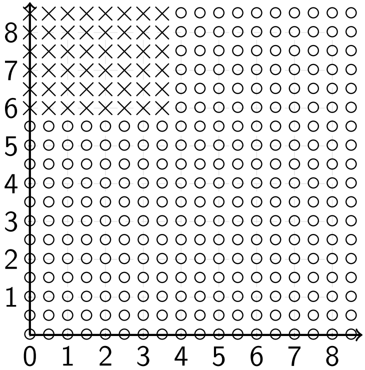
$$p = p_1 x + p_2(1-x)$$

- Zysk z wykonania podziału to zaznaczona odległość.

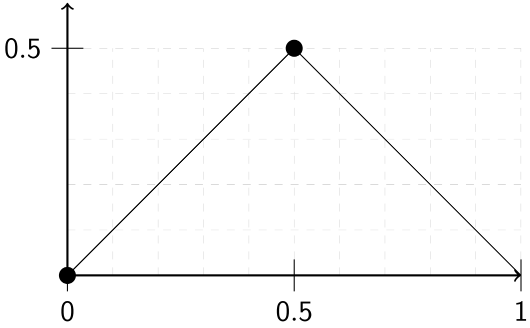
Zadanie

Problem

Zbuduj drzewo decyzyjne dla poniższego zbioru danych.

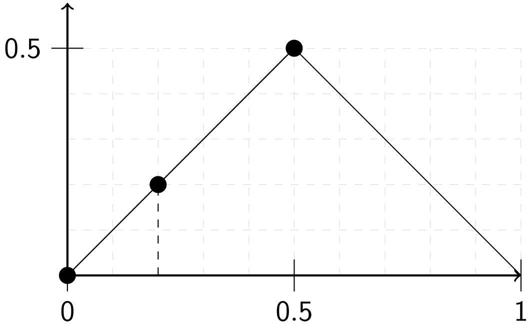


Co się stało? Interpretacja geometryczna



Przed podziałem mamy $p = 0.2$, a po potencjalnym podziale po x_1 uzyskano podzbiory o $p_1 = 0.5$ i $p_2 = 0$.

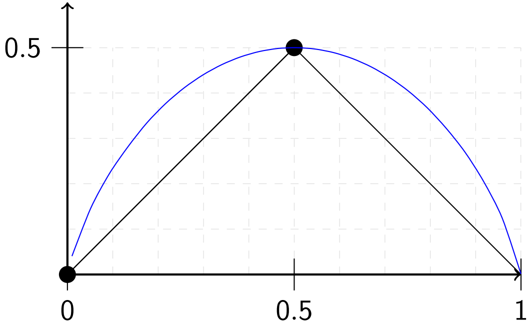
Co się stało? Interpretacja geometryczna



Przed podziałem mamy $p = 0.2$, a po potencjalnym podziale po x_1 uzyskano podzbiory o $p_1 = 0.5$ i $p_2 = 0$.

- Nie ma zysku!
- Zysk uzyskujemy tylko wówczas gdy podział dzieli podzbiór na obszary zdominowane przez dwie różne klasy.

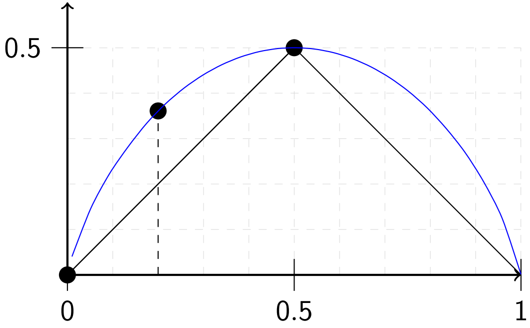
Rozwiązanie problemu: inna funkcja oceny



W praktyce zwykle stosujemy funkcje osiągające maksimum w tym samym punkcie co błąd klasyfikacji, ale które są wklęsłe.

Uwaga: na wykresie funkcja entropii została pomnożona przez 0.5, aby uzyskać tę samą skalę.

