

SYSTEMY UCZĄCE SIĘ

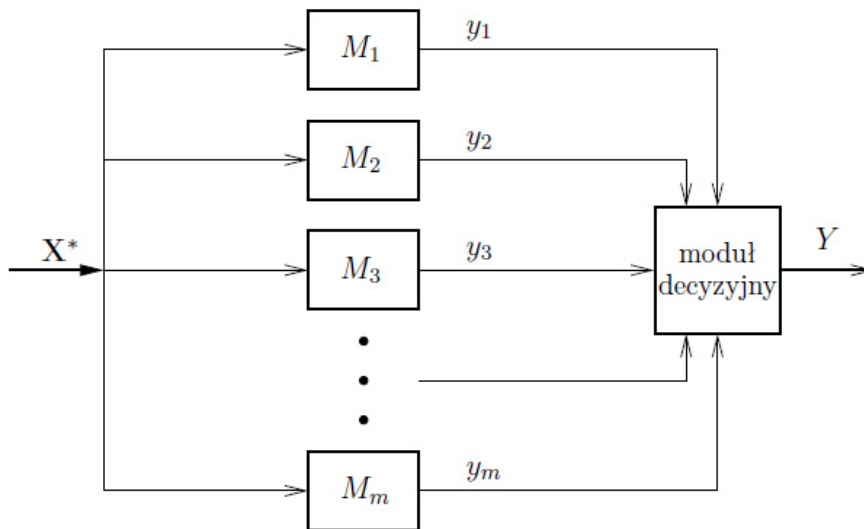
WYKŁAD 12. KOMITETY KLASYFIKATORÓW

Dr hab. inż. Grzegorz Dudek
Wydział Elektryczny
Politechnika Częstochowska

Częstochowa 2014

Komitet (*Ensemble, Committee, Multiple Classifier Systems – MCS*) to symboliczna nazwa zbioru klasyfikatorów (modeli), które, pracując wspólnie nad tym samym zadaniem, mają dać wynik lepszy, niż wyniki otrzymywane przez pojedyncze modele.

Pojedyncze modele często są niestabilne (małe zaburzenie zbioru treningowego może prowadzić do całkowicie innych granic decyzji), stochastyczne elementy w metodach uczenia (np. przypadkowa inicjalizacja wag początkowych sieci neuronowej) w połączeniu z procedurą minimalizacyjną funkcji błędu, często dają różne wyniki. Komitet ma być pozbawiony tych słabości, jest mniej podatny na zaburzenia i to stanowi jego przewagę.



Rysunek 24: Ogólny schemat budowy i działania komitetu składającego się z m klasyfikatorów. Nieznany wektor \mathbf{X}^* zostaje przeanalizowany przez wszystkich członków komitetu podejmujących własne decyzje y . Końcowy rezultat klasyfikacji Y jest zawsze wynikiem założonej filozofii działania komitetu realizowanej przez moduł decyzyjny.

1. Statyczne lub dynamiczne (*combination* lub *selection*).

W komitetach statycznych wszystkie modele mają jednakowy wkład do końcowej decyzji, w dynamicznych natomiast wybieramy zawsze jeden najbardziej kompetentny model, często nazywany ekspertem, który podejmuje decyzję za wszystkich.

2. Komitety oparte na głosowaniu (*voting*) lub prawdopodobieństwach (*non-voting*).

W głosowaniu zliczamy końcowe decyzje każdego modelu lub tworzymy ich kombinację liniową (może być ważona) i na jej podstawie tworzymy końcowy głos. W metodach *non-voting* zamiast na decyzjach bazujemy na prawdopodobieństwach $P(C|\mathbf{X})$ i na ich podstawie podejmujemy decyzję.

3. Komitety modeli niezależnych lub uczonych równocześnie.

- W pierwszym przypadku każdy model wchodzący w skład komitetu jest tworzony i trenowany niezależnie – bez świadomości istnienia pozostałych modeli, jego głos (końcowa klasyfikacja) zostaje włączony do ostatecznej decyzji. Umożliwia to budowanie komitetów z klasyfikatorów różnych typów, nawet takich, w których nie występuje uczenie. Minusem takiego podejścia jest niebezpieczeństwo, iż każdy model może wyspecjalizować się w tym samym obszarze danych – nie mając informacji, iż wcześniej dokonał tego inny model.
- W uczeniu równoczesnym modele wzajemnie oddziałują na siebie podczas procesu uczenia (zamiast uczenia pojedynczych modeli mamy *de facto* uczenie całego komitetu połączone z uczeniem modeli). Teoretycznie to podejście powinno zapewniać lepsze dopasowanie się do danych i zapobiec brakowi równomierności w tym procesie. Zmuszeni zostajemy jednak do używania modeli jednego rodzaju (np. tylko sieci neuronowe), z reguły proces uczenia takiej struktury jest także bardzo długotrwały.

4. Komitety homo- lub heterogeniczne – dzielimy komitety na składające się z klasyfikatorów tego samego rodzaju lub nie wprowadzamy żadnych ograniczeń. Pierwsze z podejść, wywodzące się z początków dziedziny, jest do dziś zdecydowanie powszechniejsze.
5. Ze względu na sposób przetwarzania danych rozróżniamy komitety, w których każdy z członków w fazie uczenia pracuje na tym samym zbiorze danych, oraz takie, w których pojedyncze klasyfikatory pracują na różnych podzbiorach jednego zbioru głównego.

Przy tworzeniu komitetu należy zwrócić uwagę na:

- **Różnorodność** – cecha bardzo potrzebna. Jeśli w Komitecie występują klasyfikatory dokonujące identycznych klasyfikacji dla wybranych podzbiorów danych, nie jest możliwe, by wynik komitetu mógł być lepszy od wyniku pojedynczego modelu. Tylko tam, gdzie występują błędy modeli w różnych miejscach – symbolizowanych przez podprzestrzenie przestrzeni cech – można otrzymać zwiększoną dokładność klasyfikacji (choć całkowita dokładność poszczególnych modeli może być jednako-
wa). Istnieją różne sposoby szacowania różnorodności modeli: definicja mówi, iż dwa klasyfikatory są różne, jeśli popełniają różne błędy dla nowych danych. Bardzo użytecznym narzędziem badania różnorodności jest także macierz konfuzji.
- **Wydajność** – działanie komitetu musi być szybkie, nie pochłaniać dużej ilości czasu i mocy obliczeniowej. Samo nauczenie i działanie pojedynczych klasyfikatorów jest już zabiegiem kosztownym, dlatego komitet nie powinien znacząco zwiększać czasu obliczeń.

- Dokładność – wspomniane już wcześniej założenie, iż ogólny wynik komitetu powinien być lepszy niż wynik klasyfikacji poszczególnych członków.

Kolejnym ważnym zagadnieniem jest dobór liczby członków komitetu. Duża liczba modeli to wzrastające koszty obliczeniowe, zarówno działania komitetu, jak i uczenia modeli. Zbyt mała ich liczba może nie zapewnić pożądanej poprawy dokładności. Generalnie powinno się preferować jak najmniejszą liczbę modeli, a rozwiązanie powinno być kompromisem pomiędzy czasem uczenia komitetu a jego dokładnością.

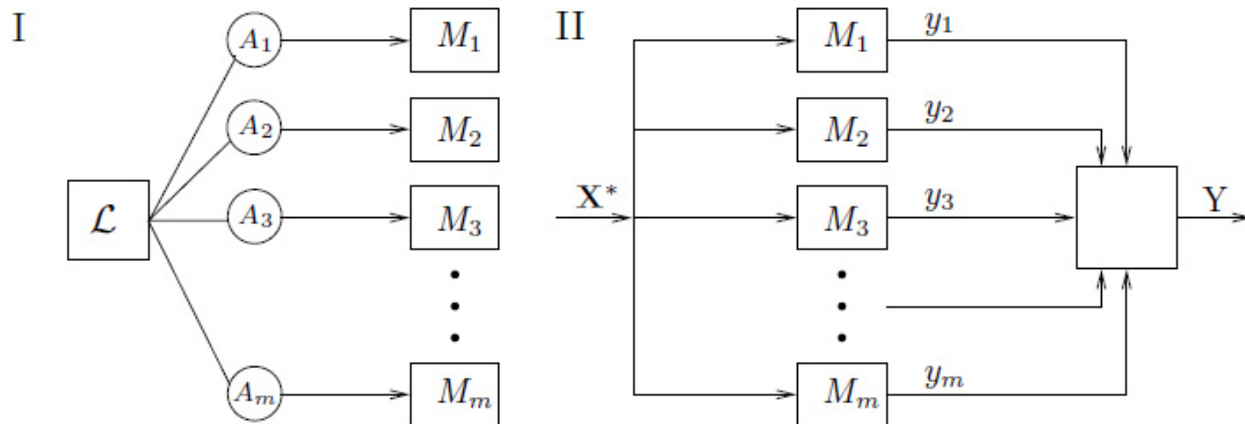
KOMITETY OPIERAJĄCE SIE NA STATYSTYCZNYM PRÓBKOWANIU DANYCH

Dalej przedstawione będą komitety opierające się na statystycznym próbkowaniu danych

Metody bazujące na statystycznym próbkowaniu (nazywanego także manipulowaniem) zbioru treningowego polegają na generowaniu serii klasyfikatorów pracujących na różnych podzbiórach zbioru głównego (rysunek 27). Metody te polecane są zwłaszcza dla niestabilnych algorytmów (praca na różnych podzbiórach ma wykluczyć znaczący wpływ przypadkowych zmian w zbiorze głównym na wynik), często wspomina się, iż polecane są dla algorytmów działających słabo (*weak learner*) – dla nich też osiągają najlepsze efekty.

Generowanie modeli wymaga bezpośredniego dostępu do algorytmu uczącego, jeśli jest on kosztowny obliczeniowo, koszt działania komitetu będzie do niego proporcjonalny. Nie występuje jednak jakakolwiek ingerencja w algorytm uczący, zróżnicowanie opiera się na zbiorach uczących, na których działają poszczególne algorytmy. Dużym atutem takiego rozwiązania jest możliwość zastosowania wszystkich dostępnych algorytmów.

