Zespoły modeli predykcyjnych

Systemy uczące się wykład 8

Jerzy Stefanowski

Instytut Informatyki PP

2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa
POPC.03.02.00-00-0001/20

Fundusze Europejskie Polska Cyfrowa

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

Plan wykładów 8 i 9

- Motywacje do tworzenia zespołów modeli
- Kiedy łączenie klasyfikatorów jest skuteczne
- Różne podejścia do tworzenia zróżnicowanych klasyfikatorów
- Zasady agregacji odpowiedzi klasyfikatorów składowych
- Metoda bagging
- Feature esmebles i Random forest

Plan wykładów 8 i 9

- Metody Boosting
  - AdaBoost
- Porównania Bagging vs. Boosting
- Zróżnicowanie klasyfikatorów składowych
- Generalizacja stosowa (stacking) i tzw. mixture of experts
- Podejścia zespołowe do danych silnie wieloklasowych
- Boosted trees ensembles
- Podsumowanie

Ogólne założenia

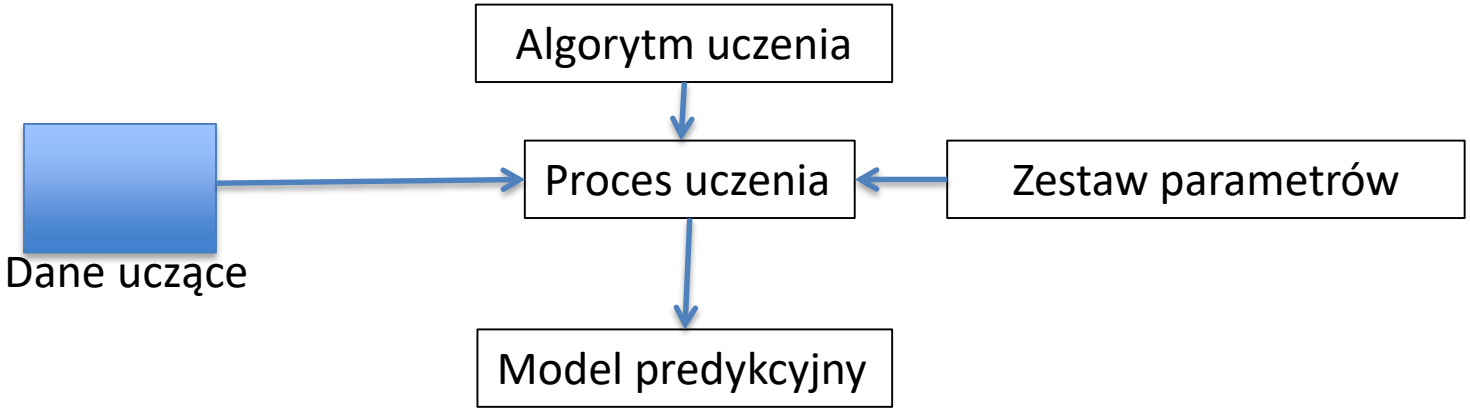
- Zespoły modeli (szerszy termin) obejmują integrację wielu składowych modeli nauczonych dla
  - Klasyfikatorów
  - Modeli regresji
  - Algorytmów tworzenia skupień (nie zajmujemy się w tym wykładzie)

Typowe podejście do uczenia nadzorowanego

Dla danego problemu – danych (przykładów uczących)

Poszukuje się jednego najlepszego modelu (klasyfikatora)

Wybór algorytmu uczenia klasyfikatora, dobór parametrów, porównanie z innych możliwymi klasyfikatorami (algorytmami), intensywne oceny eksperymentalne (np.z wykorzystaniem k-fold cross validation)



Czy zawsze szukać jednego modelu?

Lekcje z doświadczeń

- Nie ma jednego algorytmu najlepszego dla wielu możliwych problemów!
- Można skonstruować model wystarczająco skuteczny dla wybranego zestawu problemów.
- Skomplikowane, złożone problemy często mogą być zdekomponowane / rozłożone na prostsze pod-problemy (rozwiązane niezależnie i zintegrowane)

Motywacja – integracja wielu niezależnych modeli!

W stronę rozwiązań zespołowych

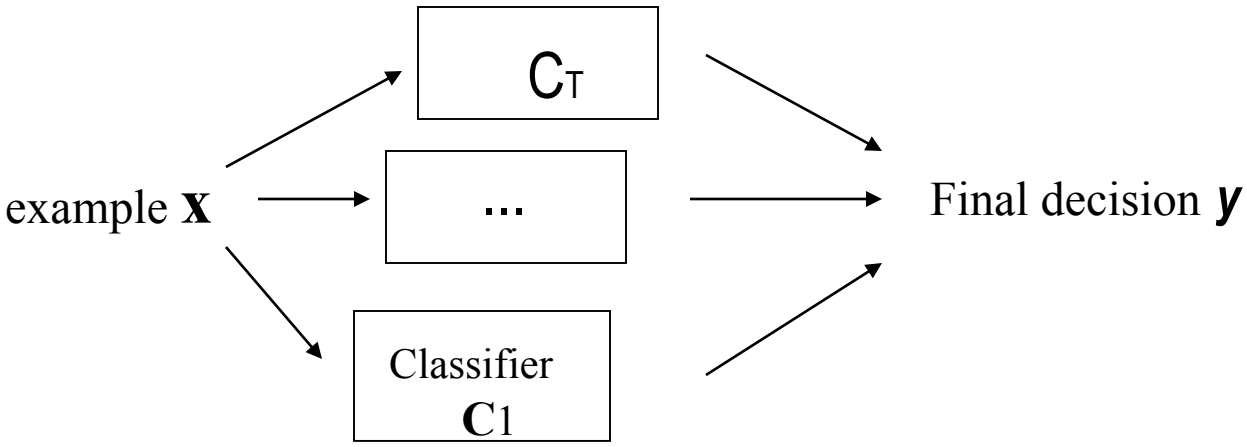
- Integracja wielu modeli w jeden system predykcyjnych może polepszyć trafność predykcji oraz pozwolić rozwiązać bardziej złożone problemy
  - Cytat:

    „Multiple learning systems try to exploit the local different behavior of the base learners to enhance the accuracy of the overall learning system”

    - G. Valentini, F. Masulli
- Terminy angielskie - ensembles lub multiple learning classifiers

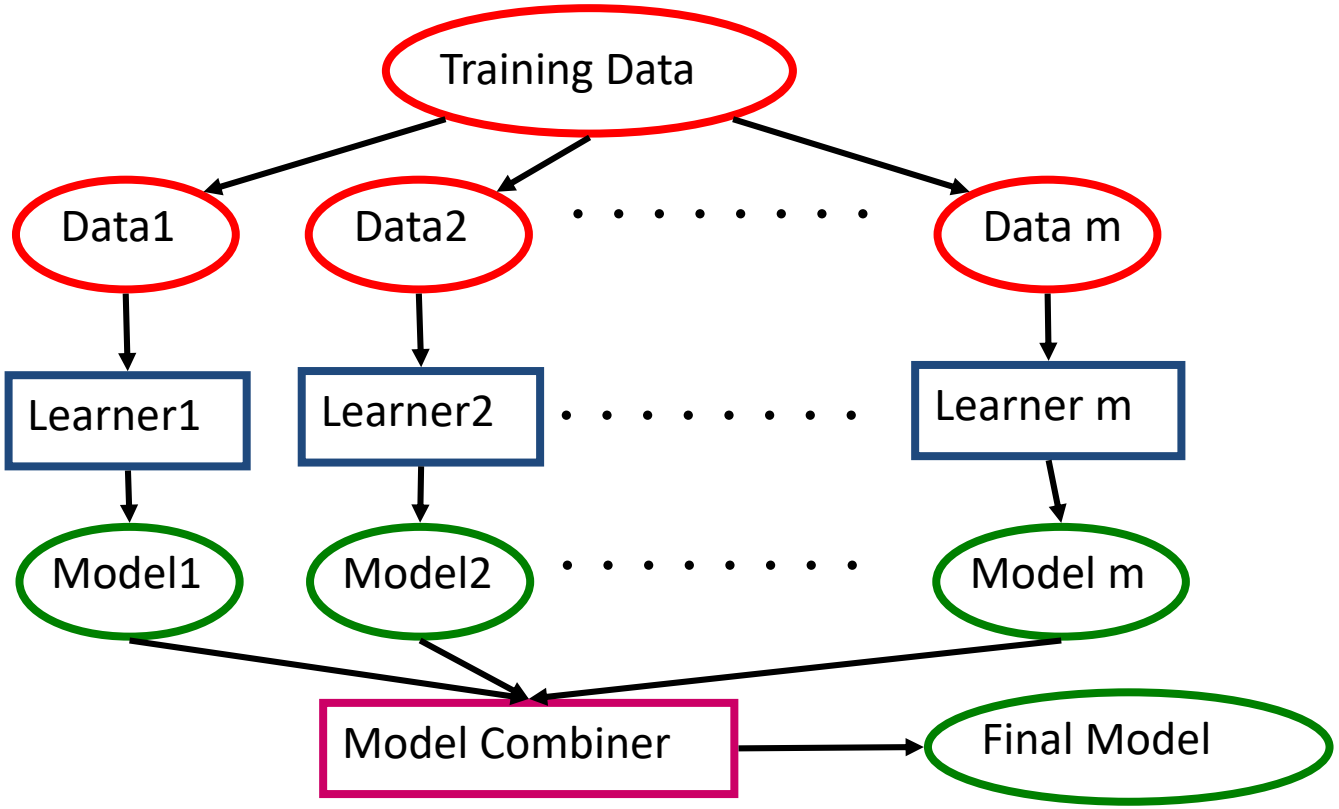
Definicja

- Zbiór wielu modeli (klasyfikatorów, predyktorów regresji) których decyzje są integrowane tak aby klasyfikować nowe przykłady
- Różne nazwy po angielsku: **ensemble** methods, multiple classifiers, committee, classifier fusion, combination,…
- Przetarg pomiędzy złożonością systemu (ang. complexity) a zdolnością poprawy predykcji



Uczenie zespołów

- Naucz się alternatywnych definicji problemów – modeli poprzez ich zróżnicowanie (np. wiele różnych wersji danych uczących albo różnych algorytmów)
- Integracja predykcji – np. poprzez różne formy głosowania



Kiedy integracja jest skuteczna?