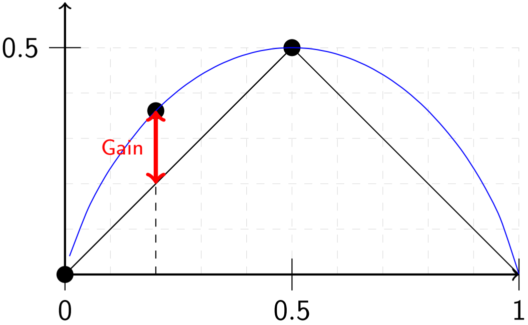
Rozwiązanie problemu: inna funkcja oceny



W praktyce zwykle stosujemy funkcje osiągające maksimum w tym samym punkcie co błąd klasyfikacji, ale które są wklęsłe.

Uwaga: na wykresie funkcja entropii została pomnożona przez 0.5, aby uzyskać tę samą skalę.

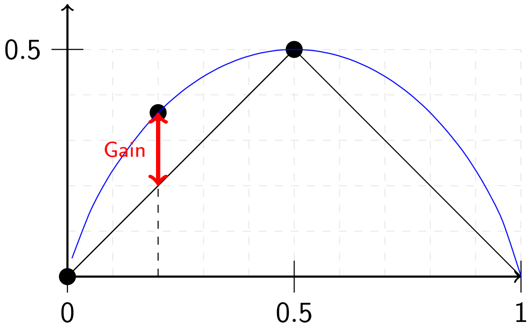
Rozwiązanie problemu: inna funkcja oceny



W praktyce zwykle stosujemy funkcje osiągające maksimum w tym samym punkcie co błąd klasyfikacji, ale które są wklęsłe.

Uwaga: na wykresie funkcja entropii została pomnożona przez 0.5, aby uzyskać tę samą skalę.

Rozwiązanie problemu: inna funkcja oceny



W praktyce zwykle stosujemy funkcje osiągające maksimum w tym samym punkcie co błąd klasyfikacji, ale które są wklęsłe.

Przykładem takiej funkcji jest entropia warunkowa:

$$\epsilon_{new} = x\epsilon_1 + (1 - x)\epsilon_2$$

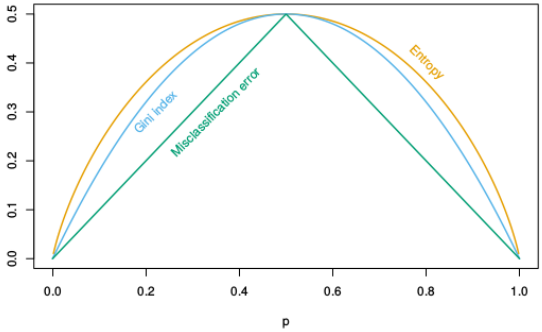
$$H_{new} = xH(p_1) + (1 - x)H(p_2)$$

gdzie:

$$H(p) = -p\log_2 p - (1 - p)\log_2(1 - p)$$

Uwaga: na wykresie funkcja entropii została pomnożona przez 0.5, aby uzyskać tę samą skalę.

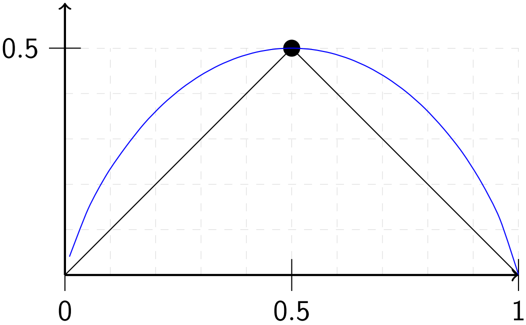
Porównanie używanych funkcji celu [Hastie et al.]

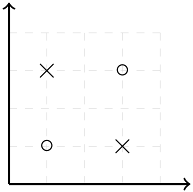


Oczywiście bardzo dużo innych możliwości!

- entropia warunkowa
- gini index $2p(1-p)$
- pierwiastek z gini index (dobry dla niezrównoważonych klas)
- oparte na testach statystycznych (likelihood ratio χ^2 statistics)
- ... chociażby $-(p-0.5)^2$

Co nie oznacza, że wszystkie problemy zostały rozwiązane...





Na zakończenie

- Drzewa decyzyjne to jeden z najpopularniejszych algorytmów uczenia maszynowego
- Zespoły drzew najczęstszym algorytmem wygrywającym konkursy Kaggle
- Ciekawe zastosowania: Kinect

Problem

Jakie są zalety i wady stosowania drzew decyzyjnych?

Dziękuję za uwagę!

Fundusze Europejskie
Polska Cyfrowa

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego