8. Drzewa decyzyjne, bagging, boosting i lasy losowe

dr inż. Urszula Libal

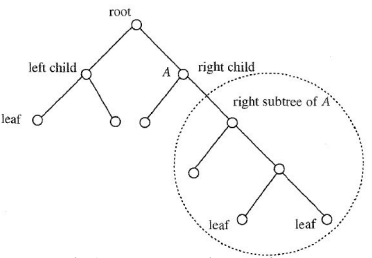
Politechnika Wrocławska

2015

1. Drzewa decyzyjne

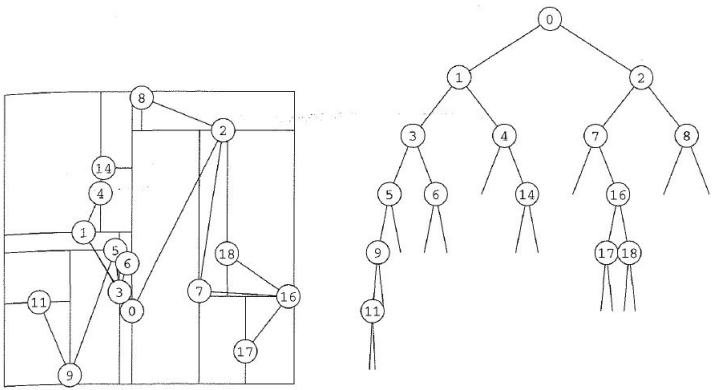
Drzewa decyzyjne (ang. *decision trees*), zwane również drzewami klasyfikacyjnymi (ang. *classification trees*), opierają się o drzewiastą strukturę. Klasyfikacja zaczyna się w **korzeniu** drzewa, a kończy po osiągnięciu jednej z klas terminalnych, czyli **liścia**. Sposób konstrukcji drzew zależy od metody podziału klas w węzłach.

Do znanych metod klasyfikacji w oparciu o drzewa decyzyjne należą CHAID, CART, C4.5 i QUEST.



Rysunek 1. Drzewo decyzyjne - zaznaczono korzeń i liście.

Źródło: [1]



Rysunek 2. Drzewo decyzyjne generuje podział przestrzeni cech na obszary decyzyjne.

Źródło: [1]

2. Binarne drzewa decyzyjne - podział w węźle

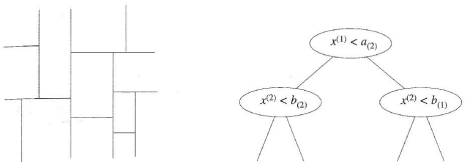
Rodzaje binarnych drzew decyzyjnych (wektor cech to $\mathbf{x} = (x_1, x_2, \ldots, x_D)$):

— **drzewa zwykłe** - podział w węźle na podstawie jednej cechy: $x_i \leq \alpha$

— **drzewa BSP** - podział binarny przestrzeni: $a_1 x_1 + \ldots + a_D x_D \leq \alpha$

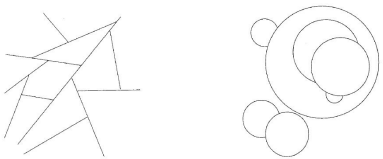
— **drzewa sferyczne**: $||\mathbf{x} - \mathbf{m}|| \leq \alpha$

W najprostszym przypadku podziały w węzłach drzewa mogą być podziałami jednowymiarowymi, czyli zależeć jedynie od jednej cechy (**drzewa zwykłe**). Jednak w przypadku wielowymiarowych problemów takie podejście może prowadzić do wielu podziałów i tym samym wielu węzłów. Rozwiązanie może stanowić użycie wielowymiarowych podziałów klas w węzłach (**drzewa BSP**, **drzewa sferyczne**).



Rysunek 3. Zwykłe drzewo decyzyjne oraz podział przestrzeni cech na obszary decyzyjne.

Źródło: [1]



a) b)

Rysunek 4. Podział przestrzeni cech za pomocą (a) drzewa BSP, (b) drzewa sferycznego.

Źródło: [1]

3. Transformacje drzew decyzyjnych

Transformacje drzew niebinarnych w drzewa binarne:

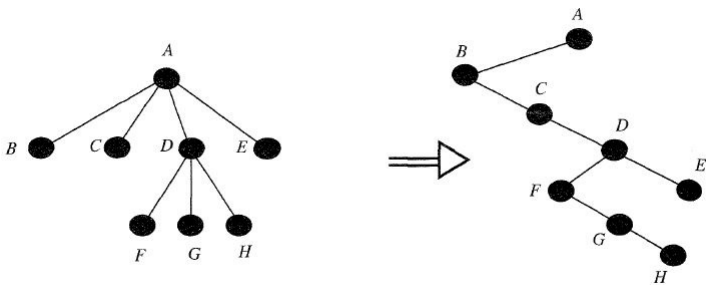
— drzewo binarne typu *najstarsze dziecko - kolejne dziecko*

(rys. 5)

— drzewo binarne oparte o *makroklasy* (zbiory klas)

(rys. 6)

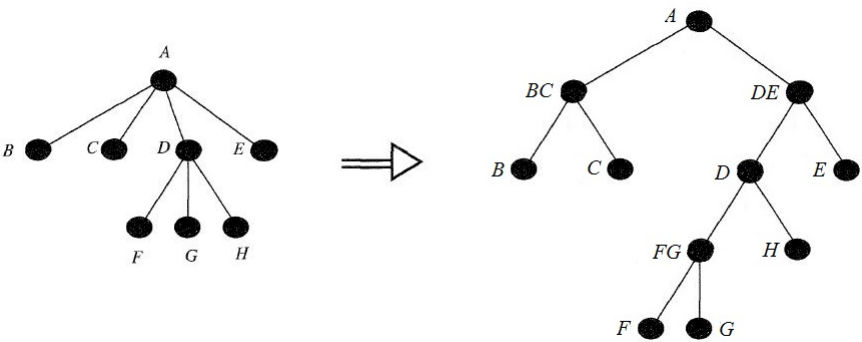
The oldest-child / next-sibling binary tree



Rysunek 5. Transformacja niebinarnego drzewa decyzyjnego w drzewo binarne.

Źródło: [1]

Łączenie klas w makroklasy



Rysunek 6. Transformacja niebinarnego drzewa decyzyjnego w drzewo binarne.

Źródło: opracowanie własne

4. Przycinanie drzew decyzyjnych

Aby zapobiec **przeuczeniu** klasyfikatora, czyli nadmiernemu dopasowaniu drzewa do danych z ciągu uczącego, stosuje się **przycinanie** (*pruning*).

— Zastępujemy poddrzewo liściem (patrz rys. 1).

— Skracamy drzewo, jednocześnie zmniejszając dokładność.

— Testujemy zastąpienie poddrzewa liściem na dodatkowym zbiorze danych - zbiorze przycinania.

5. Procedury wzmacniające klasyfikatory

Procedury agregujące rodziny klasyfikatorów (*ensemble method*) służą wzmacnianiu klasyfikatorów. Do tych metod zaliczamy:

— **algorytm bagging** - grupy klasyfikatorów (niekoniecznie drzew)

— **algorytm boosting** - grupy klasyfikatorów (niekoniecznie drzew)

— **lasy losowe** - grupy drzew decyzyjnych

6. Algorytm bagging

Bagging (*Bootstrap Aggregation*) należy do procedur agregujących rodzinę klasyfikatorów w jeden zbiorczy klasyfikator, który wynik klasyfikacji opiera na **głosowaniu większościowym**.

Rodzina L klasyfikatorów $\{\Psi_1, \Psi_2, \ldots, \Psi_L\}$ jest generowana na podstawie L ciągów uczących, utworzonych przez N-krotne losowanie ze zwracaniem (tzw. **próba bootstrapowa**) elementów ciągu uczącego $\{(\mathbf{x}_1, c_1), (\mathbf{x}_2, c_2), \ldots, , (\mathbf{x}_N, c_N)\}$ o długości N.

Wyuczone na ich podstawie klasyfikatory $\Psi_1, \Psi_2, \ldots, \Psi_L$ podejmują wspólną decyzję o klasyfikacji obrazu do jednej z klas, gdy dana klasa uzyska najwięcej wskazań.

7. Algorytm boosting

Boosting (po ang. *boosting* oznacza wzmocnienie) działa na podobnej zasadzie do baggingu, również generując rodzinę klasyfikatorów na podstawie N-elementowej pseudopróby (**próby bootstrapowej**) z ciągu uczącego $\{(\mathbf{x}_1, c_1), (\mathbf{x}_2, c_2), \ldots, (\mathbf{x}_N, c_N)\}$.

Podstawową procedują wzmacniającą klasyfikatory jest algorytm *AdaBoost* (od *Adaptive Boosting*). Polega on na sekwencyjnym losowaniu (ze zwracaniem) nowych ciągów uczących, które służą do wytrenowania kolejnych wersji klasyfikatorów. Procedura w każdym kroku konstruuje nowy klasyfikator w oparciu o wylosowany ciąg uczący, a następne przypisuje **wagi** wszystkim obrazom z oryginalnego ciągu uczącego $\{(\mathbf{x}_1, c_1), (\mathbf{x}_2, c_2), \ldots, (\mathbf{x}_N, c_N)\}$.

Wagi te wpływają na prawdopodobieństwo wylosowania danego obrazu do kolejnej wersji ciągu uczącego. W pierwszym kroku wagi wszystkich elementów ciągu uczącego są równe. W kolejnych krokach są zmieniane adaptacyjnie, w zależności od poprawności klasyfikacji elementów ciągu uczącego. Jeżeli dany obraz został błędnie zaklasyfikowany, to jego waga wzrasta, a w przeciwnym wypadku maleje. Dzięki takiemu działaniu obrazy błędnie zaklasyfikowane częściej ulegają wylosowaniu, co jest pożądane, ponieważ możliwe jest, że te elementy ciągu uczącego znajdują się najbliżej granicy między obszarami decyzyjnymi klas.

Użycie klasyfikatorów zbiorczych bagging i boosting wymaga znacznych nakładów obliczeniowych praktycznie wykluczających ich użycie w trybie *on-line*, tj. w przypadku ciągle napływających, nowych obrazów do klasyfikacji.

8. Lasy losowe

Lasy losowe (*random forests*) będące uogólnieniem idei drzew decyzyjnych zalicza się do procedur agregujących (*ensemble method*). Działanie lasów losowych polega na klasyfikacji za pomocą grupy drzew decyzyjnych. Końcowa decyzja jest podejmowana w wyniku **głosowania większościowego** nad klasami wskazanymi przez poszczególne drzewa decyzyjne.

Natomiast każde z drzew jest konstruowane w oparciu o **próbę bootstrapową**, powstałą przez wylosowanie ze zwracaniem N obiektów z ciągu uczącego o liczności N. W każdym węźle podział jest dokonywany jedynie na podstawie k losowo wybranych cech. Ich liczba jest znacznie mniejsza od p, czyli liczby wszystkich cech ($k \ll p$). Dzięki tej własności lasy losowe mogą być stosowane w problemach o olbrzymiej liczbie cech.

Literatura

[1] L. Devroye, L. Györfi, G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag, New York (1996)

[2] A.R. Webb, K.D. Copsey, *Statistical Pattern Recognition*, 3rd ed., Wiley, (2011)

[3] M. Krzyśko, W. Wołyński, T. Górecki, M. Skorzybut, *Systemy uczące się. Rozpoznawanie wzorców, analiza skupień i redukcja wymiarowości*. WNT, Warszawa (2008)