Matuszek Albert 13.12.2009

Krupczak Łukasz

Sprawozdanie z laboratorium z przedmiotu

Metody Rozpoznawania Obrazów

Temat: **Porównanie metod uczenia nadzorowanego, nienadzorowanego i semi – nadzorowanego**

# Opis zbiorów danych

## Analiza wiadomości e-mail pod kontem spamu

Zbiór danych składający się z 4601 elementów opisujący wiadomości e-mail, który pozwala sprawnie klasyfikować wiadomości, jako spam lub nie-spam. Opis wiadomości stanowi 57 liniowych atrybutów:

* 47 atrybutów jest wartościami typu real z przedziału [0,100] opisującymi % wystąpień danych słów kluczowych w emailu w stosunku do liczby słów
* 6 atrybutów jest wartościami typu real z przedziału [0,100] opisującymi % wystąpień danych liter kluczowych w emailu w stosunku do liczby liter
* 1 atrybut opisuje ilość wystąpień ciągów dużych liter
* 1 atrybut opisujący średnią długość ciągu dużych liter
* 1 atrybut opisujący najdłuższy ciąg dużych liter
* 1 atrybut opisujący sumę wszystkich ciągów dużych liter

Zbiór został podzielony na 4 podzbiory:

* TR75 – 3450 elementów - treningowy
* TR50 – 2300 elementów – treningowy, bez informacji o klasach
* TR25 – 1150 elementów - treningowy
* TE25 – 1151 elementów – testowy

## Analiza kardiologiczna serca w poszukiwaniu odstępstw od normy

Zbiór danych składający się z 267 przykładów opisujący tomografię serca (SPECT), który pozwala klasyfikować strukturę mięśnia, jako poprawną lub z zaburzeniami (odstępstwami od normy). Zbiór zawiera 22 binarne (wyekstrahowane początkowo z 44 liniowych) atrybuty przyporządkowane dla każdego zdjęcia (brak informacji, co oznaczają). Zbiór został podzielony na 4 podzbiory:

* TR75 – 200 elementów - treningowy
* TR50 – 132 elementów – treningowy, bez informacji o klasach
* TR25 – 68 elementów - treningowy
* TE25 – 67 elementów - testowy

## Rozpoznawanie typu raka (łagodny/złośliwy)

Zbiór danych składający się z 569 elementów opisujący raka, który pozwala sprawnie klasyfikować typ raka jako złośliwy lub zwykły.

Opis charakterystyki stanowi 9 atrybutów (zakres liczbowy nieciągły [1,10]):

* Clump Thickness
* Uniformity of Cell Size
* Uniformity of Cell Shape
* Marginal Adhesion
* Single Epithelial Cell Size
* Bare Nuclei
* Bland Chromatin
* Normal Nucleoli
* Mitoses

Pozwoliliśmy sobie nie tłumaczyć ich znaczeń ze względu na zawiłości medycznej terminologii. Cechy te zostały obliczone na podstawie cyfrowych zdjęć rentgenowskich (FNA). Opisują one właściwości struktury zwyrodnieniowej (rakowej).

Zbiór został podzielony na 4 podzbiory:

* TR75 – 426 elementów - treningowy
* TR50 – 283 elementów – treningowy, bez informacji o klasach
* TR25 – 143 elementów - treningowy
* TE25 – 143 elementów - testowy

# Parametry metod uczenia

## AdaBoost:

Jedynym parametrem algorytmu AdaBoost jest ilość iteracji, w których są poprawiane wagi, wartość tą ustawiliśmy metodą prób i błędów na 10 iteracji.

## SemiBoost:

Dla tego algorytmu również przyjęto liczbę iteracji równą 10. Parametr C, określający ważność nieoznaczonych próbek ustawiliśmy równy 1. Zauważyliśmy, że zmiana tego parametru nie powoduje większych zmian w działaniu algorytmu. Do uczenia słabego algorytmu klasyfikującego używamy cały dostępny zbiór danych etykietowanych oraz 20% danych nieetykietowanych.

