Statystyczna Eksploracja Danych

Wykład 5 - zespoły klasyfikatorów: bagging, boosting, lasy losowe

dr inż. Julian Sienkiewicz

28 marca 2019

Ogólny opis zespołów (rodzin) klasyfikatorów

Ogólny opis zespołów (rodzin) klasyfikatorów

- zakładamy, że rozwiązujemy zagadnienie klasyfikacji pod nadzorem i ograniczamy się do $g = 2$ klas,

Ogólny opis zespołów (rodzin) klasyfikatorów

- zakładamy, że rozwiązujemy zagadnienie klasyfikacji pod nadzorem i ograniczamy się do $g = 2$ klas,
- w odróżnieniu od innych podejść, tym razem dysponujemy nie jednym, lecz **wieloma** np. M **klasyfikatorami**,

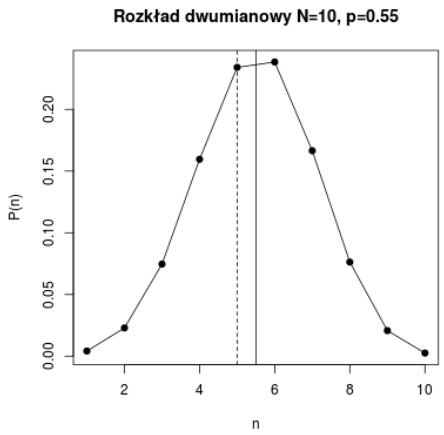
Ogólny opis zespołów (rodzin) klasyfikatorów

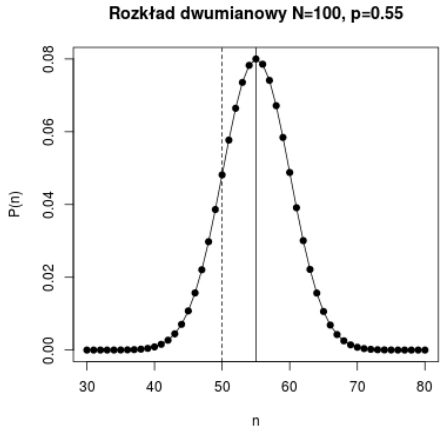
- zakładamy, że rozwiązujemy zagadnienie klasyfikacji pod nadzorem i ograniczamy się do $g = 2$ klas,
- w odróżnieniu od innych podejść, tym razem dysponujemy nie jednym, lecz **wieloma** np. M **klasyfikatorami**,
- klasyfikatory są statystycznie niezależne, a więc również dokonują one niezależnego przporządkowania do klas,

Ogólny opis zespołów (rodzin) klasyfikatorów

- zakładamy, że rozwiązujemy zagadnienie klasyfikacji pod nadzorem i ograniczamy się do $g = 2$ klas,
- w odróżnieniu od innych podejść, tym razem dysponujemy nie jednym, lecz **wieloma** np. M **klasyfikatorami**,
- klasyfikatory są statystycznie niezależne, a więc również dokonują one niezależnego przporządkowania do klas,
- poszczególne klasyfikatory charakteryzują się **słabą** skutecznością, niewiele większą od $\frac{1}{2}$ (np. 0.55) — są więc tzw. **słabymi uczniami** (ang. *weak lerners*).

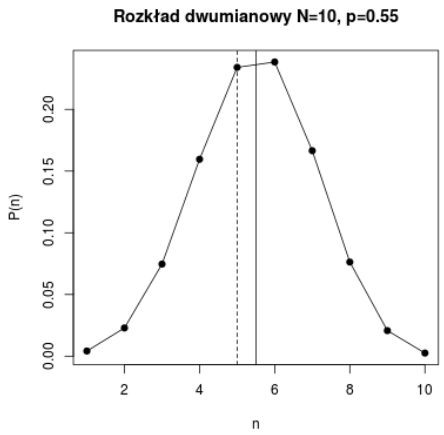
Ogólny opis

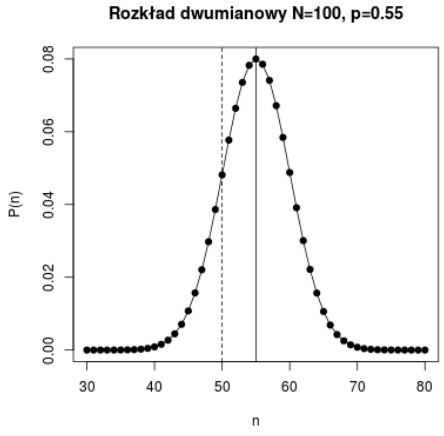




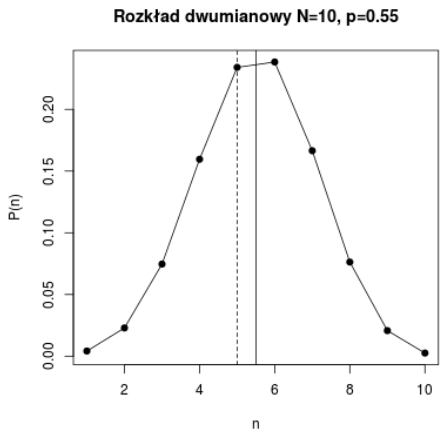
- prawdopodobieństwo podjęcia poprawnej decyzji w takim wypadku to $p = 0.55$,

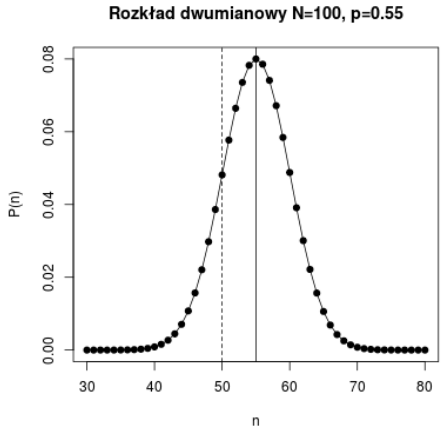
Ogólny opis



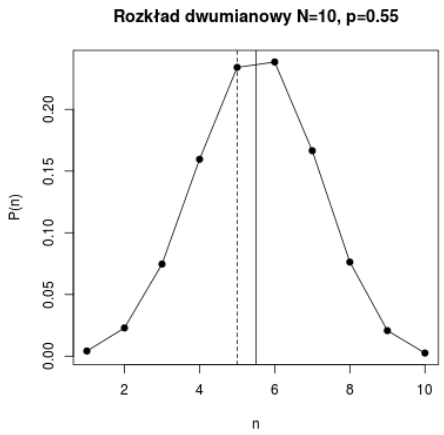


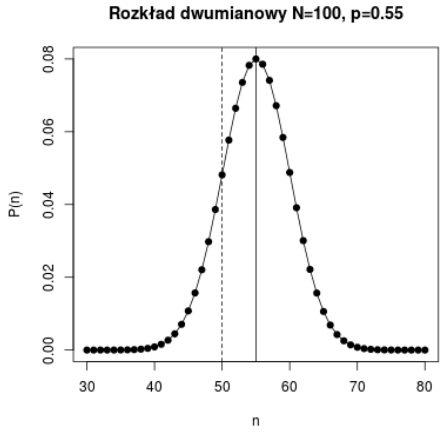
- prawdopodobieństwo podjęcia poprawnej decyzji w takim wypadku to $p = 0.55$,
- oczekiwana liczba poprawnych decyzji wśród **wszystkich** M klasyfikatorów to pM ,



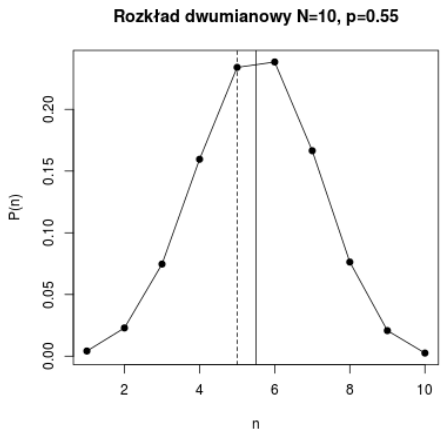


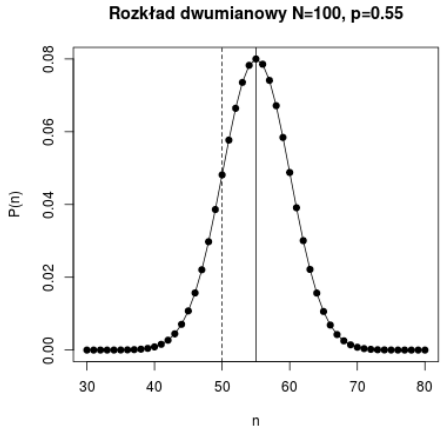
- prawdopodobieństwo podjęcia poprawnej decyzji w takim wypadku to $p = 0.55$,
- oczekiwana liczba poprawnych decyzji wśród **wszystkich** M klasyfikatorów to pM ,
- jest rozkład dwumianowy o wariancji $p(1-p)M = 0.2475M$ i odchyleniu standardowym $0.4975\sqrt{M}$,



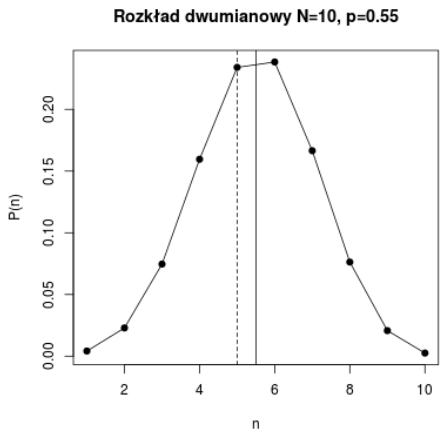


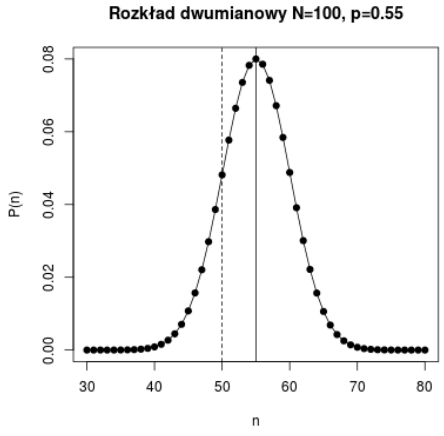
- prawdopodobieństwo podjęcia poprawnej decyzji w takim wypadku to $p = 0.55$,
- oczekiwana liczba poprawnych decyzji wśród **wszystkich** M klasyfikatorów to pM ,
- jest rozkład dwumianowy o wariancji $p(1-p)M = 0.2475M$ i odchyleniu standardowym $0.4975\sqrt{M}$,
- dla dużej liczby M możemy być prawie pewni, że większość klasyfikatorów dokona poprawnej decyzji,





- prawdopodobieństwo podjęcia poprawnej decyzji w takim wypadku to $p = 0.55$,
- oczekiwana liczba poprawnych decyzji wśród **wszystkich** M klasyfikatorów to pM ,
- jest rozkład dwumianowy o wariancji $p(1-p)M = 0.2475M$ i odchyleniu standardowym $0.4975\sqrt{M}$,
- dla dużej liczby M możemy być prawie pewni, że większość klasyfikatorów dokona poprawnej decyzji,
- czyli dla dużego M zaliczenie obserwacji do tej klasy, do której zaklasyfikowała ją większość dawałoby **poprawną decyzję**!!





- prawdopodobieństwo podjęcia poprawnej decyzji w takim wypadku to $p = 0.55$,
- oczekiwana liczba poprawnych decyzji wśród **wszystkich** M klasyfikatorów to pM ,
- jest rozkład dwumianowy o wariancji $p(1-p)M = 0.2475M$ i odchyleniu standardowym $0.4975\sqrt{M}$,
- dla dużej liczby M możemy być prawie pewni, że większość klasyfikatorów dokona poprawnej decyzji,
- czyli dla dużego M zaliczenie obserwacji do tej klasy, do której zaklasyfikowała ją większość dawałoby **poprawną decyzję**!!
- w praktyce klasyfikatory są od siebie statystycznie zależne...

Leo Breiman, University of California, Berkley, 1928-2005 (na zdjęciu koło 30-tki) — znany z prac związanych z drzewami regresyjnymi i klasyfikacyjnymi, zespołami drzew oraz lasami losowymi

Leo Breiman, University of California, Berkley, 1928-2005 (na zdjęciu koło 30-tki) — znany z prac związanych z drzewami regresyjnymi i klasyfikacyjnymi, zespołami drzew oraz lasami losowymi

Jedna z pierwszych i najprostszych rodzin klasyfikatorów to zaproponowana w 1996 r. przez Leo Breimana rodzina oparta na **agregacji bootstrapowej** — metoda **bagging** (od ang. *bootstrap aggregation*):

Leo Breiman, University of California, Berkley, 1928-2005 (na zdjęciu koło 30-tki) — znany z prac związanych z drzewami regresyjnymi i klasyfikacyjnymi, zespołami drzew oraz lasami losowymi

Jedna z pierwszych i najprostszych rodzin klasyfikatorów to zaproponowana w 1996 r. przez Leo Breimana rodzina oparta na **agregacji bootstrapowej** — metoda **bagging** (od ang. *bootstrap aggregation*):

1. dla każdego $m = 1, ...M$ wylosuj pseudopróbę P_m z oryginalnej N-elementowej próby uczącej,

Leo Breiman, University of California, Berkley, 1928-2005 (na zdjęciu koło 30-tki) — znany z prac związanych z drzewami regresyjnymi i klasyfikacyjnymi, zespołami drzew oraz lasami losowymi

Jedna z pierwszych i najprostszych rodzin klasyfikatorów to zaproponowana w 1996 r. przez Leo Breimana rodzina oparta na **agregacji bootstrapowej** — metoda **bagging** (od ang. *bootstrap aggregation*):

1. dla każdego $m = 1, ...M$ wylosuj pseudopróbę P_m z oryginalnej N-elementowej próby uczącej,
2. na każdej pseudopróbie P_m naucz klasyfikator (drzewo) T_m,

Leo Breiman, University of California, Berkley, 1928-2005 (na zdjęciu koło 30-tki) — znany z prac związanych z drzewami regresyjnymi i klasyfikacyjnymi, zespołami drzew oraz lasami losowymi

Jedna z pierwszych i najprostszych rodzin klasyfikatorów to zaproponowana w 1996 r. przez Leo Breimana rodzina oparta na **agregacji bootstrapowej** — metoda **bagging** (od ang. *bootstrap aggregation*):

1. dla każdego $m = 1, ...M$ wylosuj pseudopróbę P_m z oryginalnej N-elementowej próby uczącej,
2. na każdej pseudopróbie P_m naucz klasyfikator (drzewo) T_m,

W efekcie posiadamy rodzinę (lub zespół, ang. *ensemble*) M drzew, każde wyuczone na oddzielnej N-elementowej próbie powstałej poprzez metodę repróbkowania (bootstrap).

Leo Breiman, University of California, Berkley, 1928-2005 (na zdjęciu koło 30-tki) — znany z prac związanych z drzewami regresyjnymi i klasyfikacyjnymi, zespołami drzew oraz lasami losowymi

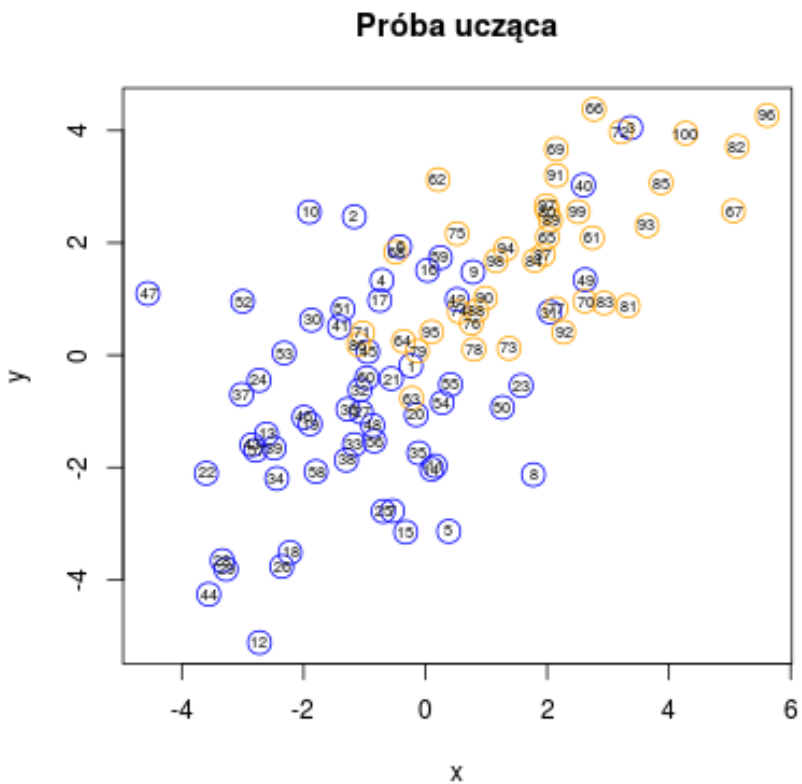
Jedna z pierwszych i najprostszych rodzin klasyfikatorów to zaproponowana w 1996 r. przez Leo Breimana rodzina oparta na **agregacji bootstrapowej** — metoda **bagging** (od ang. *bootstrap aggregation*):

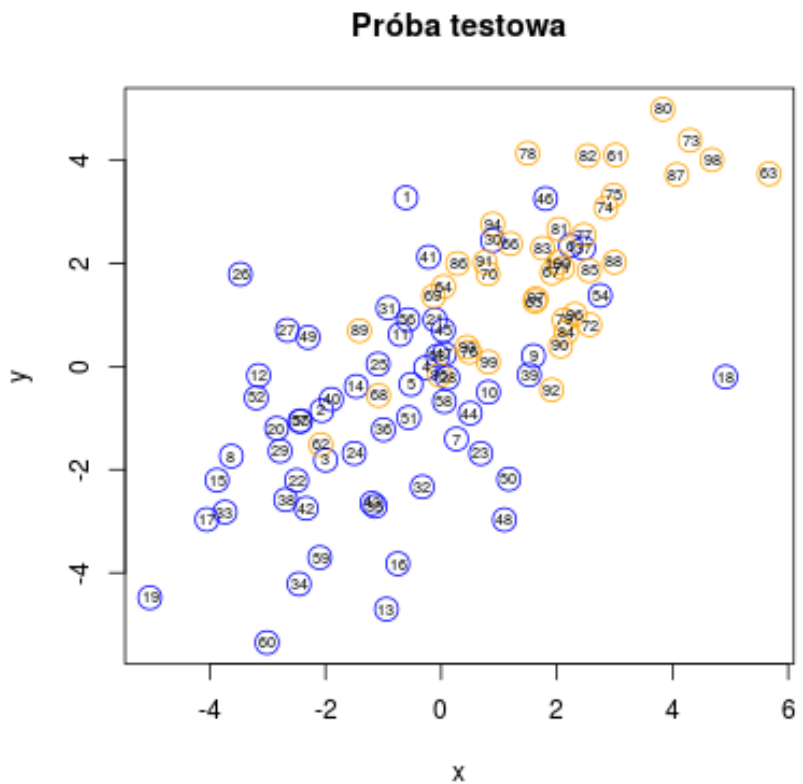
1. dla każdego $m = 1, ...M$ wylosuj pseudopróbę P_m z oryginalnej N-elementowej próby uczącej,
2. na każdej pseudopróbie P_m naucz klasyfikator (drzewo) T_m,

W efekcie posiadamy rodzinę (lub zespół, ang. *ensemble*) M drzew, każde wyuczone na oddzielnej N-elementowej próbie powstałej poprzez metodę repróbkowania (bootstrap).

Aby zaklasyfikować nową obserwację sprawdzamy jaka jest odpowiedź **każdego** z drzew i wybieramy taką klasę, która została przypisana przez **większość** drzew.

W poniższym przykładzie generujemy dwuwymiarowe rozkłady Gaussa o średnich $\mathbf{m}_1 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$, $\mathbf{m}_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$ oraz macierzach kowariancji $\mathbf{S}_1 = \begin{pmatrix} 4 & 2 \\ 2 & 4 \end{pmatrix}$, $\mathbf{S}_2 = \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix}$ i rozmiarach $n_1 = 60$, $n_2 = 40$.

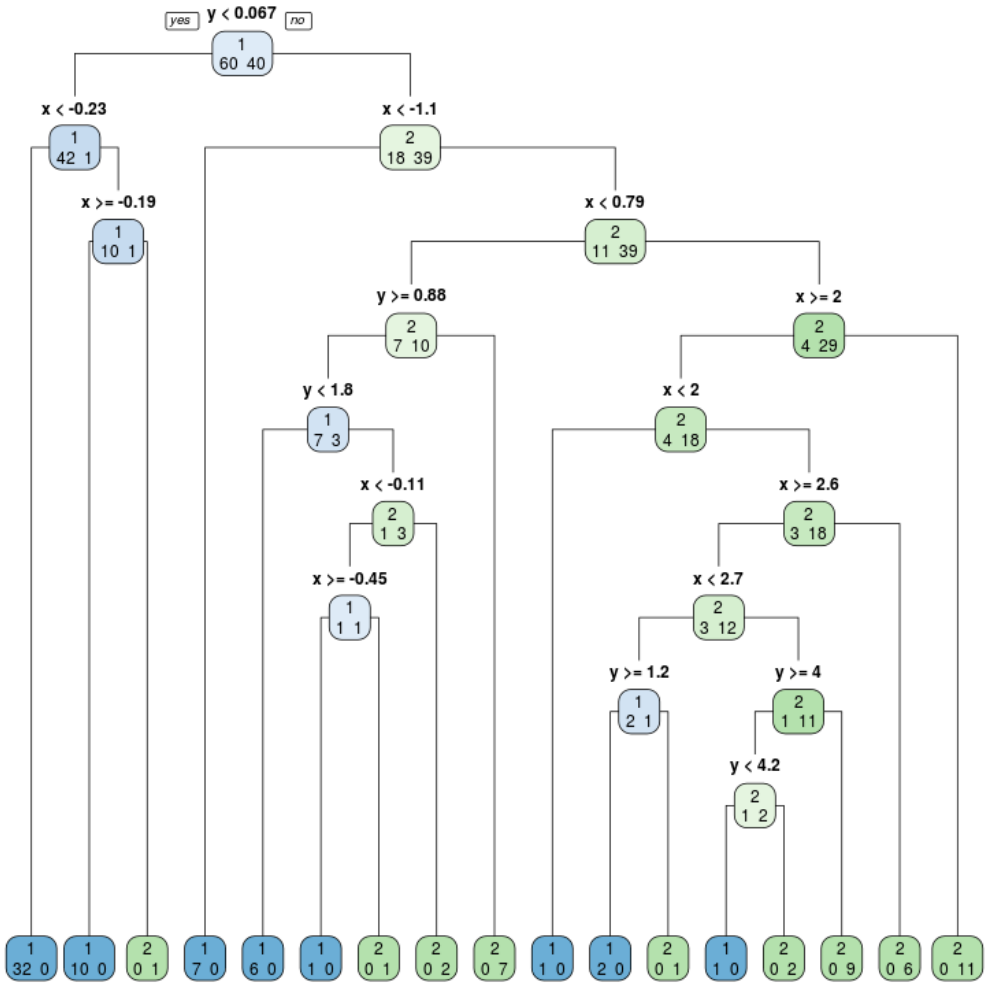




Bagging - przykład

Pojedyncze drzewo wyuczone na całej próbie składa się z 17 węzłów.

Pojedyncze drzewo wyuczone na całej próbie składa się z 17 węzłów.

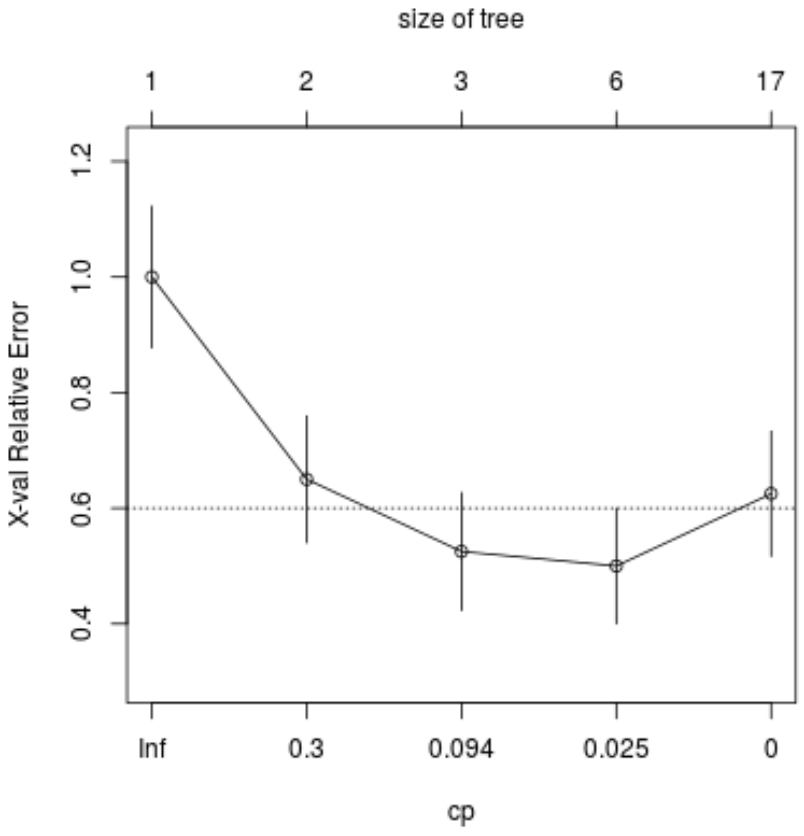


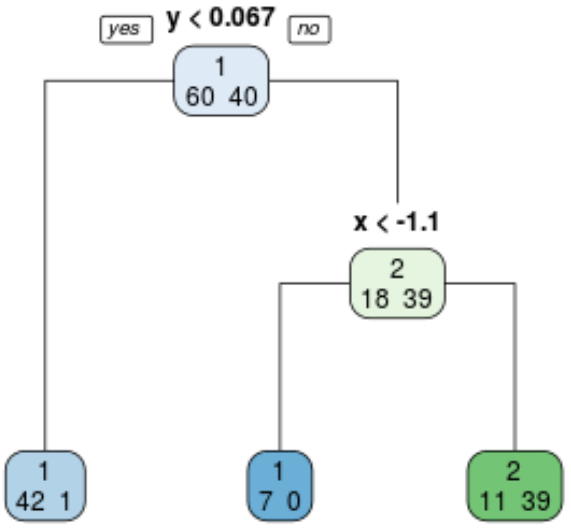
Bagging - przykład

Sprawdzamy jak wygląda kryterium kosztu-złożoności i przycinamy drzewo.