- Kiedy zespół może być skuteczniejszy niż pojedynczy model?
- Łączenie tak samo działających modeli jest nieskuteczne!

- Niezbędny jest pewien poziom niezgodności klasyfikatorów składowych, rozumiany w ten sposób, że jeśli popełniają błędne decyzje, to są one niezależne pomiędzy nimi (czyli nie popełniają równocześnie takich samych błędów)

- Pierwsze systematyczne prace (np. Hansen&Salamon90, Ali&Pazzani96), cyt:
  Member classifiers should **make uncorrelated errors** with respect to one another; each classifier should perform better than a random guess.

Intuicja nieskorelowania błędnych predykcji

Rozważmy 3 klasyfikatory i głosowanie większościowe

Poprawna klasa	Model 1	Model 2	Model 3	Zespół
A	A	B	A	A
A	A	A	B	A
B	A	B	B	B
A	B	B	B	B
B	B	B	A	B
B	A	B	B	B
B	B	A	B	B
A	B	A	A	A
B	B	B	A	B
A	A	A	A	A
Accuracy	60%	70%	60%	90%

Oczekiwana trafność predykcji zespołu

Niech: *L* – nieparzysta liczba składowych klasyfikatorów; *p* – prawdopodobieństwo poprawnej klasyfikacji składnika; wszystkie predykcje składników są prob. niezależne

Oczekiwana trafność predykcji zespołu L klasyfikatorów w głosowaniu większościowym:

$$p_{maj} = \sum_{m=\lfloor L/2 \rfloor + 1}^{L} \binom{L}{m} p^m (1-p)^{L-m}$$

P> 0.5 to p_{maj} -> 1 wraz ze wzrostem L

P<0.5 to p_{maj} -> 0 wraz ze wzrostem L

P=0.5 to p_{maj}=0.5

Dokładniejsza analiza w książce L.Kuncheva

Głosowanie większościowe p_{maj}

L niezależnych klasyfikatorów, każdy z oczekiwana trafnością *p*

	L=5	L=7	L=9
p=0.6	0.6827	0.7102	0.7334
p=0.7	0.8369	0.8740	0.9012
p=0.8	0.9421	0.9667	0.9804
p=0.9	0.9914	0.9973	0.9991

Poprawa predykcji wobec pojedynczego klasyfikatora

- Binarny zbalansowany problem; składowe klasyfikatory (o tym samym spodziewanym błędzie) I braku korelacji predykcji; głosowanie większościowe w zespole klasyfikatorów
- Oczekiwany błąd predykcji zespołu maleje wraz ze liczbą klasyfikatorów

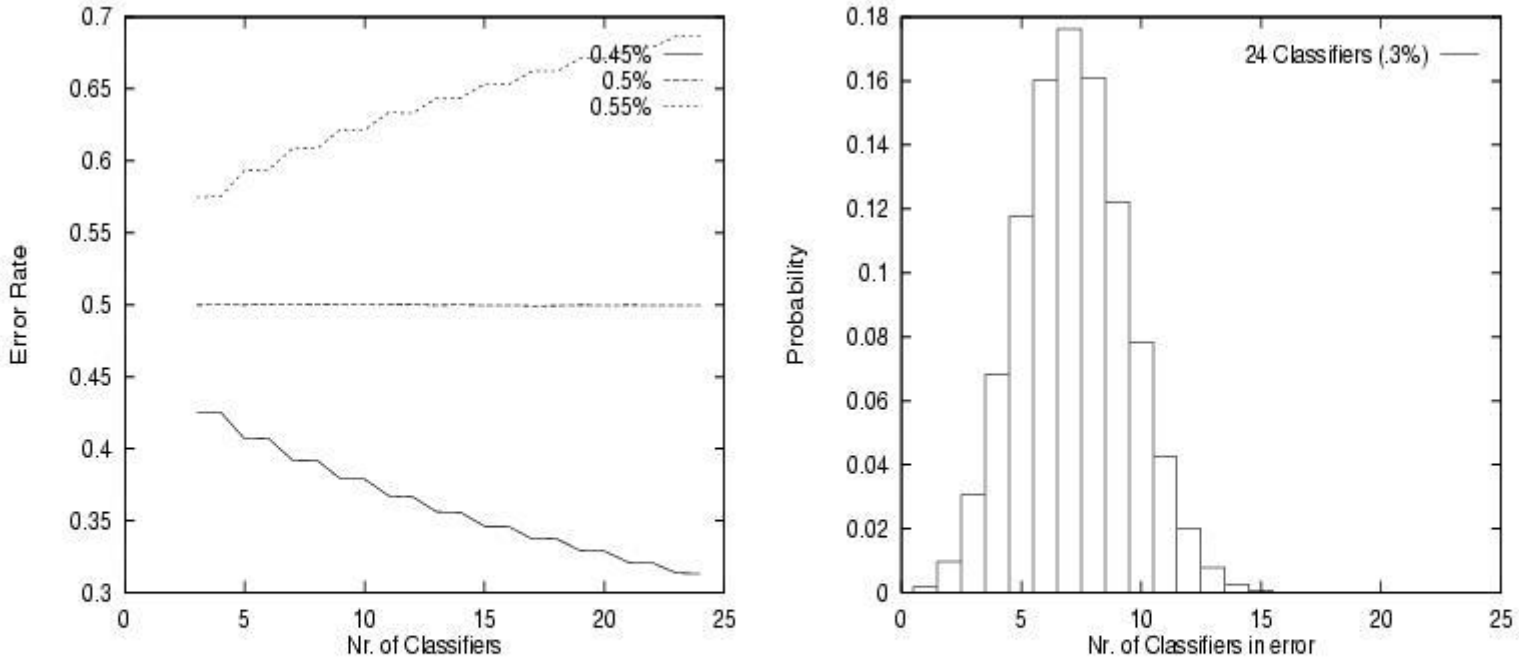


Figure 5.1: (a) Error rate versus nr. of classifiers in an ensemble. (b) Probability that exactly n of 24 classifiers will make an error.

Za doktorat J.Gama

Inna interpretacja poprawy zespół vs. pojedynczy model

T.Dietterich: statystyczna (dobór próby uczącej); wybór języka reprezentacji (rodzaju klasyfikatora); perturbacja parametrami uczenia.

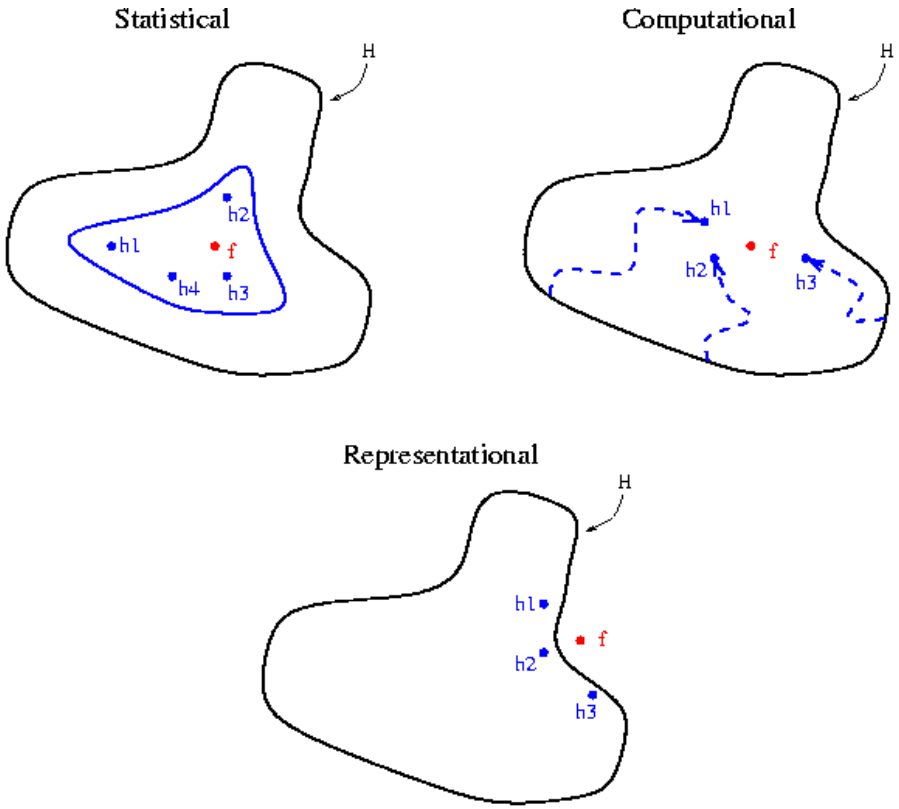
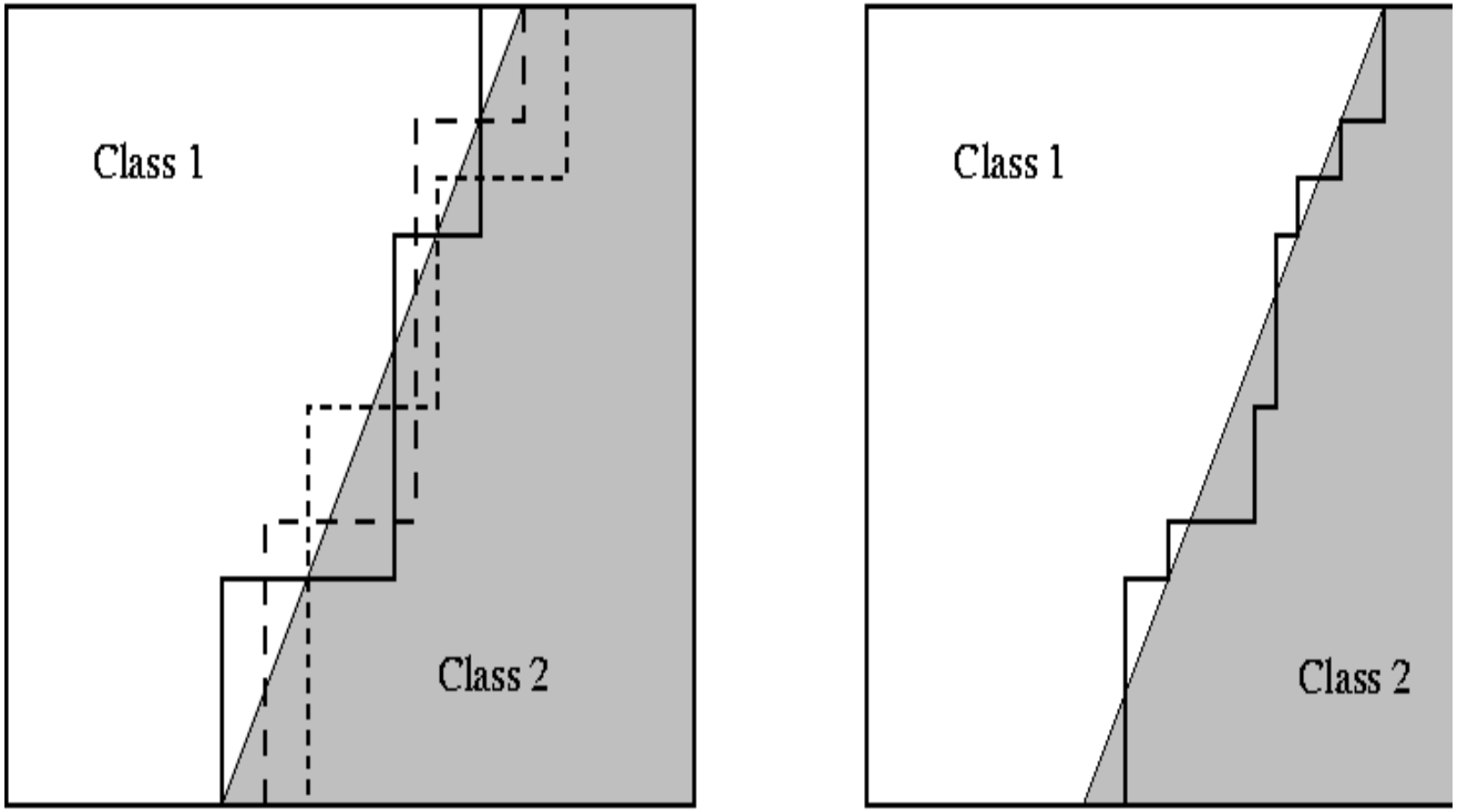


Fig. 2. Three fundamental reasons why an ensemble may work better than a single classifier

Zespół drzew klasyfikacyjnych vs. pojedyncze drzewo



Tworzenie zróżnicowanych klasyfikatorów składowych w zespołach

- Różne zbiory uczące (losowanie, wagowanie przykładów, inne podziały oryginalnego zbioru uczącego)
- Różne algorytmy uczące (zastosowane do tego samego zbioru uczącego)
- Wybór różnych podzbiorów atrybutów

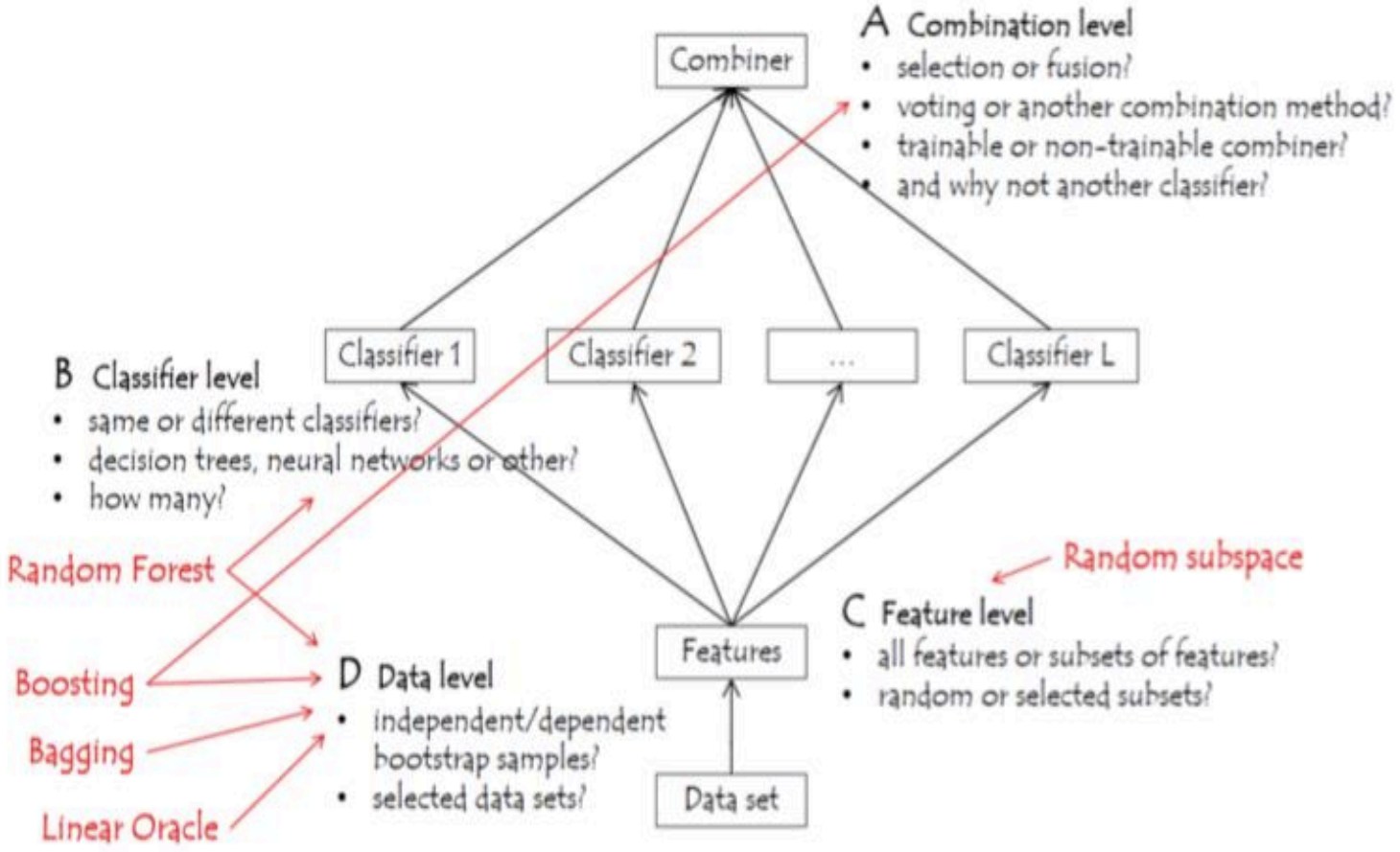
  (częste dla tekstów, mowy, obrazów)
- Zmiany parametrów algorytmu uczącego

  (np. w ANN, stopień uproszczenia drzewa)
- Inne inicjalizacje algorytmów (stochastycznych)

Typowa kategoryzacja algorytmów

- **Jednorodne modele** (Homogeneous classifiers) – użycie tego samego algorytmu na wielu zróżnicowanych zbiorach danych
  - Bagging (Breiman)
  - Boosting (Freund, Schapire)
  - Random Forest (Breiman)
  - Inne podzialy zbioru (np. Ho, Lattine)
  - Specjalizowane dla wieloklasowości, (np.. ECOC pairwise classification)
- **Niejednorodne** (Heterogeneous classifiers) – jawnie inne algorytmy zastosowane do tego samego zbioru danych
  - Stacked generalization lub meta-learning
  - Rozwiązania dla innych złożonych danych

Ogóle spojrzenie na tworzenie zespołów



Jak agregować predykcje?

- Klasyfikacja
  - Głos (zerojedynkowy) lub tzw. współczynnik score / prawdopodobieństwa)
- Regresja
  - Uśrednianie predykcji liczbowych $y_{maj}^{-}=\sum_{i=1}^{L} y_i$
  - Inne formy
- Czy wszystkie modele składowe biorą udział w wypracowaniu decyzji zespołu?

Agregacje predykcji klasyfikatorów

Głosowanie vs. inne podejścia

- Odmiany głosowania
  - Każdy model ma taki głos o tej samej wadze – decyzja końcowa największa liczba głosów na klasę
  - Głos każdego klasyfikatora ma wagę (suma wag albo intepretacja klasyfikacji Bayesowskiej)
- Non-voting - najczęściej dotyczy agregacji wskazań liczbowych (współczynniki score, prawdopodobieństwa klas)
  - Specjalne formuły (product, sum, min, max, median,…)
- Uczenie się b. złożonych agregacji – tzw. meta uczenie (extra meta-learner / combiner)

Przykłady formuł dla wskazań liczbowych

Rule	Fusion function $f(\cdot)$
Sum	$y_i = \frac{1}{L}\sum_{j=1}^{L} d_{ji}$
Weighted sum	$y_i = \sum_j w_j d_{ji}, w_j \geq 0, \sum_j w_j = 1$
Median	$y_i = \text{median}_j\, d_{ji}$
Minimum	$y_i = \min_j d_{ji}$
Maximum	$y_i = \max_j d_{ji}$
Product	$y_i = \prod_j d_{ji}$

	C_1	C_2	C_3
d_1	0.2	0.5	0.3
d_2	0.0	0.6	0.4
d_3	0.4	0.4	0.2
Sum	0.2	**0.5**	0.3
Median	0.2	**0.5**	0.4
Minimum	0.0	**0.4**	0.2
Maximum	0.4	**0.6**	0.4
Product	0.0	**0.12**	0.032

Grupowe lub selektywne podejmowanie decyzji

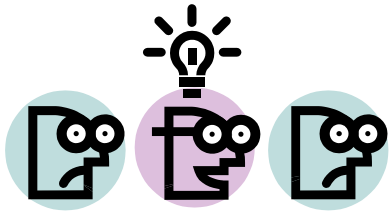
- Grupowe (statyczne) – wszystkie klasyfikatory składowe biorą udział w wypracowaniu decyzji końcowej zespołu.
- Specjalizowany wybór / dynamiczna integracja – wybór podzbioru “kompetentnych” klasyfikatorów
  - Poszukiwanie tych klasyfikatorów, które są ekspertami dla opisu klasyfikowanego przykładu

Dynamiczne składanie głosów

Zamiast wyboru najbardziej kompetentnych klasyfikatorów

Dynamic voting: [propozycja A.Tsymbal]

- For każdy nowy obiekt do klasyfikacji:
  - Znajdź jego *h*-nearest neighbors w oryginalnym zbiorze uczącym
  - Reklasyfikuj je przez klasyfikatory składowe
  - Użyj informacji o skuteczności reklasyfikacji do oszacowania wag klasyfikatorów.



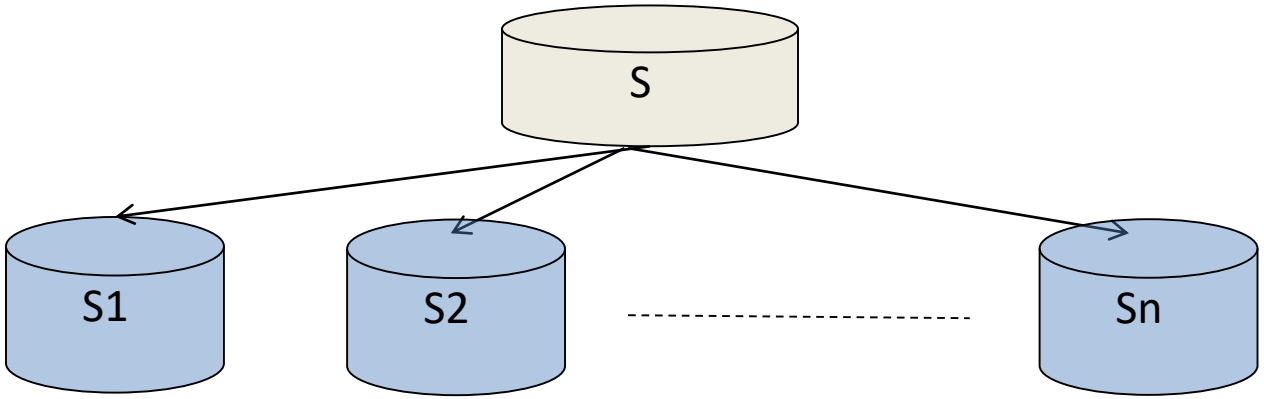
Jednorodne klasyfikatory

- Zróżnicowanie poprzez modyfikacje zbiorów uczących
  - **Data1 ≠ Data2 ≠ … ≠ DataT**
- Użycie tego samego algorytmu uczącego
  - Różne dane -> różne klasyfikatory
- Najpopularniejsze propozycje
  - Bagging: losowanie bootstratpowe
  - Boosting: modyfikacja wag przykładów
  - Random Subspace (losowanie cech)
  - Random Forest

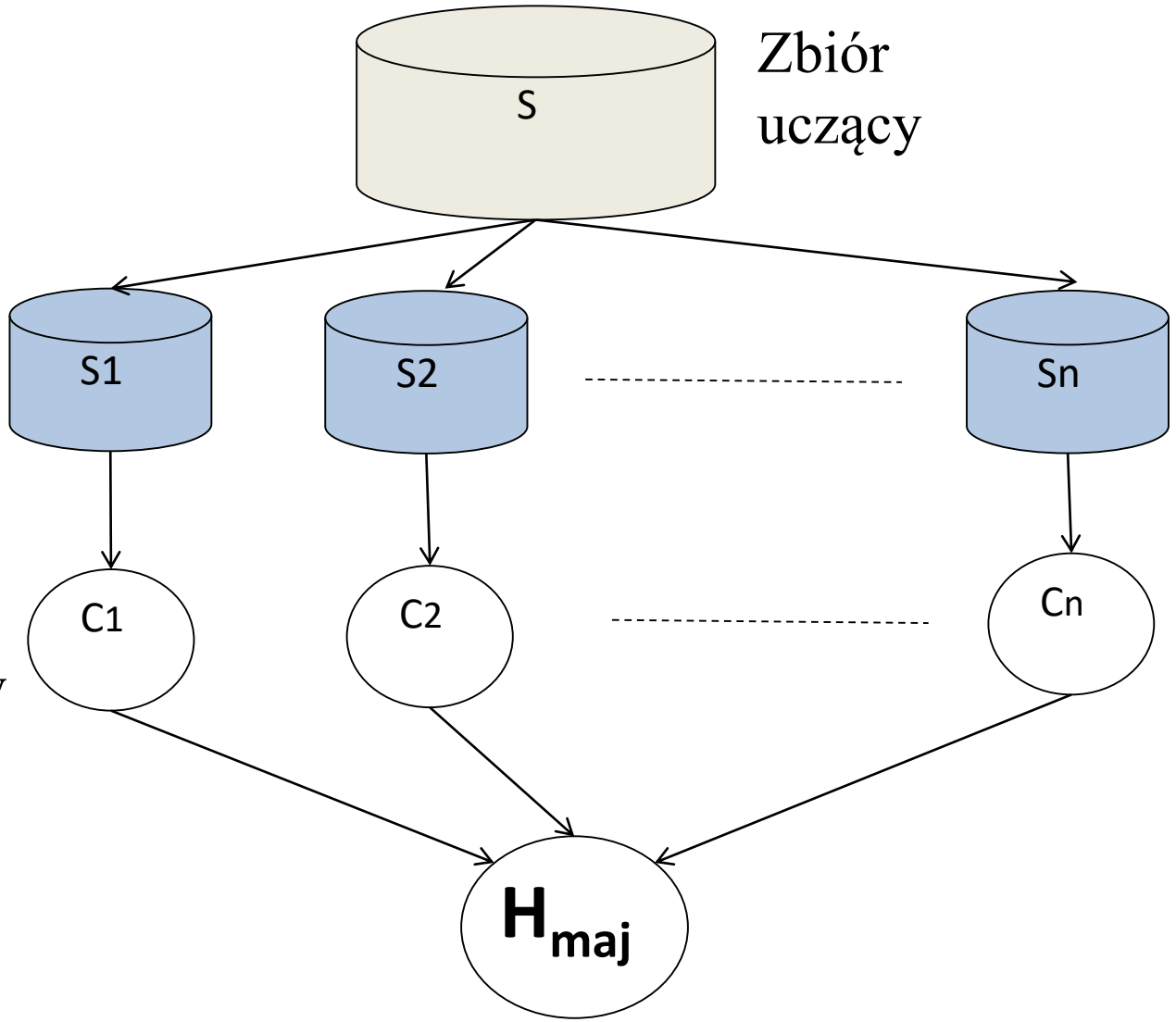
Bagging [L.Breiman, 1996]

- Bagging = **B**ootstrap **agg**regation
- Wielokrotne losowanie różnych podzbiorów przykładów z początkowego zbioru
- Zastosowanie tego samego algorytmu uczącego
- Agregacja predykcji klasyfikatorów składowych
  - Klasyfikacja – różne formy głosowania
  - Regresja – uśrednianie odpowiedzi

Próby bootstrap-owe



Schemat zespołu klasyfikatorów bagging



Bootstrap aggregation

- Losowanie bootstrapowe – losowanie ze zwracaniem
  - Do danej próbki niektóre przykłady zostaną wylosowane kilka razy, a niektóre nie zostaną wylosowane
  - Przy wielkości próbki zbliżone do wielkości oryginalnego zbioru danych, średnio do próbki trafia 63.2% przykładów z tego zbioru
- Oszacowanie:
  - Dla danych o wielkości N, każdy przykład ma prawdopodobieństwo wylosowania przynajmniej raz równe $1-(1-1/N)^N$
  - Dla wysokich N – dąży to do (1-1/e) or 0.632 [Bauer and Kohavi, 1999]
- Próbka bootstrapowa – wielkość zbliżona do oryginalnego zbioru albo może być mniejsza (np. pasting small votes lub inne uogólnienia bagging)

Losowanie bootstrapowe

Początkowy zbiór przykładów

S ={x1,x2,x3,x4}

Wylosowane próbki do bagging

S1={x1,x2,x3,x3}

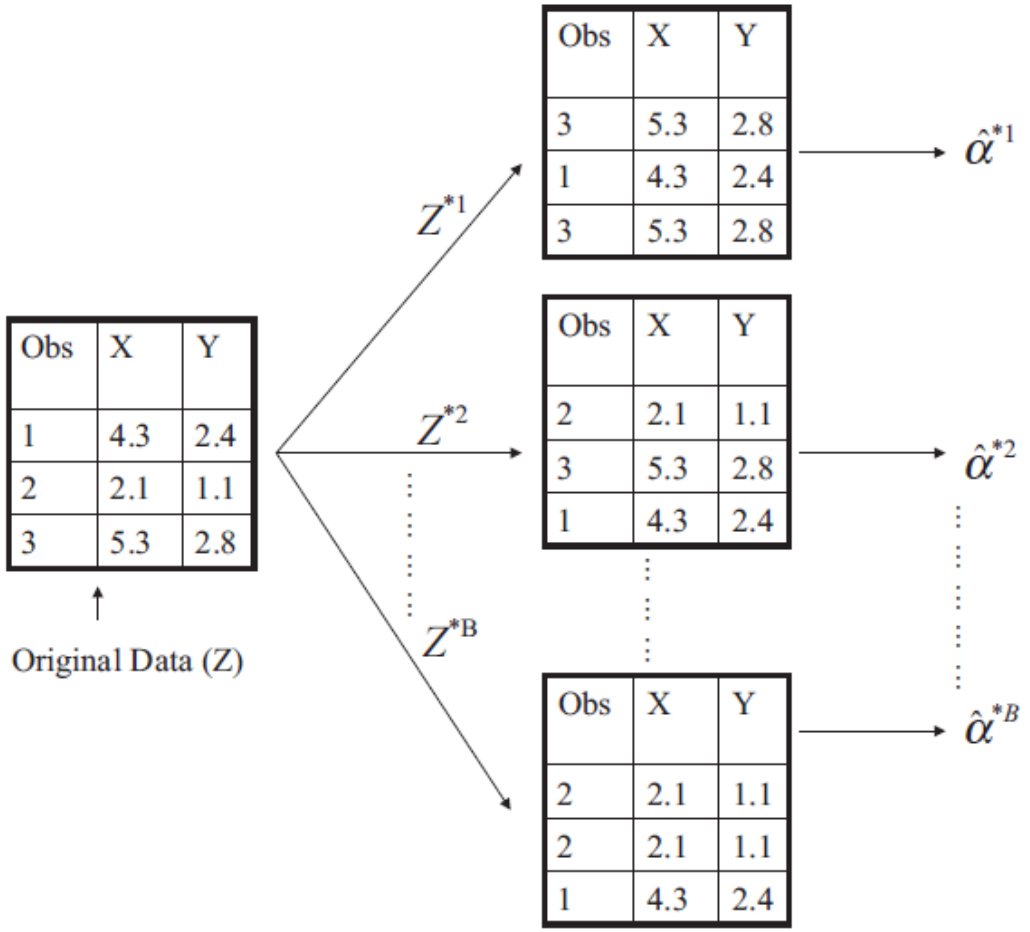
S2={x1,x4,x4,x4}

S3={x1,x1,x2,x2}

S4=…

Losowanie boostrap-owe

Ilustracja



Ogólny zapis metody bagging

Uczenie

input S – zbiór uczący, T – # składowych model, LA – algorytm uczący
output E- zespoł złożony z H_i składników
for i=1 **to** T **do**
begin
S_i:=bootstrap sample from S;
H_i:=$LA(S_i)$;
add H_i to ensemble E
end;

Predykcja – przykład **x**

Klasyfikuj **x** przez każdy klasyfikator H_i – wskaznie etykiety klasy

Agreguj do Dj wskazania (oryginalnie sumuj głosy za każdą z klas)

Wybierz klase maksymalizującą Dj

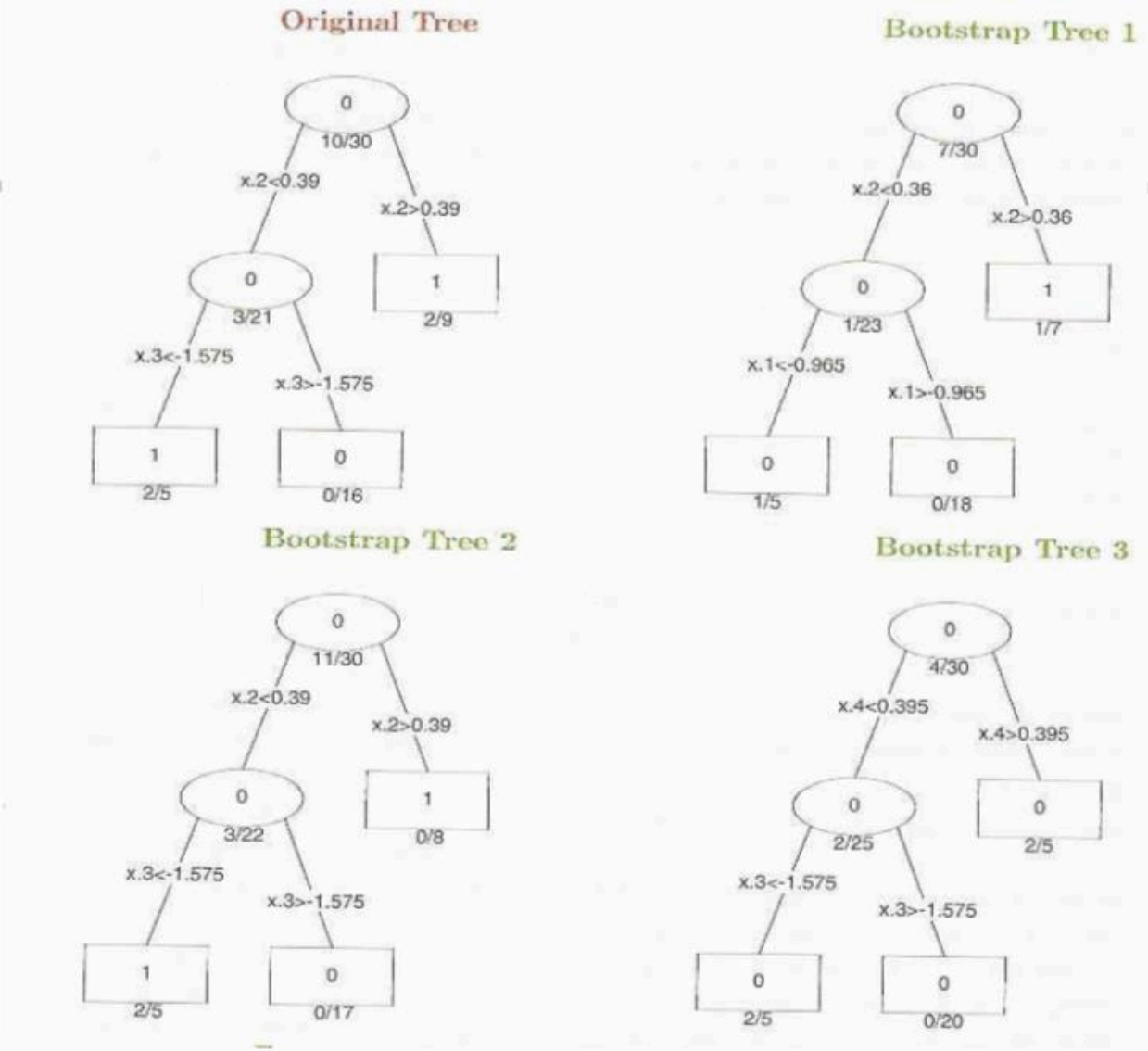
Przykład oceny eksperymentalnej

Misclassification error rates [Percent]

Data	Single	Bagging	Decrease
waveform	29.0	19.4	33%
heart	10.0	5.3	47%
breast cancer	6.0	4.2	30%
ionosphere	11.2	8.6	23%
diabetes	23.4	18.8	20%
glass	32.0	24.9	22%
soybean	14.5	10.6	27%

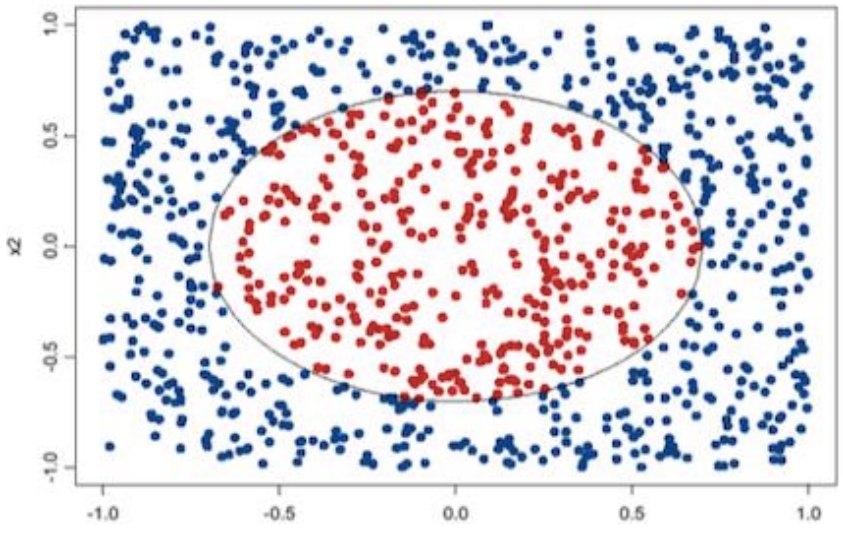
Breiman “Bagging Predictors” Berkeley Statistics Department TR#421, 1994

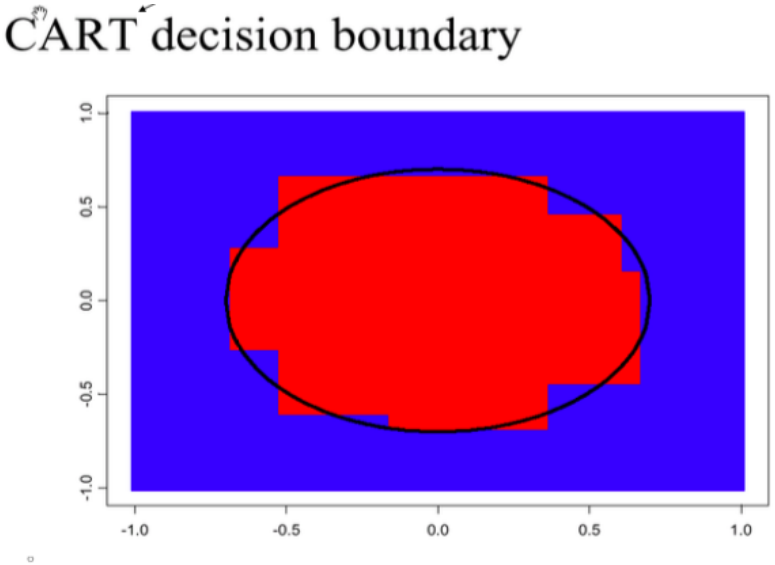
Ilustracja perturbacji i różnych drzew

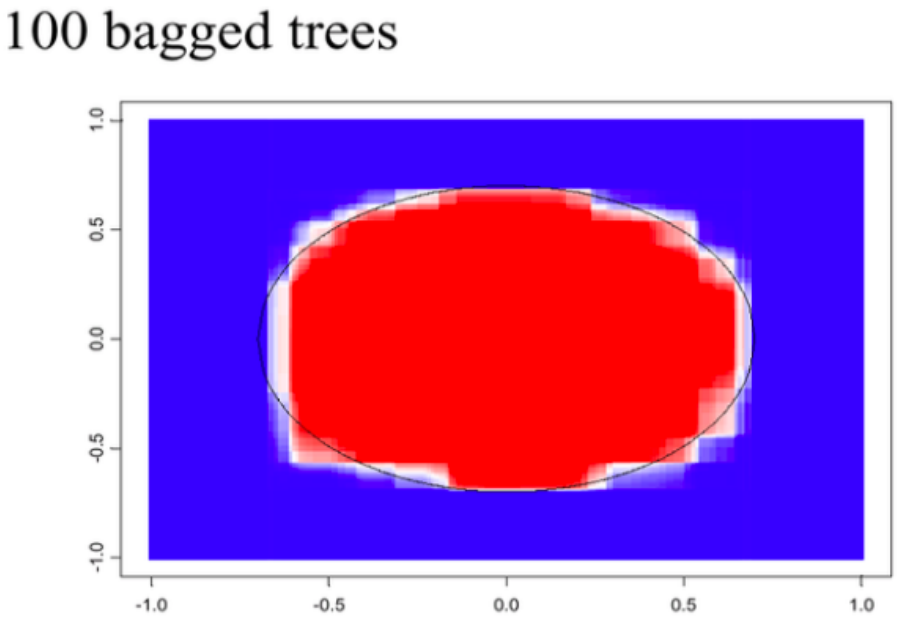


from Hastie, Tibshirani, Friedman: The Elements of Statistical Learning, Springer Verlag 2001

Inne ilustracja graficzna agregacji bagging







Bagging – dlaczego może działać?

- Liczne studia eksperymentalne (zwłaszcza dla drzew) – poprawa trafności
- Tzw. małe perturbacje w zbiorze uczącym + hipoteza o tzw. niestabilnym algorytmie generującym klasyfikatory
- Cytat z prac Breiman
  - This approach works well for unstable algorithms:
  - Whose major output classifier undergoes major changes in response to small changes in learning data.
- Typowe niestabilne algorytmy – drzew, decision stumps, reguly, liniowa regresja
- Inne spojrzenie na błąd – bias – variance decomposition
- Bagging naturalnie redukuje składnik wariancji

Bias-variance decomposition

- Oczekiwany błąd predykcji klasyfikatora
  - Dwa składniki: bias + variance
  - “The *bias* of a classifier” oczekiwany element błędu wynikający z założeń algorytmu uczącego, które nie pasują do problemu
  - “The *variance* of a classifier” wynika z rozważania konkretnego zbioru przykładów, który może wpływać na działanie klasyfikatora
- Najczęściej przetarg pomiędzy nimi:
  - niski bias => wyższa variance
  - niska variance => wyższy bias

Bagging głównie redukuje wariancje proporcjonalnie do liczby składników T oraz stopnie nieskorelowanie ich predykcji

Analiza teoretyczna zmian wariancji

Probability detour - Variance reduction by averaging

Let z_b, $b = 1, \ldots, B$ be identically distributed random variables with mean $\mathbb{E}[z_b] = \mu$ and variance $\mathrm{Var}[\sigma^2]$. Let ρ be the correlation between distinct variables.

Then,

$$\mathbb{E}\left[\frac{1}{B}\sum_{b=1}^{B} z_b\right] = \mu,$$

$$\mathrm{Var}\left[\frac{1}{B}\sum_{b=1}^{B} z_b\right] = \underbrace{\frac{1-\rho}{B}\sigma^2}_{\text{small for large } B} + \rho\sigma^2.$$

The variance is reduced by averaging (if $\rho < 1$) !

Analiza w próbach bootstrapowych

Bagging (I/II)

For now, assume that we have access to B **independent** datasets $\mathcal{T}^1, \ldots, \mathcal{T}^B$. We can then train a separate deep tree $\widehat{y}^b(\mathbf{x})$ for each dataset, $1, \ldots, B$.

- Each $\widehat{y}^b(\mathbf{x})$ has a **low bias** but **high variance**
- By averaging

$$\widehat{y}_{\text{bag}}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} \widehat{y}^b(\mathbf{x})$$

the bias is kept small, but variance is reduced by a factor B!

Analiza redukcji wariancji

Rozpatrzmy bagging regresyjny $f_{bag} = \frac{1}{B}\sum_{i=1}^{B} f_i^*$ oparty na próbie *D* (predyktory składowe nieskorelowane)

Wtedy

$E(f(x) - \hat{f}(x))^2$=Var$(\hat{f}(x))$+$(E\hat{f}(x) - f(x))^2$

$E(f(x) - \hat{f}_{bag}(x))^2$=Var$(\hat{f}_{bag}(x))$+$(E\hat{f}_{bag}(x) - f(x))^2$

$Var\left(\hat{f}_{bag}(x)\right) = \frac{1}{B}Var(\hat{f}(x))$

Wariancja zespołu (bagging) będzie B razy mniejsza niż wariancja pojedynczego predyktora (klasyfikatora)

Analiza wariancji baggingu

W rzeczywistości próby bootstrapowe są zależne więc zyskujemy mniej na uśrednianiu

$\mathrm{Var}(\hat{f}_{bag}(x)) = \frac{1}{B}\mathrm{Var}(\hat{f}(x)) + \frac{B(B-1)}{B^2} Cov(f_i f_j) =$

$= \frac{1-\rho}{B}\sigma^2 + \rho\sigma^2 < \sigma^2$

Drzewa regresji i redukcja wariancji

- Dla *T* składników naucz drzewa regresji
- Uśrednij wyniki predykcji

- Jeśli tworzymy niezredukowane (unpruned) drzewa, to będą się charakteryzowały większą wariancją i mniejszym obciążeniem (bias)
- Połączenie drzew w zespół bagging – zredukuje wariancje i częściowo bias

Bagging – decyzja zespołu

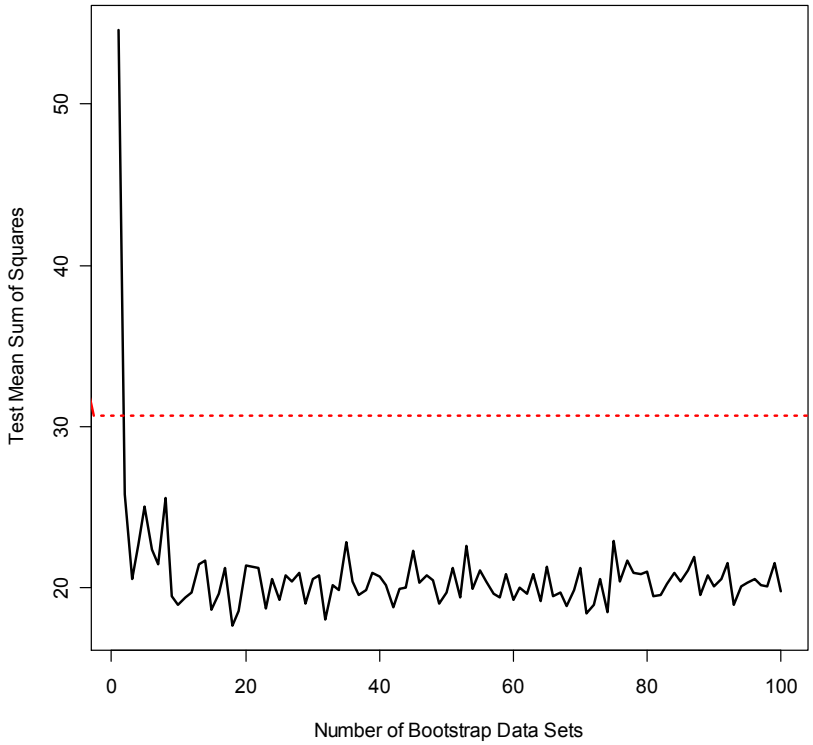
Wskazanie deterministyczne vs. prawdopodobieństwa

- Inne operatory agregacji niż max suma predykcji
- Standard – głosowanie z równymi wagami
- Głosowanie większościowe => każdy z klasyfikatorów ma różną wagę podczas agregacji “głosów” / predykcji
- Lecz jak ocenić wagę klasyfikatora składowego / jego kompetencje?
  - Globalnie, statycznie – oceń jego zdolności predykcyjne
  - Lecz, oszacowanie wymaga zbioru walidującego; czy jest inna alternatywa?
  - **Out-of-bag (OOB) estimate** (2/3 przykładów wylosowano do próbki bootstrapowej, lecz 1/3 pozostaje na zewnątrz.

Liczba składowych modeli

Czy redukcja oczekiwanego błędu zmienia się wraz ze wzrostem składników w bagging?

- Im więcej, tym lepiej ? Nie aż tak bardzo – Breiman wskazywał, że dla większości jego zbiorów danych 20-50 drzew wystarczało
- Trochę związane z wielkością i charakterystyką danych oraz miarami oceny



House prices data

Intepretowalność zespołu

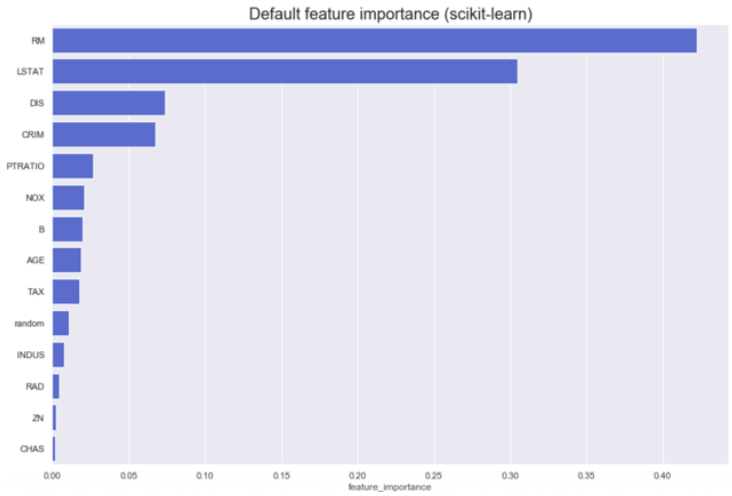
Bagging obejmuje wiele różnych drzew

- Pytanie – czy są dostatecznie zróżnicowane
- Lecz równocześnie tracimy interpretowalność która była osiągalna dla pojedynczego drzewa

Jak to rozwiązać? = pomysł L.Breimana / analiza feature importance

analizy struktury drzewa -> Oceń dla warunku redukcję miary (impurity) oraz wagę – liczbę przykładów w węźle

Permutations – “noised-up” method



Feature-Selection Ensembles

Niektóre dane/zadania– zbyt wiele cech

Pomysł: Użyj innego zbioru cech dla każdego z algorytmów (częste w przetwarzaniu tekstów, obrazów, sygnałów,…)

Przykład: Venus&Cherkauer (1996) zespół 32 sieci ANN, każda na innym podzbiorze cech – co doprowadziło do poprawy trafności

Propozycja: Random Subspace Methods autorstwa Ho.

Dla każdego klasyfikatora składowego– losowo wybierz podzbiór artrybutów nie zmieniając wyboru przykładów

Ho proponowała : m = 50% losowo wybranych atrybutów

Lattine (pozniej Stefanowski) – połączenie bootstrap sampling z losowym wyborem cech

Random forests [Breiman 2001]

Motywacja: Oprócz radzenia sobie z wysoką wymiarowością cech, dodatkowo zdekorelować / zróżnicować klasyfikatory składowe

Resampling przykładów jest niewystarczający

Pomysł: Dodatkowa perturbacja w tworzeniu drzewa

- Wykorzystaj losowanie bootstrapowe przykładów
- W każdym z węzłów drzewa losowo wybierz pozdbiór *m* cech z oryginalnie q cech i znajdź warunek podziału z wykorzystaniem kryterium Gini index lub entropii
  - Breiman proponuje $m = \text{sgrt}(q)$ dla drzew klasyfikacyjnych i $m = q/3$ drzew regresji
- Predykcje drzew agregowane tak jak w bagging

Intuicja losowego wyboru cech

- Załóżmy, że wśród *q* cech jest wyjątkowo silny predyktor wyjścia *y* oraz ew. Inne silne cechy
- Wtedy każde drzewo będzie używało tego predyktora w korzeniu drzewa, a inne na wysokich poziomach
- Drzewa będą zbyt podobne i skorelowane
- Agregacja, uśrednianie zbyt podobnych drzew nie zredukuje wariancji i nie poprawi predykcji
- Losowania podzbiorów cech – zapobiega powyższym ograniczeniom

Random forest pseudo-code

input S – learning set (n), T – no. of bootstrap samples, LA – learning algorithm

output C* - multiple classifier

for i=1 **to** T **do**

begin

S_i:=bootstrap sample from S – n examples ;

C_i:= learn tree from S_i with extra conditions

for each node

In each node select m out of the q input attributes uniformly at random

Choose the best split test among m attributes and split tree

until a stopping condition (may be max depth)

end;

$$C^*(x) = \text{argmax}_y \sum_{i=1}^{T} (C_i(x) = y)$$

Lub średnia predykcji dla wersji drzew regresji

Oceny eskperymentalne

- Praca Breiman, L., Random forests, Machine Learning, 2001, vol 45, 5-32
- Random forest często trafniejszy niż podstawowy boostrap bagging i konkurencyjny do Adaboost
- Jest mniej podatny na przeuczenie wobec trudnych danych

Random forests

Podatność na przeuczenie

- Czy wzrost liczby drzew składowych nie wpływa na nadmierne dostosowanie się do specyfiki zbiorów uczących?
- NIE!

Łatwa implementacja, równoległa oraz przydatna dla analizy “Big data”

Można także wylosowywać mniej przykładów do próby bootstrapowej

Cytat, We could bootstrap fewer than q features, say sqrt(q) useful for “big data” problems.

Interpretowalność – tak jak bagging

Bardzo dogodne dla analizy wysoko-wymiarowych danych

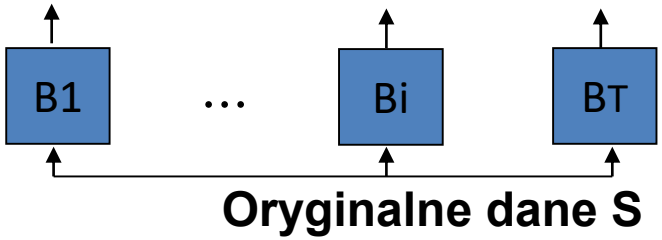
... Liczne zastosowania (być może najpopularniejszy zespół klasyfikatorów)

Uwagi nt. rozszerzeń

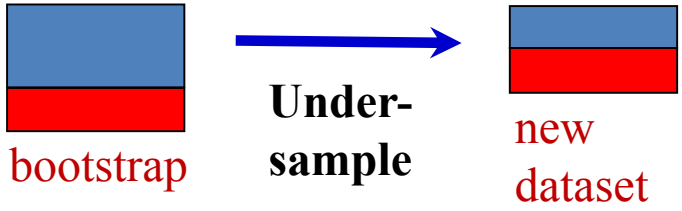
- Zarówno podstawowy bagging jak i random forest są często rozszerzane dla innych problemów (schemat jest elastyczny), np.
  - Online bagging dla uczenia przyrostowego i dalsze modyfikacja dla klasyfikacji zmiennych strumieni danych
  - Podstawy typ zespołu modyfikowany dla niezbalanasowanych danych
  - Równoległe implementacje dla Big Data
  - Metody specjalnych perturbacji do odkrywania nieskorelowanych, znaczących cech dla danych wysoce wielowymiarowych (np. bioinformatyczne eksperymenty)

Under-Bagging – dane niezbalansowane

- Standardowy Bagging → wykorzystuje boostraps
  - Losowanie przykładów ze zwracaniem
  - Nie rozwiązuje obciążenia w stronę klasy większościowej



Propozycje z Undersampling



- Exactly Balanced Bagging [Ch03]
  - Bootstraps = przekopiuj wszystkie przykładu mniejszościowe + wylosuj podobną liczbę przykładów większościowych (N_maj = N_min)
- Rough Balanced Bagging [Hido 09]
  - Inaczej wyrównuje prawdopodobieństwa klas w losowaniu do próbek bootstrapowych

Roughly Balanced Bagging

Hido S., Kashima H.: Roughly balanced bagging for imbalance data (2008)

Modyfikacja losowania

- Under-sampling – zmniejszanie liczności klas większościowej
- Zamiast ustawienia sztywnych liczności klas jak w EBB, wyrównać prawdopodobieństwa losowania w klasach – czyli na poziomie rozkładu prawdopodobieństwa
- Dla każdej T itertacji liczność klasy większościowej w próbie bootstrapowej Bs_{maj} jest zmienną losową określoną wg. negatywnego rozkładu dwumianowego

  For each bootstrap
  - Random size BS_{maj}
  - Wylosuj ze zwracaniem N_{min} oraz BS_{maj}

  Predykcja – odmiany losowania większościowego
- Przykładowe rozszerzenia:
  - Attribute Selection with RBBag dla wysoko wymiarowych danych
  - Multi-class generalization (zmiana rozkładów prawdopodobieństwa)

Lango M., Stefanowski J.: The Usefulness of Roughly Balanced Bagging for Complex and High-dimensional Imbalanced Data (2016)

Porównanie wielu zespołów klasyfikatorów

Studia eksperymentalne

Galar, Herrera et al [2011]

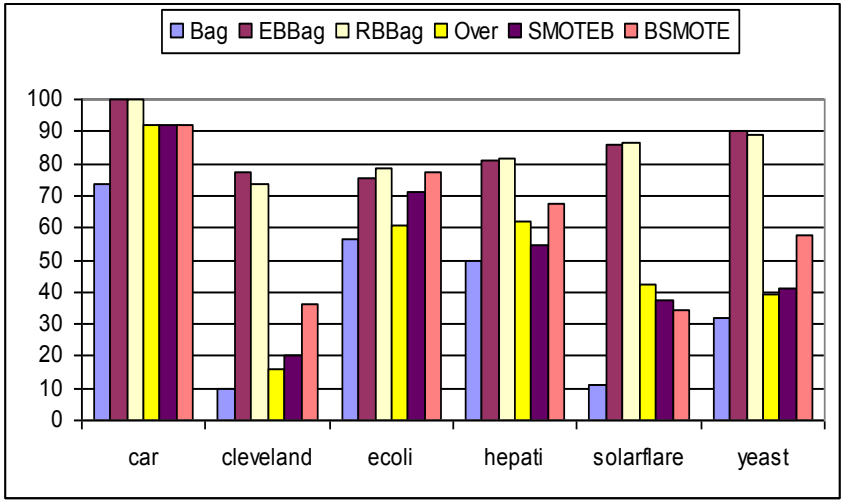
- Bagging działa lepiej niż inne zespoły w tym tzw. cost based