KOMITETY OPIERAJĄCE SIE NA STATYSTYCZNYM PRÓBKOWANIU DANYCH



Rysunek 27: Schemat komitetu wykorzystującego próbkowanie danych. Działanie komitetu składa się z dwóch etapów, I: generowanie klasyfikatorów (za pomocą algorytmów A) działających na podzbiórach zbioru treningowego \mathcal{L} , II: klasyfikacja, czyli podejmowanie końcowej decyzji o wektorze X^* .

Bagging

Jest to jeden z najprostszych pomysłów [4], który opiera się na idei wprowadzonej w *bootstrappingu* (stąd nazwa *Bootstrap Aggregation*). Z głównego zbioru danych generujemy kilka podzbiorów i uczymy każdy ze składników komitetu na oddzielnym podziorze. Dla m klasyfikatorów oraz zbioru uczącego \mathcal{L} składającego się z N wektorów tworzy się m zbiorów uczących, w każdym znajduje się N wektorów losowo wybranych z \mathcal{L} . Prawdopodobieństwo wyboru każdego wektora jest równe i wynosi $\frac{1}{N}$, w niektórych zbiorach pewne wektory mogą być umieszczone więcej niż raz, inne wcale. Każdy model uczony jest według tego samego algorytmu, każdy proces uczenia przebiega niezależnie. Końcowa decyzja dla zagadnienia klasyfikacji podejmowana jest przez głosowanie większościowe.

Boosting

Opisany w [16], jeden z najwydajniejszych i najczęściej stosowanych pomysłów mający na celu zwiększenie dokładności dowolnego algorytmu uczenia. Od *baggingu* różni się tym, iż następny klasyfikator pracuje na błędach pozostałego. Wektory źle klasyfikowane przez poprzedni model mają większe znaczenie w procesie uczenia następnego modelu, każdy proces uczenia przebiega niezależnie.

Początkowa wersja komitetu zakładała istnienie trzech modeli. Uczenie zaczynamy od modelu M_1 i filtrujemy zbiór danych treningowych tak, by nauczyć na utworzonych podzbiorach dwa dodatkowe modele: M_2 i M_3 . M_1 jest modelem głównym, którego klasyfikację chcemy zwiększyć, kolejne kroki algorytmu są następujące:

1. Trenujemy M_1 na zbiorze treningowym \mathcal{L} . Uczenie przebiega tak długo, aż napotkany zostanie pierwszy wektor niedający się poprawnie sklasyfikować (wystąpi błąd).

2. Wówczas błędny wektor umieszczamy w zbiorze treningowym drugiego modelu. Wracamy do punktu pierwszego i powtarzamy ten etap tak długo, aż w zbiorze treningowym drugiego modelu znajdzie się dostatecznie duża liczba wektorów.
3. Rozpoczynamy uczenie modelu M_2 .
4. Klasyfikujemy pozostałe wektory przez modele M_1 i M_2 , jeśli otrzymujemy sprzeczne wyniki dla danego wektora, umieszczamy go w zbiorze treningowym dla modelu M_3 .
5. Zgromadzenie dostatecznie dużej liczby wektorów dla modelu M_3 rozpoczyna jego proces uczenia. Wraz z zakończeniem uczenia modelu M_3 uczenie komitetu zostaje zatrzymane.

Ostateczna decyzja zostaje podjęta przez głosowanie większościowe i prowadzi do kombinowanych granic decyzji (rysunek 28).

AdaBoost

Popularną odmianą *boostingu* jest *AdaBoost* (*Adaptive Boosting*), algorytm, w którym przechowujemy serię wag przypisanych do każdego wektora ze zbioru treningowego. Wagi tworzą zbiór \mathcal{D}_t – jest to główna cecha algorytmu. Początkowo waga każdego wektora jest równa, w miarę postępowania procesu uczenia wagi wektorów klasyfikowanych błędnie zostają zwiększane. Na podstawie zbioru wag tworzymy nowy zbiór danych (np. poprzez odrzucenie wektorów, których wagi nie przekraczają pewnego progu) – w ten sposób kolejny uczony klasyfikator może skoncentrować się na trudniejszych (z punktu widzenia klasyfikacji) przypadkach.

Arcing

Arcing [3] jest odmianą baggingu, nazwa metody pochodzi od słów *Adaptively Resample and Combine*. Mając N elementów w zbiorze treningowym, początkowo losujemy wektory z prawdopodobieństwem $p = \frac{1}{N}$. Oznaczamy wektory błędnie klasyfikowane i następane tworzone klasyfikatory pracują już na zbiorach danych, w których prawdopodobieństwo wylosowania wektora błędnego jest większe (w wyniku skalowania prawdopodobieństw $p \mapsto \beta p$, $\beta > 1$). W ten sposób nowe modele przystosowują się bardziej do obszarów źle klasyfikowanych przez poprzedników. Końcowe prawdopodobieństwo powstaje poprzez proste lub ważone głosowanie.