Bagging - przykład

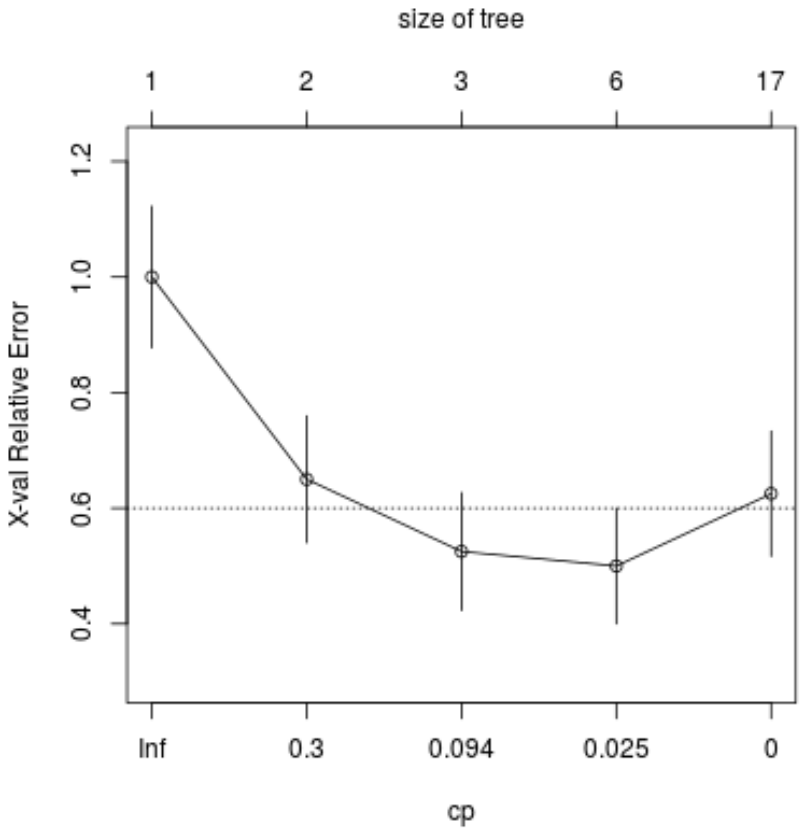
Sprawdzamy jak wygląda kryterium kosztu-złożoności i przycinamy drzewo.

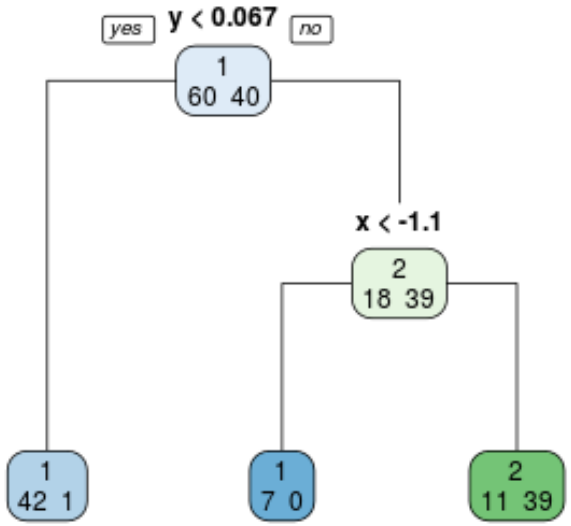




Bagging - przykład

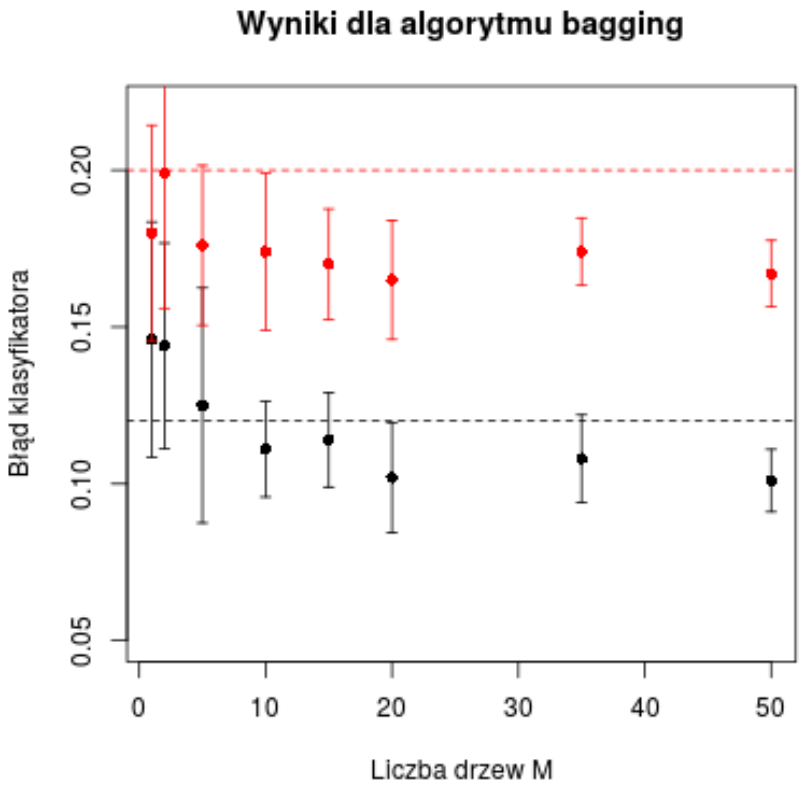
Sprawdzamy jak wygląda kryterium kosztu-złożoności i przycinamy drzewo.



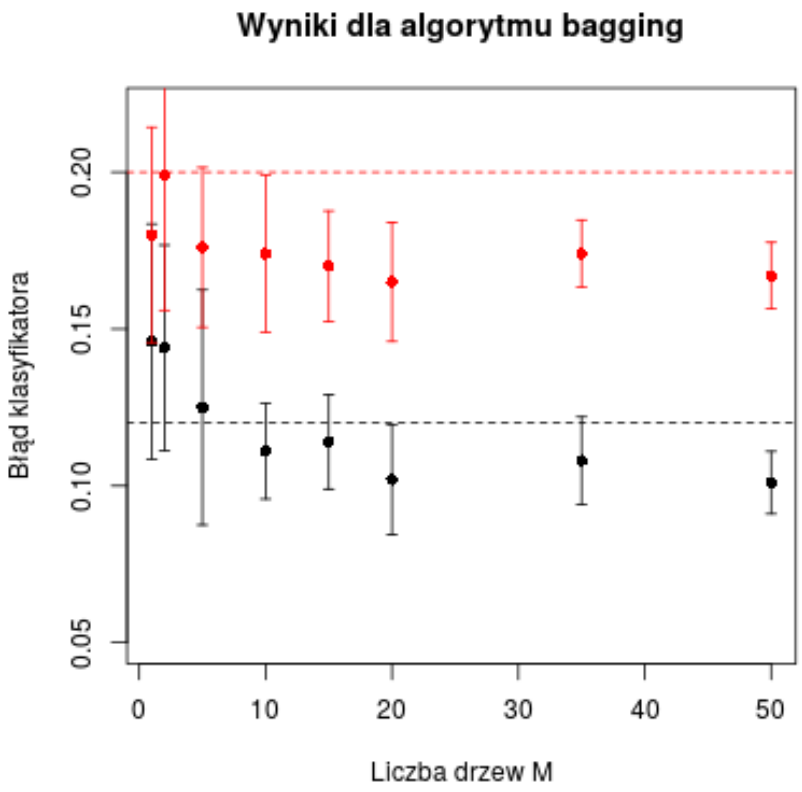


Skuteczność na przyciętym drzewie dla PU wynosi **0.88**, dla PT - **0.80**.

Bagging - przykład

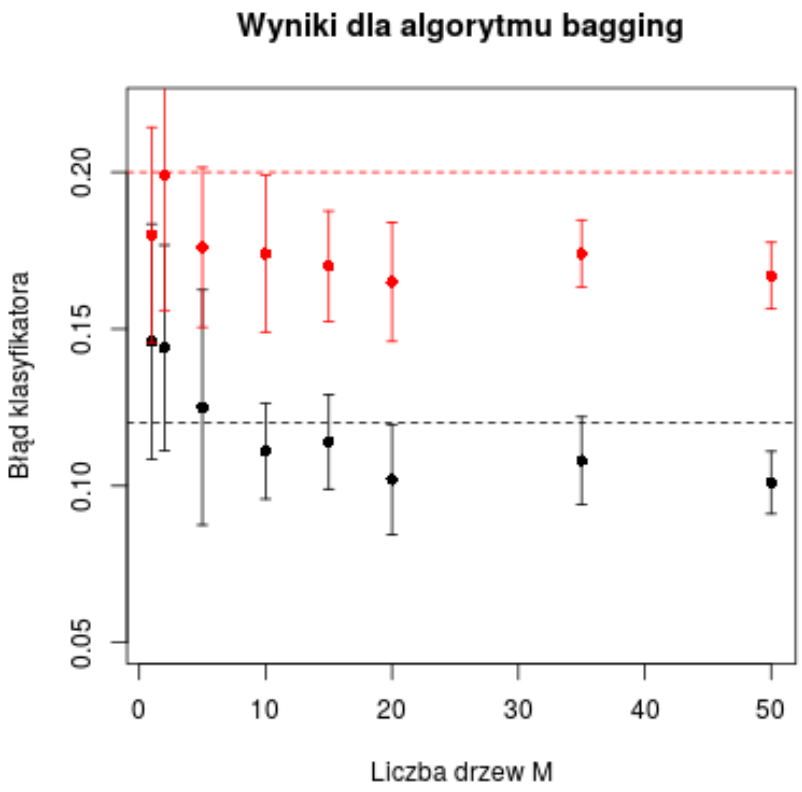


Bagging - przykład



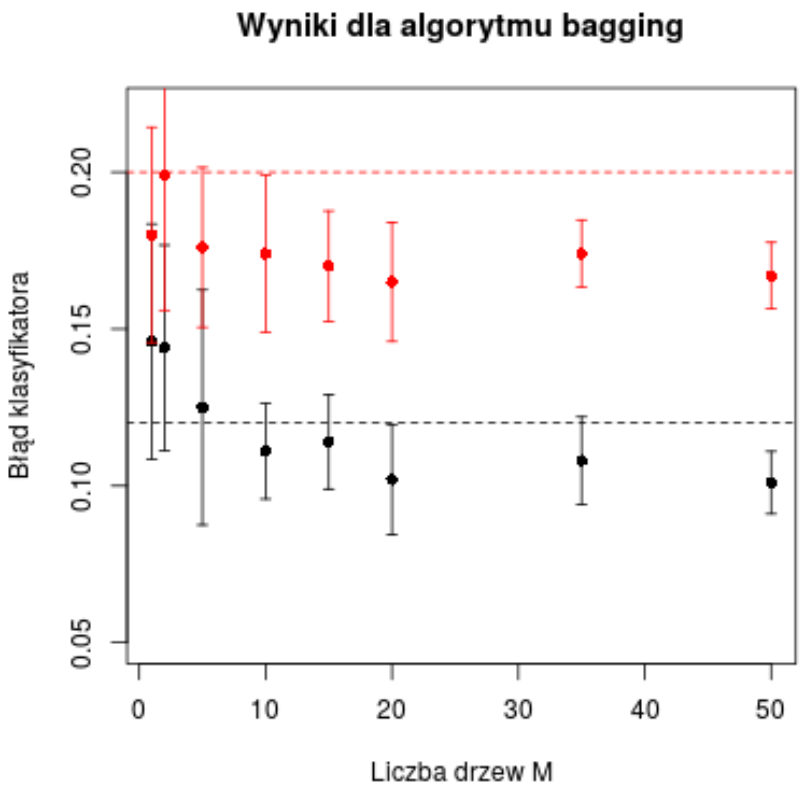
Uruchamiamy algorytm bagging dla naszego przykładu:

Bagging - przykład



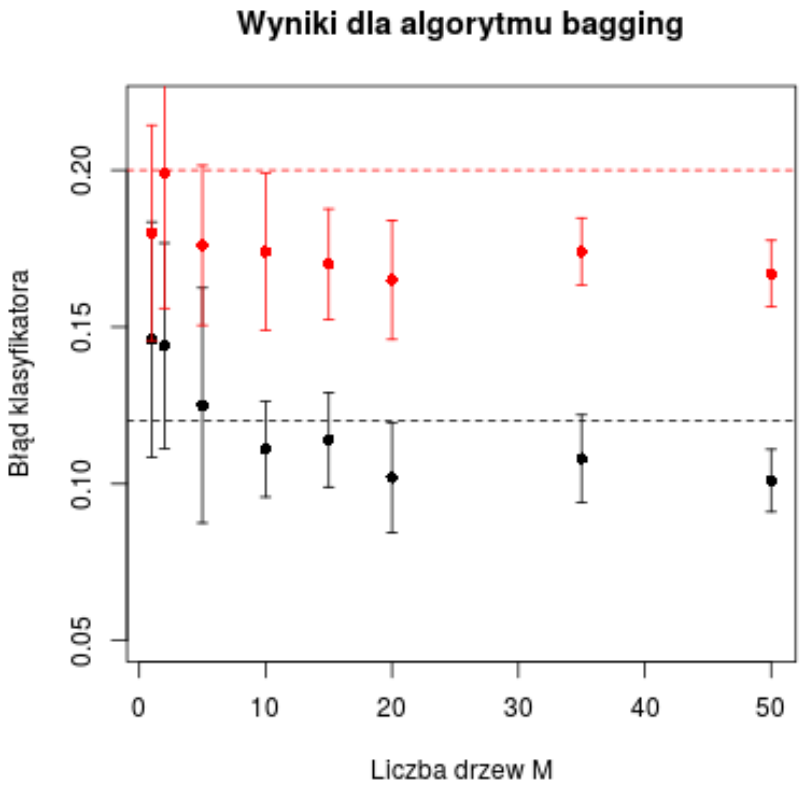
Uruchamiamy algorytm bagging dla naszego przykładu:

- wraz ze wzrostem liczby drzew błąd klasyfikatora spada (czarne - PU, czerwone - PT),



Uruchamiamy algorytm bagging dla naszego przykładu:

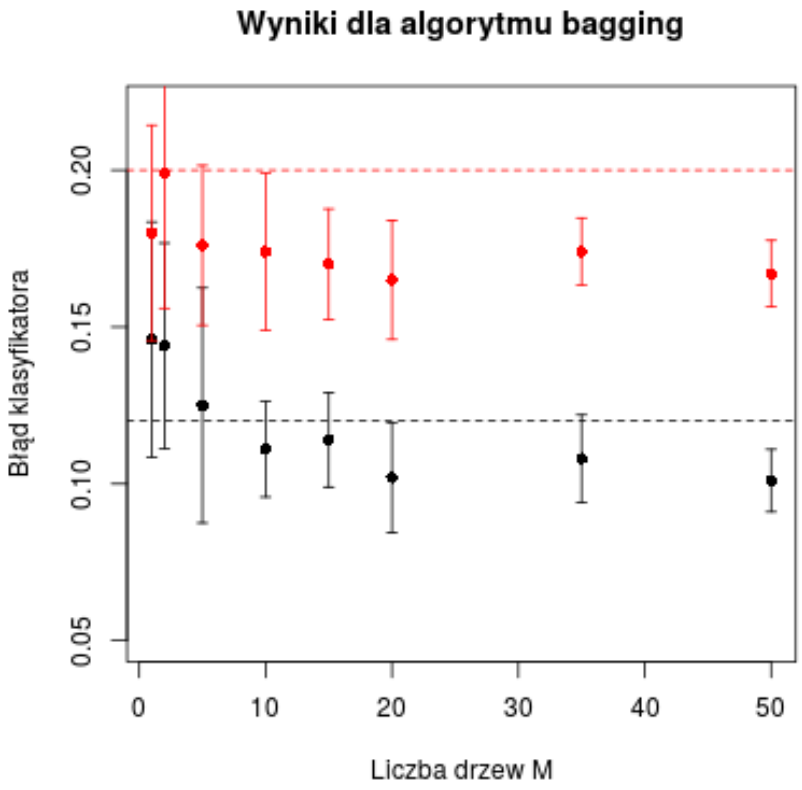
- wraz ze wzrostem liczby drzew błąd klasyfikatora spada (czarne - PU, czerwone - PT),
- dla około $M = 50$ wartość błędu nasyca się,



Uruchamiamy algorytm bagging dla naszego przykładu:

- wraz ze wzrostem liczby drzew błąd klasyfikatora spada (czarne - PU, czerwone - PT),
- dla około $M = 50$ wartość błędu nasyca się,
- dla większej ilości drzew otrzymujemy coraz niższe wartości odchylenia,

Bagging - przykład



Uruchamiamy algorytm bagging dla naszego przykładu:

- wraz ze wzrostem liczby drzew błąd klasyfikatora spada (czarne - PU, czerwone - PT),
- dla około $M = 50$ wartość błędu nasyca się,
- dla większej ilości drzew otrzymujemy coraz niższe wartości odchylenia,
- dla dużych M zarówno w przypadku PU jak i PT wartości są niższe niż dla pojedynczego, optymalnego drzewa.

Yoav Freund

Robert Schapire

Pewnym rodzajem ulepszenia metody bagging jest **boosting** (1997 r. Freunda i Schapire). Jest ona niezależna od metody bagging, ale oba podejścia noszą pewne cechy wspólne:

Yoav Freund

Robert Schapire

Pewnym rodzajem ulepszenia metody bagging jest **boosting** (1997 r. Freunda i Schapire). Jest ona niezależna od metody bagging, ale oba podejścia noszą pewne cechy wspólne:

- w bagging wszystkie pseudopróby powstają poprzez losowanie ze zwracaniem, zgodnie z rozkładem jednostajnym,

Yoav Freund

Robert Schapire

Pewnym rodzajem ulepszenia metody bagging jest **boosting** (1997 r. Freunda i Schapire). Jest ona niezależna od metody bagging, ale oba podejścia noszą pewne cechy wspólne:

- w bagging wszystkie pseudopróby powstają poprzez losowanie ze zwracaniem, zgodnie z rozkładem jednostajnym,
- w przypadku boosting również możliwe jest losowanie ze zwracaniem, ale rozkład prawdopodobieństwa **zmienia się** z pseduopróby na pseudopróbę,

Yoav Freund

Robert Schapire

Pewnym rodzajem ulepszenia metody bagging jest **boosting** (1997 r. Freunda i Schapire). Jest ona niezależna od metody bagging, ale oba podejścia noszą pewne cechy wspólne:

- w bagging wszystkie pseudopróby powstają poprzez losowanie ze zwracaniem, zgodnie z rozkładem jednostajnym,
- w przypadku boosting również możliwe jest losowanie ze zwracaniem, ale rozkład prawdopodobieństwa **zmienia się** z pseduopróby na pseudopróbę,
- po wylosowaniu każdej pseudopróby zostaje na jej podstawie skonstruowany klasyfikator

Yoav Freund

Robert Schapire

Pewnym rodzajem ulepszenia metody bagging jest **boosting** (1997 r. Freunda i Schapire). Jest ona niezależna od metody bagging, ale oba podejścia noszą pewne cechy wspólne:

- w bagging wszystkie pseudopróby powstają poprzez losowanie ze zwracaniem, zgodnie z rozkładem jednostajnym,
- w przypadku boosting również możliwe jest losowanie ze zwracaniem, ale rozkład prawdopodobieństwa **zmienia się** z pseduopróby na pseudopróbę,
- po wylosowaniu każdej pseudopróby zostaje na jej podstawie skonstruowany klasyfikator
- krok procedury kończy się sprawdzeniem jakości klasyfikatora

Yoav Freund

Robert Schapire

Pewnym rodzajem ulepszenia metody bagging jest **boosting** (1997 r. Freunda i Schapire). Jest ona niezależna od metody bagging, ale oba podejścia noszą pewne cechy wspólne:

- w bagging wszystkie pseudopróby powstają poprzez losowanie ze zwracaniem, zgodnie z rozkładem jednostajnym,
- w przypadku boosting również możliwe jest losowanie ze zwracaniem, ale rozkład prawdopodobieństwa **zmienia się** z pseduopróby na pseudopróbę,
- po wylosowaniu każdej pseudopróby zostaje na jej podstawie skonstruowany klasyfikator
- krok procedury kończy się sprawdzeniem jakości klasyfikatora

Yoav Freund

Robert Schapire

Podstawowe cechy boostingu:

- Losowanie elementów do pierwszej pseudopróby odbywa się tak samo jak w metodzie bagging, czyli z rozkładu jednostajnego.

Yoav Freund

Robert Schapire

Podstawowe cechy boostingu:

- Losowanie elementów do pierwszej pseudopróby odbywa się tak samo jak w metodzie bagging, czyli z rozkładu jednostajnego.
- Począwszy od drugiego korku, rozkład prawdopodobieństwa jest **adaptacyjnie zmieniany**.

Yoav Freund

Robert Schapire

Podstawowe cechy boostingu:

- Losowanie elementów do pierwszej pseudopróby odbywa się tak samo jak w metodzie bagging, czyli z rozkładu jednostajnego.
- Począwszy od drugiego korku, rozkład prawdopodobieństwa jest **adaptacyjnie zmieniany**.
- Jeżeli w m-tym kroku procedury i-ta obserwacja była losowana z pewną wagą w_i i została źle zaklasyfikowana przez m-ty klasyfikator, to w kroku $m+1$ **prawdopodobieństwo** jej wylosowania zostaje **zwiększone** poprzez pomnożenie wartości w_i przez ustaloną liczbę, taką samą dla wszystkich źle zaklasyfikowanych obserwacji.

Yoav Freund

Robert Schapire

Podstawowe cechy boostingu:

- Losowanie elementów do pierwszej pseudopróby odbywa się tak samo jak w metodzie bagging, czyli z rozkładu jednostajnego.
- Począwszy od drugiego korku, rozkład prawdopodobieństwa jest **adaptacyjnie zmieniany**.
- Jeżeli w m-tym kroku procedury i-ta obserwacja była losowana z pewną wagą w_i i została źle zaklasyfikowana przez m-ty klasyfikator, to w kroku $m+1$ **prawdopodobieństwo** jej wylosowania zostaje **zwiększone** poprzez pomnożenie wartości w_i przez ustaloną liczbę, taką samą dla wszystkich źle zaklasyfikowanych obserwacji.
- Inaczej mówiąc obserwacje **źle zaklasyfikowane** przez m-ty klasyfikator mają **większą szansę** bycia wylosowanymi do kolejnej pseudopróby.

Yoav Freund

Robert Schapire

Podstawowe cechy boostingu:

- Losowanie elementów do pierwszej pseudopróby odbywa się tak samo jak w metodzie bagging, czyli z rozkładu jednostajnego.
- Począwszy od drugiego korku, rozkład prawdopodobieństwa jest **adaptacyjnie zmieniany**.
- Jeżeli w m-tym kroku procedury i-ta obserwacja była losowana z pewną wagą w_i i została źle zaklasyfikowana przez m-ty klasyfikator, to w kroku $m+1$ **prawdopodobieństwo** jej wylosowania zostaje **zwiększone** poprzez pomnożenie wartości w_i przez ustaloną liczbę, taką samą dla wszystkich źle zaklasyfikowanych obserwacji.
- Inaczej mówiąc obserwacje **źle zaklasyfikowane** przez m-ty klasyfikator mają **większą szansę** bycia wylosowanymi do kolejnej pseudopróby.