Khoshgoftaar et al. [2011]

- EBBag, RBBag lepsze niż SMOTEBoost and RUBoost

Własne studium [2013]

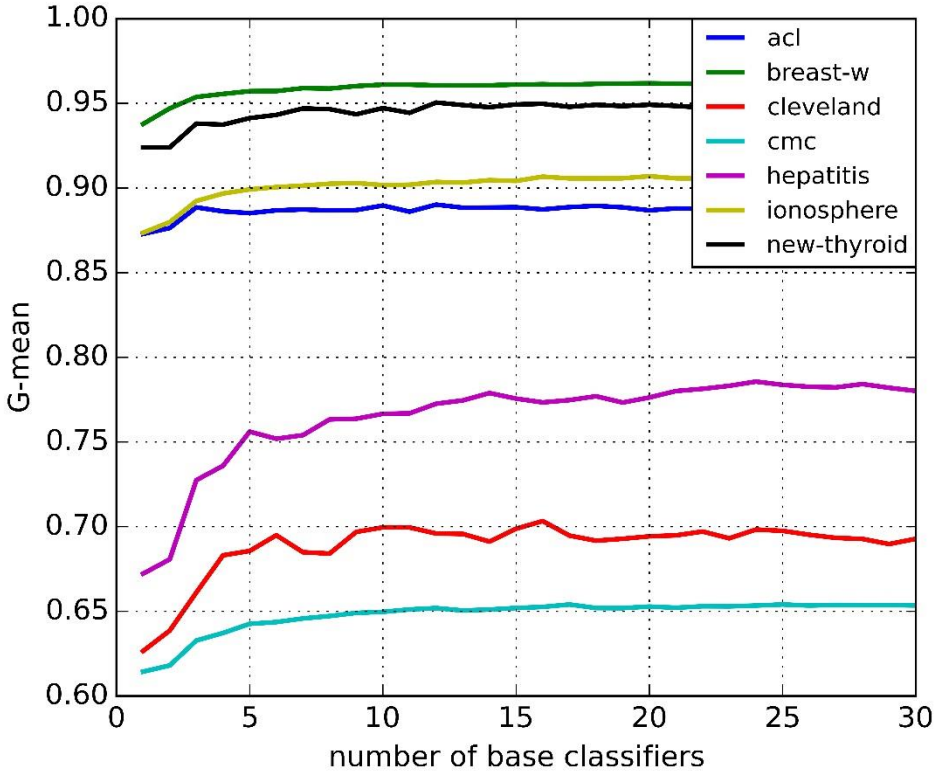
- **RBBag ≈> EBBag** >OverBag> SMOTEBag>Bagging

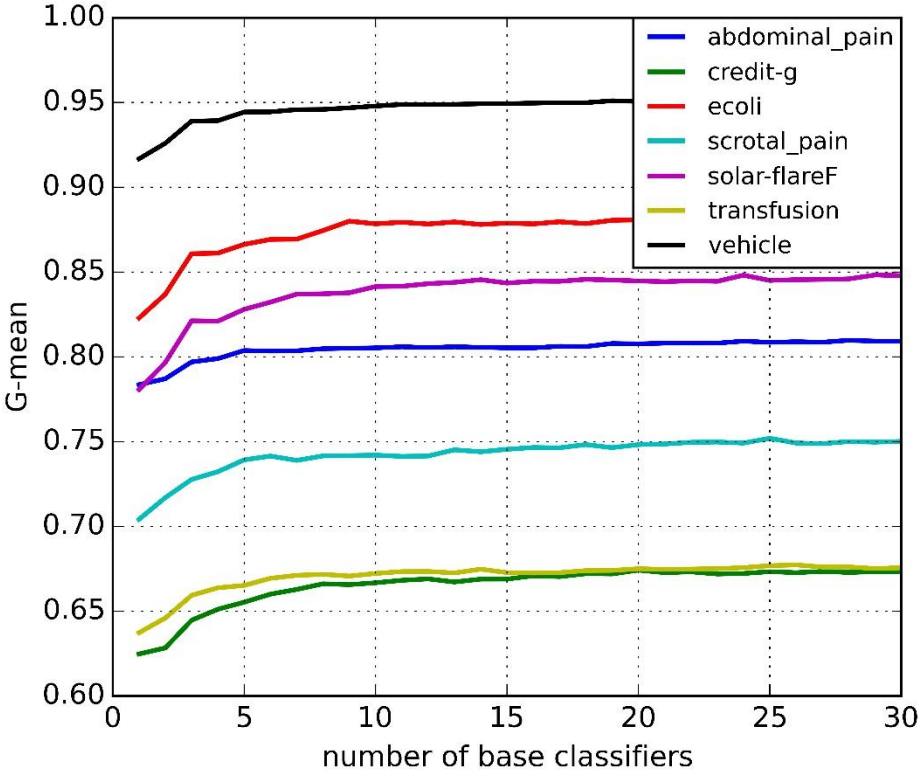


Dataset	Bag	EBBag	RBBag	OvBag	SmBag	BagSm
breast-w	95.88	96.03	96.37	96.23	95.88	96.77
abdominal-pain	78.95	80.65	80.35	79.44	80.85	79.86
acl	88.18	90.71	89.35	88.35	88.64	87.81
new-thyroid	92.41	96.91	96.58	95.36	95.18	92.89
vehicle	93.91	94.58	95.44	94.61	94.34	94.20
car	84.53	96.73	96.58	95.29	95.26	95.18
scrotal-pain	70.75	73.18	75.65	72.01	70.42	70.68
ionosphere	88.96	90.44	90.67	90.47	90.30	90.26
pima	71.54	74.22	75.64	73.54	72.33	71.38
credit-g	63.98	65.82	67.82	71.75	80.68	66.11
ecoli	68.67	72.24	88.85	51.42	58.38	80.11
hepatitis	62.81	78.93	78.66	72.16	68.47	74.29
haberman	43.11	65.41	63.43	58.11	60.02	62.82
breast-cancer	54.30	58.82	59.37	56.17	52.57	57.25
cmc	52.76	64.61	65.27	59.95	57.74	62.77
cleveland	12.61	72.32	71.02	22.77	25.03	50.96
hsv	0.00	36.27	35.74	2.84	5.37	16.61
abalone	49.58	78.93	79.32	61.95	63.67	69.65
postoperative	1.99	24.97	34.03	15.01	1.57	11.55
solar-flare	13.70	85.39	83.21	58.07	55.04	54.40
transfusion	55.72	66.75	67.32	64.83	63.96	65.76
yeast	51.48	84.55	84.68	59.70	59.41	57.94
balance-scale	0.00	59.07	54.23	1.40	0.00	0.67
average rank	5.61	1.96	1.61	3.65	4.26	3.91

J. Blaszczynski, J., Stefanowski: Extending bagging for imbalanced data. Proc. CORES 2013.

RBBag (liczba drzew decyzyjnych)





Relatywnie mała:

- Dla większości danych wystarczy kilkanaście

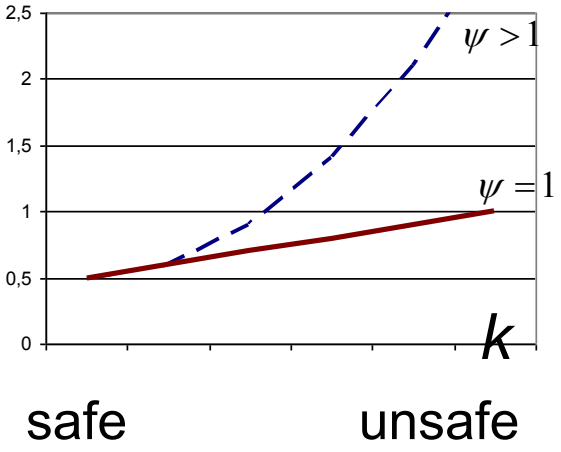
Neighbourhood Balanced Bagging

- Propozycja wykorzystujące inne zasady:
  - Zmodyfikuj prawdopodobieństwo losowania do próbki bootstrapowej z wykorzystaniem “safe level” przykładu
  - Zwiększ szanse wyboru przykładów mniejszościowych kosztem większościowych (global prob.)
- Poziom globalny
  - $p^1_{\min} = 1$ (mniejszościowa)
  - $p^1_{maj} = N_{\min} / N_{maj}$ (decrease $\rightarrow$ inverse global imbalance)
- Lokalny
  - Minority local neighb. $\psi \geq 1$

$$L = \frac{\left(N'_{maj}\right)^{\psi}}{k}$$

$$P_{global} \cdot P_{local}$$

- Eksperymenty – porównywalny do RBBag, lepszy dla b. trudnychdanych



J.Blaszczynski, J.Stefanowski: Neighbourhood sampling in bagging for imbalanced data. Neurocomputing (2015)

Odnośniki do literatury

- Intensywny rozwój od lat 90 poprzedniego wieku
- Wiele różnych propozycji
- Przykładowe pozycje:
  - L.Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, 2004 (large review + list of bibliography).
  - T.Dietterich, Ensemble methods in machine learning, 2000.
  - J.Gama, Combining classification algorithms, 1999.
  - G.Valentini, F.Masulli, Ensemble of learning machines, 2001 [obszerna lista referencyjna]
  - R.Polikar,Ensemble based systems in decision making, IEEE Circuits and Systems Magazine, vol. 6, no. 3, pp. 21–45, 2006.
  - See also many papers by L.Breiman, J.Friedman, Y.Freund, R.Schapire, T.Hastie, R.Tibshirani,
  - W Polsce – przykładowo prace M.Woźniak i współpracownicy

Pytanie i komentarze?

Dalszy kontakt:

jerzy.stefanowski@cs.put.poznan.pl

http://www.cs.put.poznan.pl/jstefanowski/

Fundusze Europejskie Polska Cyfrowa

Rzeczpospolita Polska

Unia Europejska Europejski Fundusz Rozwoju Regionalnego