W praktyce odchodzi się od losowania pseudoprób i jeśli korzysta się z drzew decyzyjnych, możliwe jest wykonanie klasyfikatora opartego na "obserwacjach ważonych". Załóżmy, że posługujemy się drzewem i

- rozpatrujemy węzeł m o n_m elementach PU znajdujących się w obszarze R_m,
- estymatorem przynależności elementu do klasy 1 jest

$$p_{m1} = \frac{1}{n_m} \sum_{\mathbf{x}_i \in R_m} \mathrm{I}(y_i = -1) = \frac{\sum_{\mathbf{x}_i \in R_m} \mathrm{I}(y_i = -1)}{\sum_{i=1}^{N} \mathrm{I}(\mathbf{x}_i \in R_m)}$$

- jeżeli każdemu elementowi próby można przypisać pewne wagi w_i, $i = 1, ..., N$, to estymator przynależności należy zastąpić

$$p_{m1} = \frac{\sum_{\mathbf{x}_i \in R_m} w_i \mathrm{I}(y_i = -1)}{\sum_{i=1}^{N} w_i \mathrm{I}(\mathbf{x}_i \in R_m)}$$

Podstawowy algorytm **AdaBoost**

Zakładamy $g = 2$, liczność próby to N a klasy zapisane są jako $y_i = \{1, -1\}$:

Podstawowy algorytm **AdaBoost**

Zakładamy $g = 2$, liczność próby to N a klasy zapisane są jako $y_i = \{1, -1\}$:

1. Przyjmij wagi $w_i = \frac{1}{n}$, $i = 1, ..., N$.
2. Dla $m = 1, ..., M$:
   - wytrenuj klasyfikator $f_m(\mathbf{x}$, stosując do danych uczących wagi w_i,

Podstawowy algorytm **AdaBoost**

Zakładamy $g = 2$, liczność próby to N a klasy zapisane są jako $y_i = \{1, -1\}$:

1. Przyjmij wagi $w_i = \frac{1}{n}$, $i = 1, ..., N$.
2. Dla $m = 1, ..., M$:
   - wytrenuj klasyfikator $f_m(\mathbf{x}$, stosując do danych uczących wagi w_i,
   - oblicz

$$\text{err}_m = \sum_{i=1}^{N} w_i \text{I}[y_i \neq f_m(\mathbf{x}_i)] \qquad \gamma_m = \ln \frac{1-\text{err}_m}{\text{err}_m}$$

Podstawowy algorytm **AdaBoost**

Zakładamy $g = 2$, liczność próby to N a klasy zapisane są jako $y_i = \{1, -1\}$:

1. Przyjmij wagi $w_i = \frac{1}{n}$, $i = 1, ..., N$.
2. Dla $m = 1, ..., M$:
   - wytrenuj klasyfikator $f_m(\mathbf{x}$, stosując do danych uczących wagi w_i,
   - oblicz

$$\text{err}_m = \sum_{i=1}^{N} w_i \text{I}[y_i \neq f_m(\mathbf{x}_i)] \qquad \gamma_m = \ln \frac{1-\text{err}_m}{\text{err}_m}$$

   - podstaw,

Podstawowy algorytm **AdaBoost**

Zakładamy $g = 2$, liczność próby to N a klasy zapisane są jako $y_i = \{1, -1\}$:

1. Przyjmij wagi $w_i = \frac{1}{n}$, $i = 1, ..., N$.
2. Dla $m = 1, ..., M$:
   - wytrenuj klasyfikator $f_m(\mathbf{x}$, stosując do danych uczących wagi w_i,
   - oblicz

$$\text{err}_m = \sum_{i=1}^{N} w_i \text{I}[y_i \neq f_m(\mathbf{x}_i)] \qquad \gamma_m = \ln \frac{1-\text{err}_m}{\text{err}_m}$$

   - podstaw,

$$w_i = w_i \exp(\gamma_m \text{I}[y_i \neq f_m(\mathbf{x}_i)]) \qquad i = 1, ..., N$$

   i dokonaj renormalizacji tak, aby $\sum_i w_i = 1$.

Podstawowy algorytm **AdaBoost**

Zakładamy $g = 2$, liczność próby to N a klasy zapisane są jako $y_i = \{1, -1\}$:

1. Przyjmij wagi $w_i = \frac{1}{n}$, $i = 1, ..., N$.
2. Dla $m = 1, ..., M$:
   - wytrenuj klasyfikator $f_m(\mathbf{x}$, stosując do danych uczących wagi w_i,
   - oblicz

$$\text{err}_m = \sum_{i=1}^{N} w_i \text{I}[y_i \neq f_m(\mathbf{x}_i)] \qquad \gamma_m = \ln \frac{1-\text{err}_m}{\text{err}_m}$$

   - podstaw,

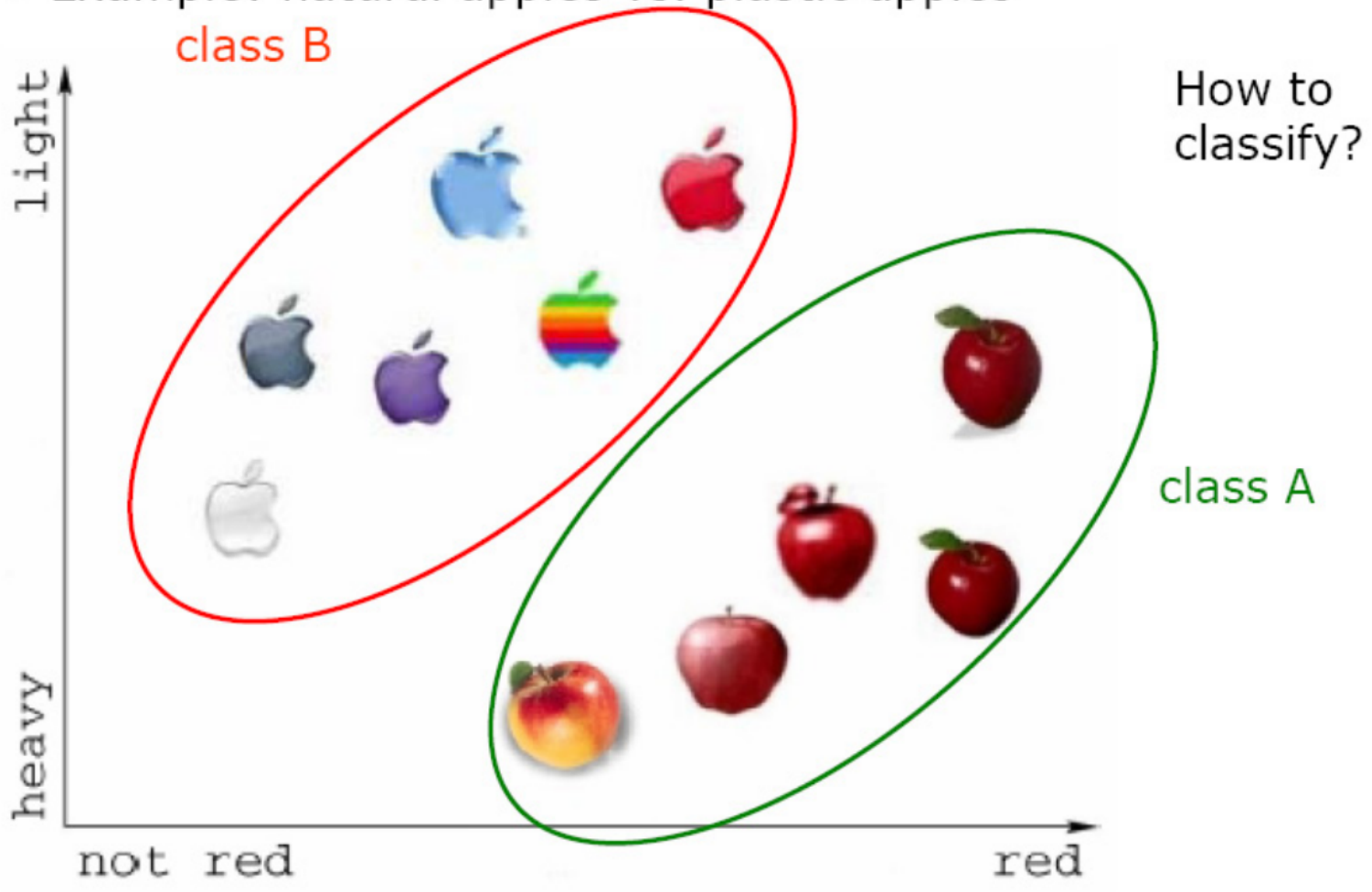
$$w_i = w_i \exp(\gamma_m \text{I}[y_i \neq f_m(\mathbf{x}_i)]) \qquad i = 1, ..., N$$

   i dokonaj renormalizacji tak, aby $\sum_i w_i = 1$.
3. Podaj

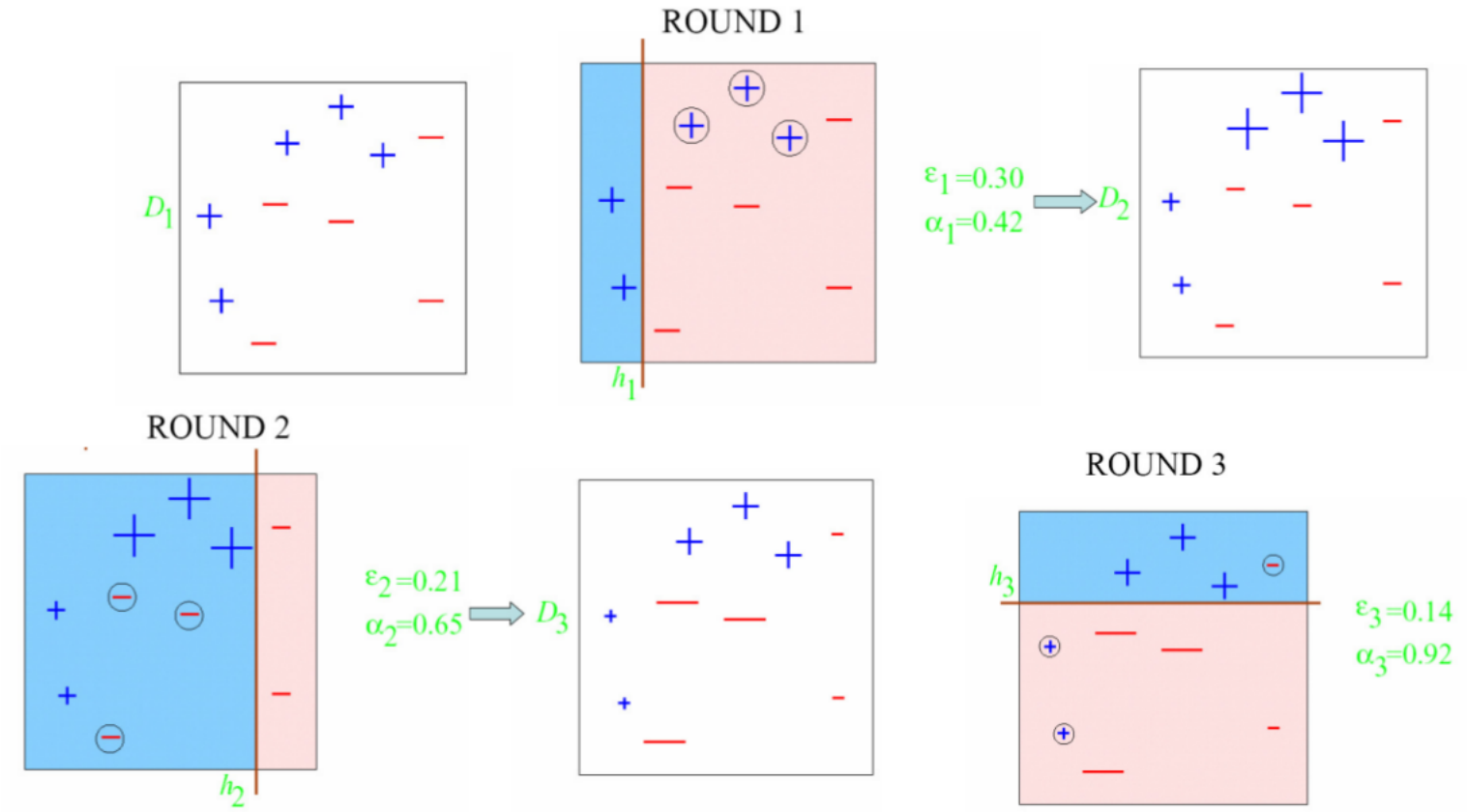
$$\text{sgn}\left[\sum_{m=1}^{M} \gamma_m f_m(\mathbf{x})\right]$$

Poglądowe przedstawienie działania algorytmu

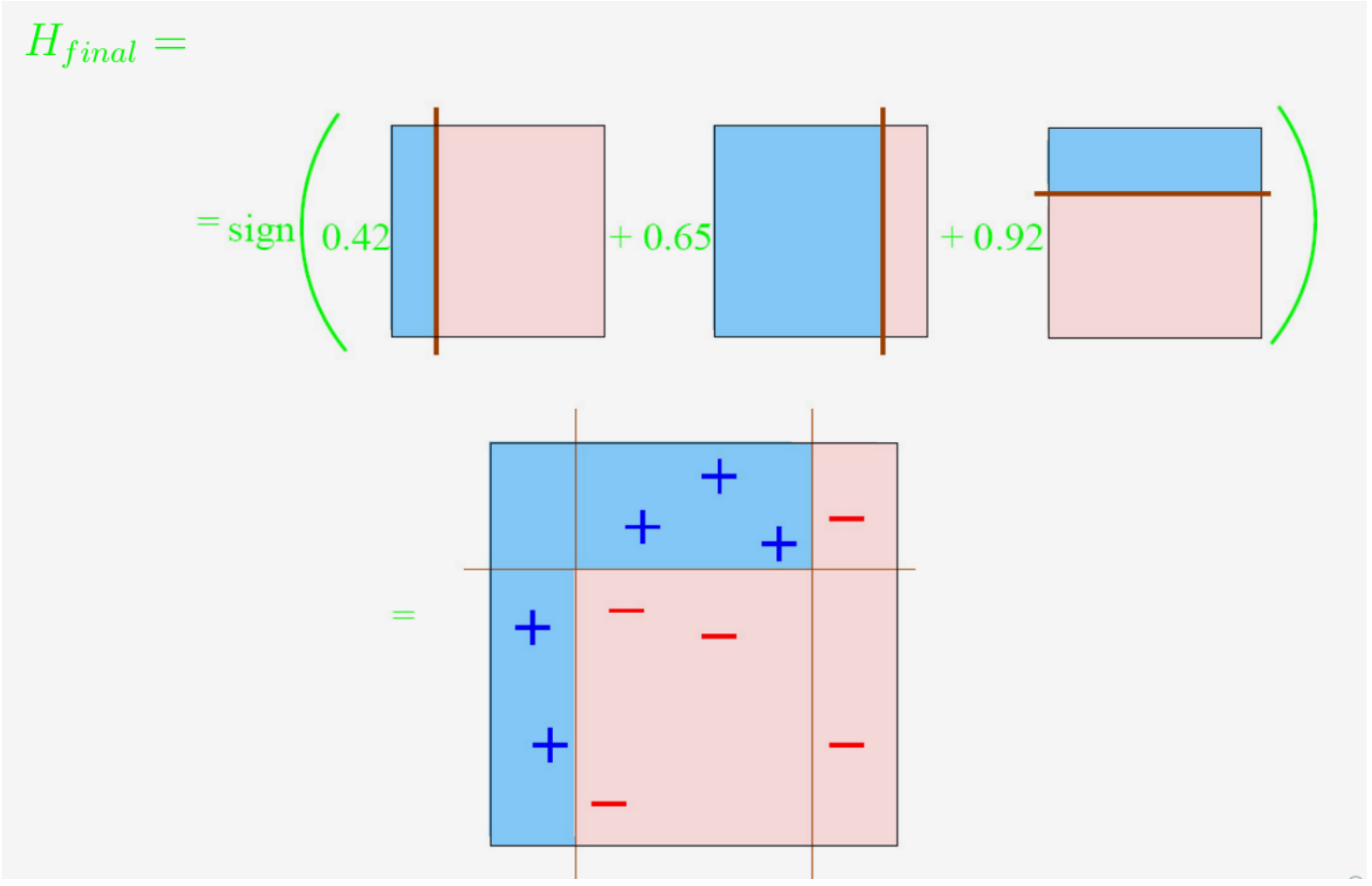
- Example: natural apples vs. plastic apples



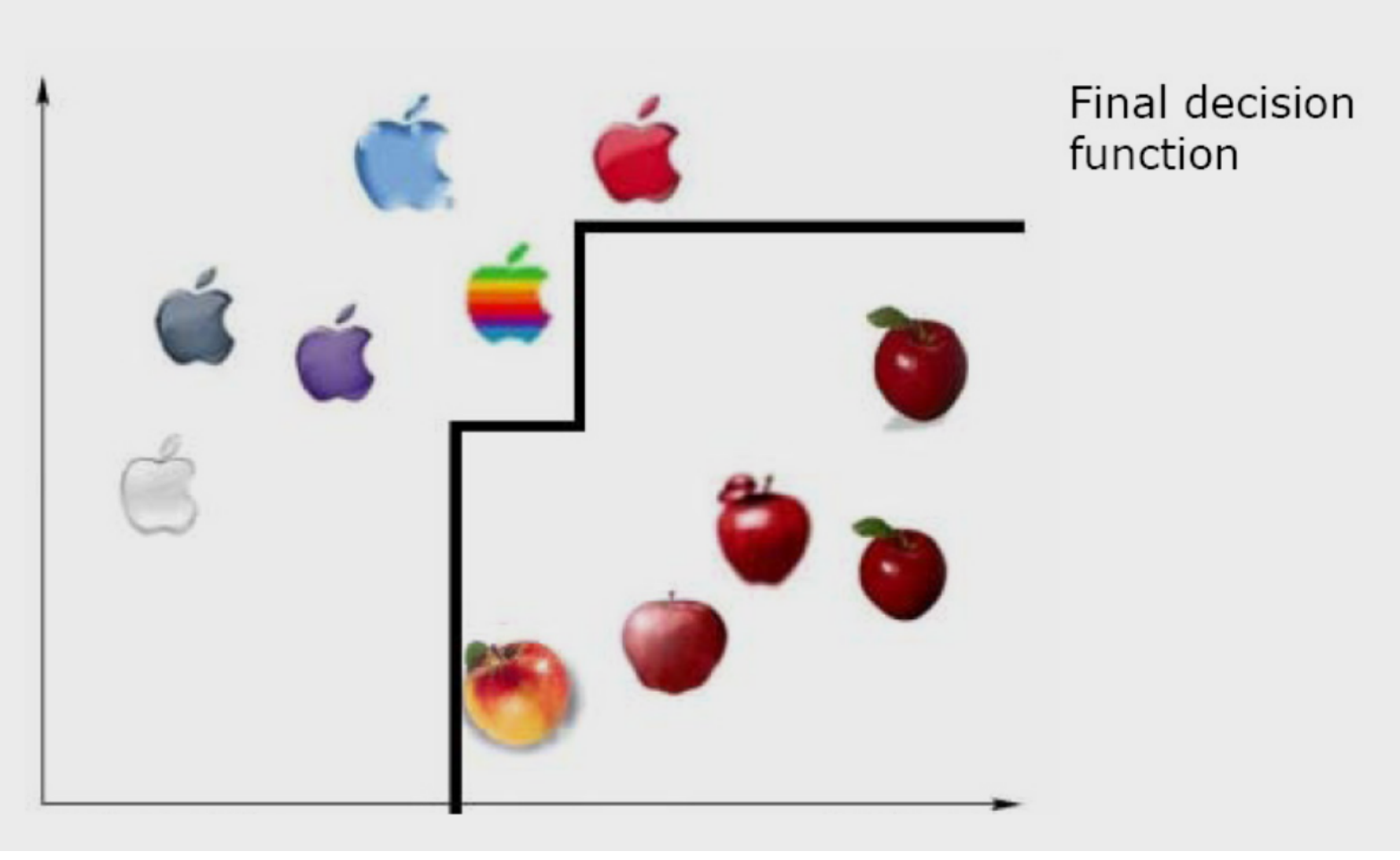
Poglądowe przedstawienie działania algorytmu



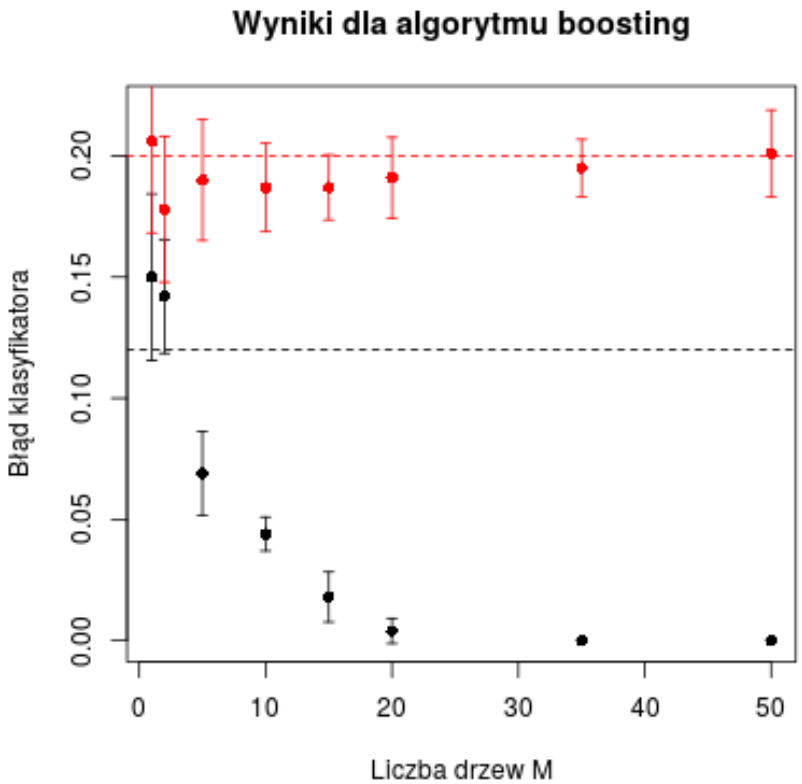
Poglądowe przedstawienie działania algorytmu

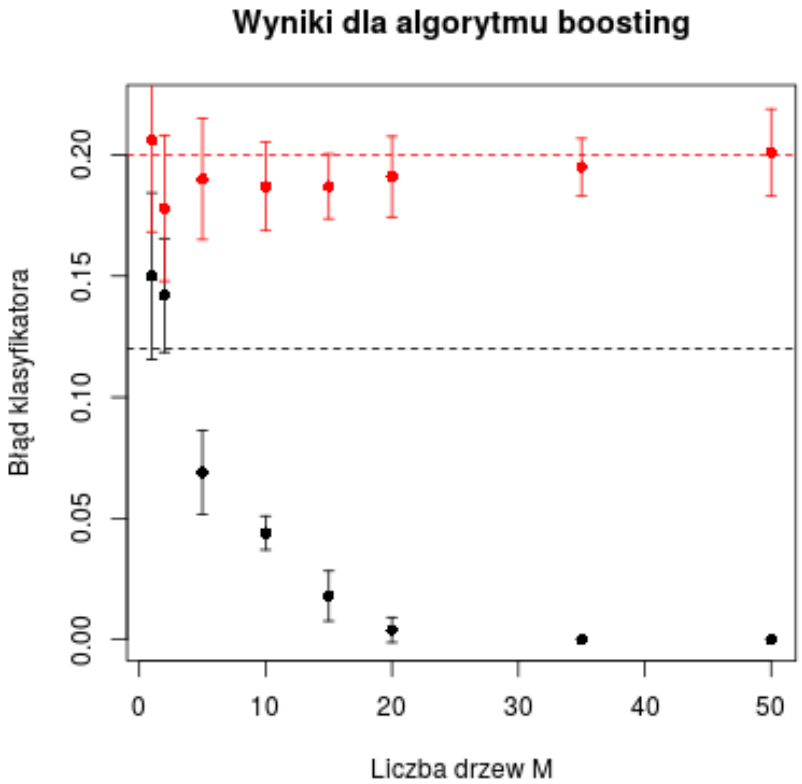


Poglądowe przedstawienie działania algorytmu

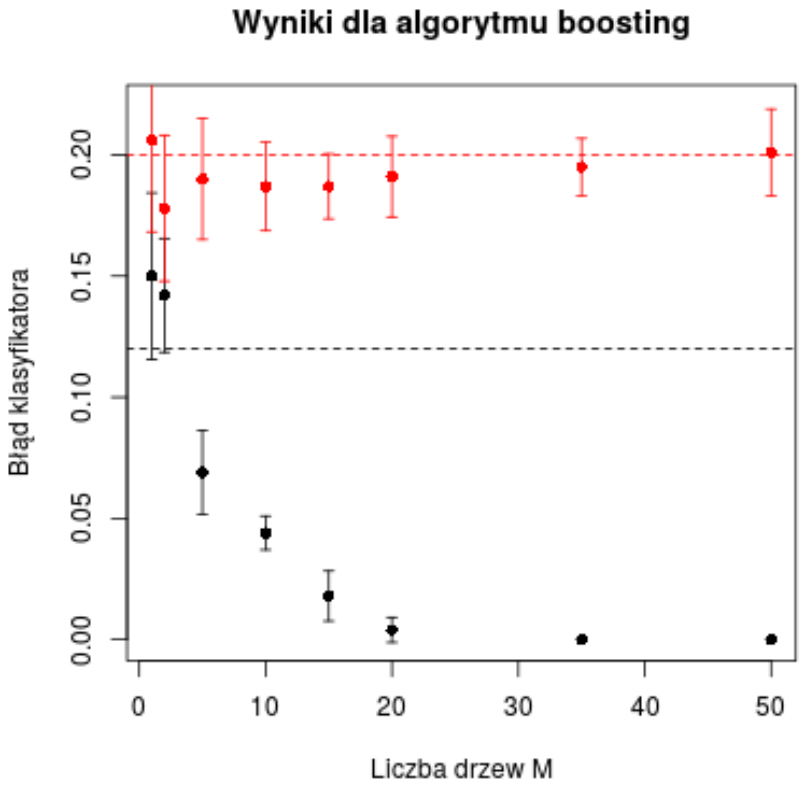


Przykład


Wyniki dla algorytmu boosting
Błąd klasyfikatora
0.00
0.05
0.10
0.15
0.20
0
10
20
30
40
50
Liczba drzew M

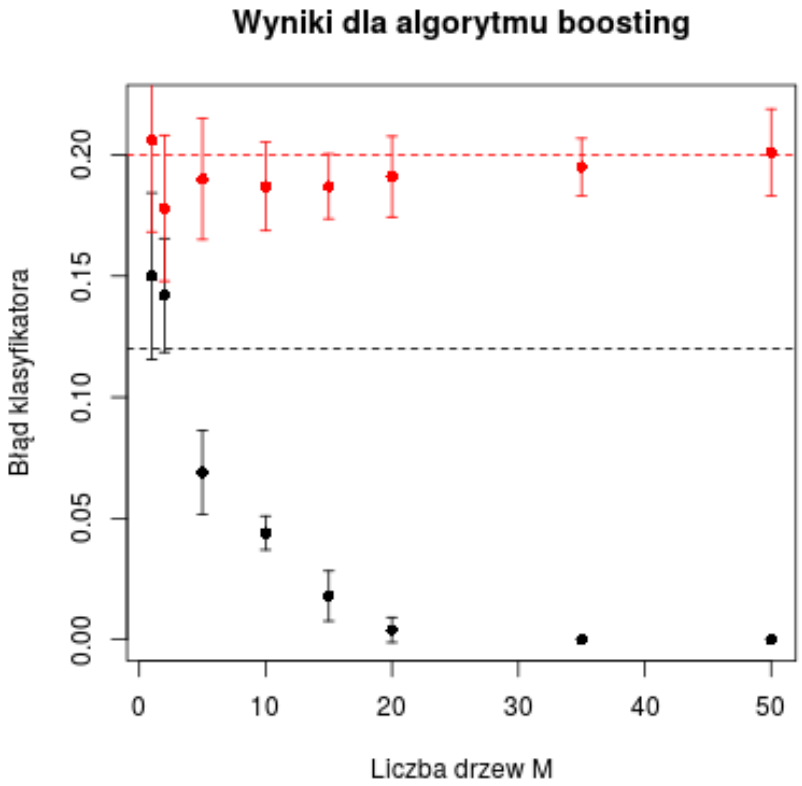


Uruchamiamy algorytm boosting dla naszego przykładu:



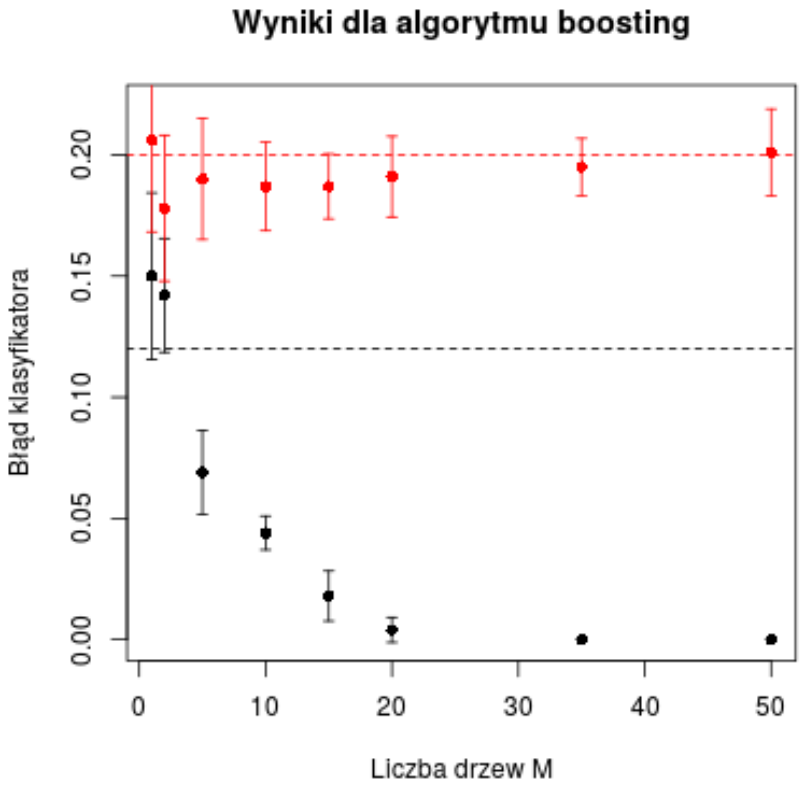
Uruchamiamy algorytm boosting dla naszego przykładu:

- wraz ze wzrostem liczby drzew błąd klasyfikatora spada (czarne - PU, czerwone - PT),



Uruchamiamy algorytm boosting dla naszego przykładu:

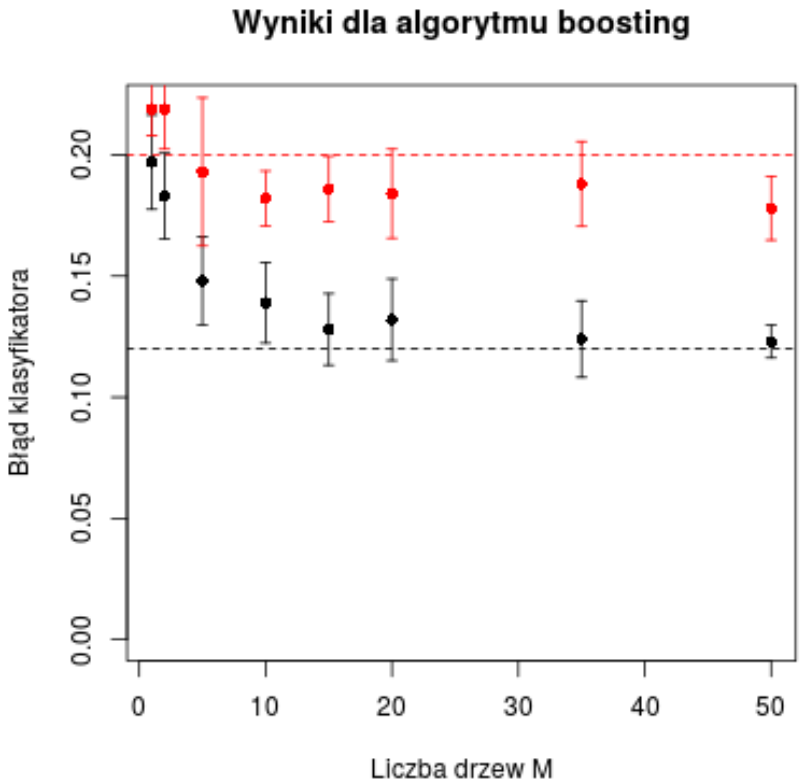
- wraz ze wzrostem liczby drzew błąd klasyfikatora spada (czarne - PU, czerwone - PT),
- ujawnia się podstawowe cecha boostingu: znaczące obniżenie dla błędu treningowego,

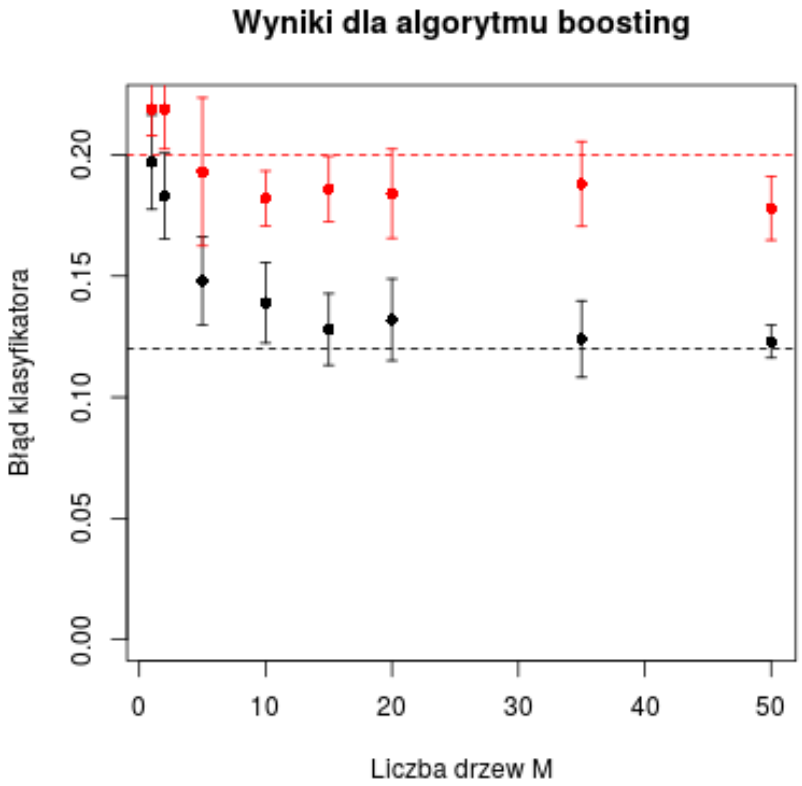


Uruchamiamy algorytm boosting dla naszego przykładu:

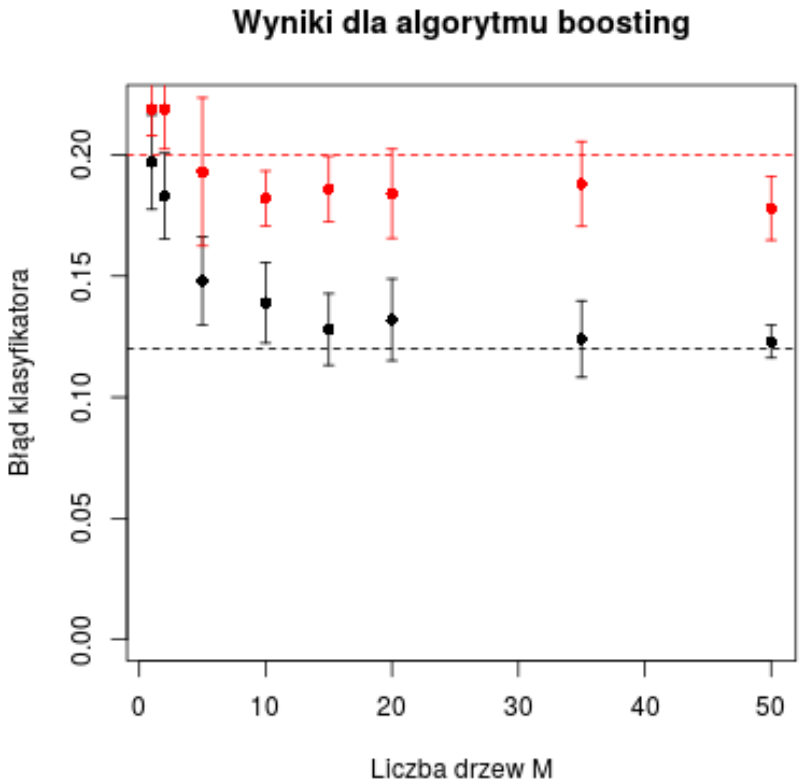
- wraz ze wzrostem liczby drzew błąd klasyfikatora spada (czarne - PU, czerwone - PT),
- ujawnia się podstawowe cecha boostingu: znaczące obniżenie dla błędu treningowego,
- dla PT wartość błędu jest podobna jak dla optymalnego drzewa.

Przykład


Wyniki dla algorytmu boosting
Błąd klasyfikatora
0.00
0.05
0.10
0.15
0.20
0
10
20
30
40
50
Liczba drzew M

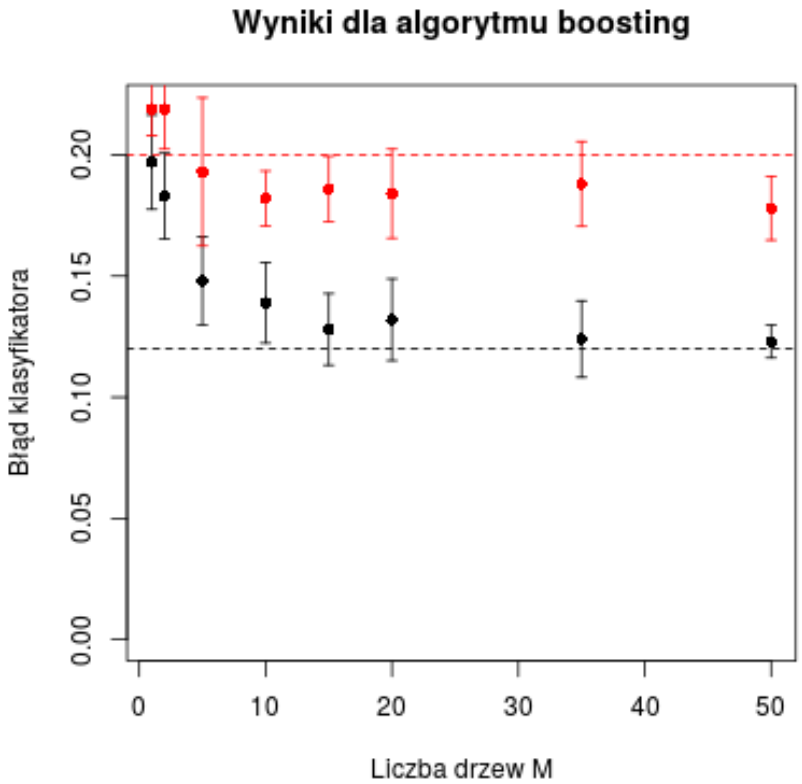


Uruchamiamy algorytm boosting dla naszego przykładu, ale tym razem podstawowym klasyfikatorem jest drzewa składające się jedynie z korzenia i dwóch liści:



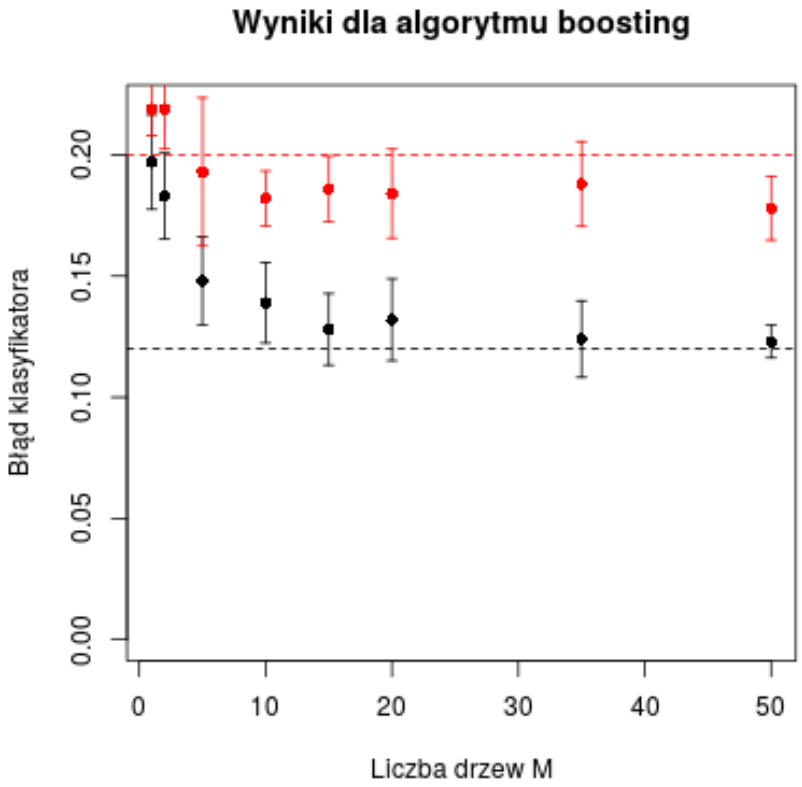
Uruchamiamy algorytm boosting dla naszego przykładu, ale tym razem podstawowym klasyfikatorem jest drzewa składające się jedynie z korzenia i dwóch liści:

- jakość dla PU spada,



Uruchamiamy algorytm boosting dla naszego przykładu, ale tym razem podstawowym klasyfikatorem jest drzewa składające się jedynie z korzenia i dwóch liści:

- jakość dla PU spada,
- dla PT otrzymaliśmy poprawę,



Uruchamiamy algorytm boosting dla naszego przykładu, ale tym razem podstawowym klasyfikatorem jest drzewa składające się jedynie z korzenia i dwóch liści:

- jakość dla PU spada,
- dla PT otrzymaliśmy poprawę,
- dalej problemem jest prawdopodobnie zbyt wysoka skuteczność podstawowego klasyfikatora

Rzeczywisty **AdaBoost**

1. Przyjmij wagi $w_i = \frac{1}{n}$, $i = 1, ..., N$.
2. Dla $m = 1, ..., M$:
   - stosując do danych uczących wagi w_i, wytrenuj klasyfikator, dający w wyniku estymator

Rzeczywisty **AdaBoost**

1. Przyjmij wagi $w_i = \frac{1}{n}$, $i = 1, ..., N$.
2. Dla $m = 1, ..., M$:
   - stosując do danych uczących wagi w_i, wytrenuj klasyfikator, dający w wyniku estymator

     $$p_m(\mathbf{x}) = p_w(y = 1|\mathbf{x})$$

     prawdopodobieństwa a posteriori przynależności do klasy $y = 1$, obliczany względem rozkładu wag w_i

Rzeczywisty **AdaBoost**

1. Przyjmij wagi $w_i = \frac{1}{n}$, $i = 1, ..., N$.
2. Dla $m = 1, ..., M$:
   - stosując do danych uczących wagi w_i, wytrenuj klasyfikator, dający w wyniku estymator

     $$p_m(\mathbf{x}) = p_w(y = 1|\mathbf{x})$$

     prawdopodobieństwa a posteriori przynależności do klasy $y = 1$, obliczany względem rozkładu wag w_i
   - podstaw,

Rzeczywisty **AdaBoost**

1. Przyjmij wagi $w_i = \frac{1}{n}$, $i = 1, ..., N$.
2. Dla $m = 1, ..., M$:
   - stosując do danych uczących wagi w_i, wytrenuj klasyfikator, dający w wyniku estymator

     $$p_m(\mathbf{x}) = p_w(y = 1|\mathbf{x})$$

     prawdopodobieństwa a posteriori przynależności do klasy $y = 1$, obliczany względem rozkładu wag w_i
   - podstaw,

     $$f_m(\mathbf{x}) = \frac{1}{2} \ln \frac{p_m(\mathbf{x})}{1 - p_m(\mathbf{x})}$$

   - podstaw,

Rzeczywisty **AdaBoost**

1. Przyjmij wagi $w_i = \frac{1}{n}$, $i = 1, ..., N$.
2. Dla $m = 1, ..., M$:
   - stosując do danych uczących wagi w_i, wytrenuj klasyfikator, dający w wyniku estymator

     $$p_m(\mathbf{x}) = p_w(y = 1|\mathbf{x})$$

     prawdopodobieństwa a posteriori przynależności do klasy $y = 1$, obliczany względem rozkładu wag w_i
   - podstaw,

     $$f_m(\mathbf{x}) = \frac{1}{2} \ln \frac{p_m(\mathbf{x})}{1 - p_m(\mathbf{x})}$$

   - podstaw,

     $$w_i = w_i \exp(y_i f_m(\mathbf{x}_i)) \qquad i = 1, ..., N$$

     i dokonaj renormalizacji tak, aby $\sum_i w_i = 1$.

Rzeczywisty **AdaBoost**

1. Przyjmij wagi $w_i = \frac{1}{n}$, $i = 1, ..., N$.
2. Dla $m = 1, ..., M$:
   - stosując do danych uczących wagi w_i, wytrenuj klasyfikator, dający w wyniku estymator

     $$p_m(\mathbf{x}) = p_w(y = 1|\mathbf{x})$$

     prawdopodobieństwa a posteriori przynależności do klasy $y = 1$, obliczany względem rozkładu wag w_i
   - podstaw,

     $$f_m(\mathbf{x}) = \frac{1}{2} \ln \frac{p_m(\mathbf{x})}{1 - p_m(\mathbf{x})}$$

   - podstaw,

     $$w_i = w_i \exp(y_i f_m(\mathbf{x}_i)) \qquad i = 1, ..., N$$

     i dokonaj renormalizacji tak, aby $\sum_i w_i = 1$.
3. Podaj

   $$\operatorname{sgn}\left[\sum_{m=1}^{M} f_m(\mathbf{x})\right]$$

Rozszerzenie rzeczywistego AdaBoost na $g > 2$

1. Utwórz z N-elementowej PU nastepującą PU o Ng elementach:

$$((\mathbf{x}_i, 1), y_{i1}), ((\mathbf{x}_i, 2), y_{i2}), ..., ((\mathbf{x}_i, g), y_{ig})$$

$i = 1, ..., N$, gdzie y_{ik} odgrywa rolę etykiety obserwacji i równa się 1, jeżeli obserwacja $(\mathbf{x}_i, k)$ należy do klasy k oraz $y_{ik} = -1$ w przeciwnym przypadku.

Rozszerzenie rzeczywistego AdaBoost na $g > 2$

1. Utwórz z N-elementowej PU nastepującą PU o Ng elementach:

$$((\mathbf{x}_i, 1), y_{i1}), ((\mathbf{x}_i, 2), y_{i2}), ..., ((\mathbf{x}_i, g), y_{ig})$$

   $i = 1, ..., N$, gdzie y_{ik} odgrywa rolę etykiety obserwacji i równa się 1, jeżeli obserwacja $(\mathbf{x}_i, k)$ należy do klasy k oraz $y_{ik} = -1$ w przeciwnym przypadku.

2. Zastosuj algorytm rzeczywisty AdaBoost do próby Ng-elementowej, by otrzymać funkcję

$$F((x), k) = \sum_{m=1}^{M} f_c(\mathbf{x}, k))$$

3. Podaj

$$\arg\max_k F(\mathbf{x}, k)$$

Opisane rodziny klasyfikatorów mogą korzystać z różnych typów cegiełek, tj. elementarnych klasyfikatorów. Opieramy się zwykle na drzewach, gdyż są one dość uniwersalne, jednocześnie wymagając najczęściej dozy stabilizacji.

Opisane rodziny klasyfikatorów mogą korzystać z różnych typów cegiełek, tj. elementarnych klasyfikatorów. Opieramy się zwykle na drzewach, gdyż są one dość uniwersalne, jednocześnie wymagając najczęściej dozy stabilizacji.

Okazuje się jednak, że algorytmy związane z boostingiem nie są jedynymi najlepszymi klasyfikatorami. Można je ustawić w tym samym rzędzie z odkryciem Leo Breimana z 2001 r. — **lasami losowymi** (ang. *random forests*).

Opisane rodziny klasyfikatorów mogą korzystać z różnych typów cegiełek, tj. elementarnych klasyfikatorów. Opieramy się zwykle na drzewach, gdyż są one dość uniwersalne, jednocześnie wymagając najczęściej dozy stabilizacji.

Okazuje się jednak, że algorytmy związane z boostingiem nie są jedynymi najlepszymi klasyfikatorami. Można je ustawić w tym samym rzędzie z odkryciem Leo Breimana z 2001 r. — **lasami losowymi** (ang. *random forests*).

Lasy losowe (w odróżnieniu od boostingu) **muszą** używać drzew decyzyjnych jako pojedynczych klasyfikatorów i wyróżniają je następujące cechy:

- prawdopodobieństwo popełnienia błędu rośnie wraz ze stopniem korelacji pomiędzy poszczególnymi drzewami,
- prawdopodobieństwo popełnienia błędu maleje wraz ze wzrostem siły pojedynczych drzew,

Lasy losowe - algorytm

Algorytm jest bardzo prosty:

1. Wylosuj ze zwracaniem z oryginalnej N-elementowej próby uczącej N wektorów obserwacji do pseudopróby, na której zostanie zbudowane drzewo.

Lasy losowe - algorytm

Algorytm jest bardzo prosty:

1. Wylosuj ze zwracaniem z oryginalnej N-elementowej próby uczącej N wektorów obserwacji do pseudopróby, na której zostanie zbudowane drzewo.
2. W każdym węźle drzewa podział podpróby odbywa się następująco: niezależnie od innych losowań wylosuj r spośród p atrybutów wektora obserwacji (bez zwracania), a następnie zastosuj przyjętą regułę podziału do wylosowanych r atrybutów ($r \ll p$).
3. Drzewo jest budowane bez przycinania, jeśli to możliwe aż do otrzymania liści zawierających elementy tylko z jednej klasy.

Klasyfikacja przez las losowy odbywa się tak jak w baggingu: dany wektor obserwacji zostaje poddany klasyfikacji przez wszystkie drzewa i zaliczony do klasy, która uzyskała większość głosów.

Uwagi

- Jedynie parametr r wymaga ustalenia.

Uwagi

- Jedynie parametr r wymaga ustalenia.
- Przyjmuje się, że wartością dającą dobre wyniki jest ustalenie $r = \sqrt{p}$ – można go też dobrać adaptacyjnie,na podstawie szacowania błędu lasu.
- Z uwagi na prostotę procedury, algorytm często jest używany do dużych zbiorów (np. p rzędu **tysięcy**).

Lasy losowe - przykład