## Klasteryzacja:

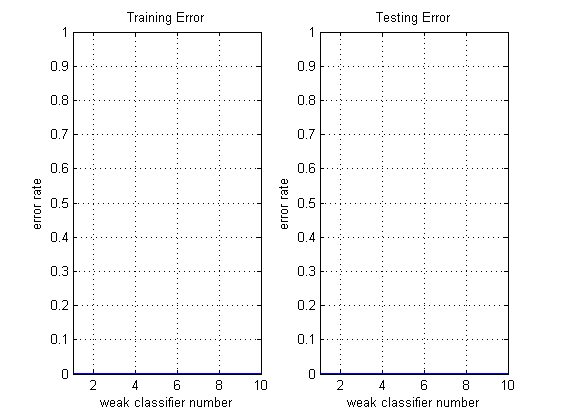
Użyto do testów dwóch wariantów:

* „Complete”: oparty na analizie największej odległości między sąsiadami (idea: w każdej iteracji dla najbardziej odległego sąsiada jest poszukiwana nowa klasa)
* „Ward”: metoda najmniejszych kwadratów , metoda ta daje wynik o najmniejszej sumie kwadratów błędów. Nie ma jednak gwarancji, że wynik ten ma jakikolwiek praktyczny sens. W szczególności, jeśli w danych występuje dużo elementów odstających, rezultaty mogą nie mieć nic wspólnego z rzeczywistą linią trendu, czy zależnością między zjawiskami opisywanymi przez zmienne losowe.

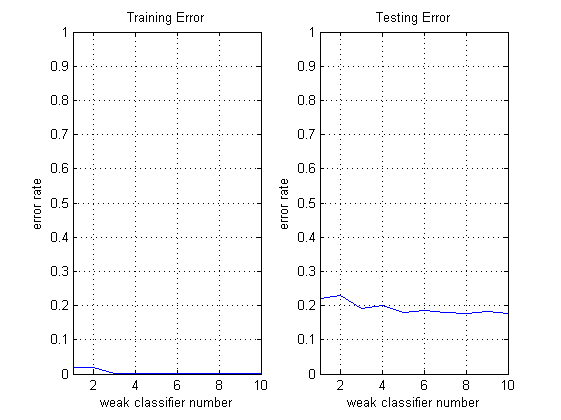
# Wyniki

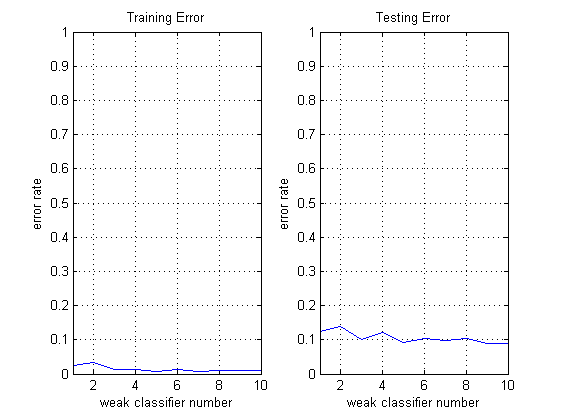
## Analiza wiadomości e-mail pod kontem spamu

Algorytm AdaBoost dla zbioru TR25 (pole pod ROC = 0,5):



Algorytm AdaBoost dla zbioru TR75(pole pod ROC = 0):



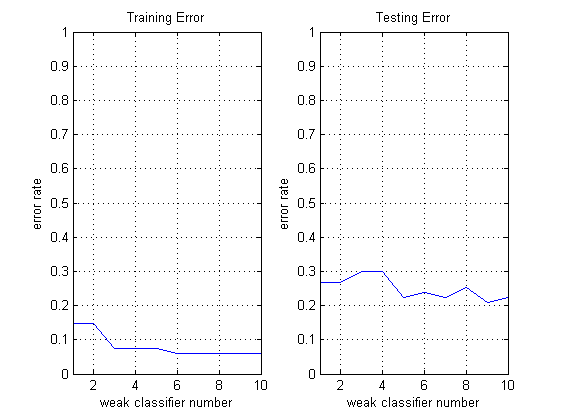
Algorytm SemiBoost nauczony na zbiorach TR25 i TR50.

Klasteryzacja aglomeratywna dla zbioru TR25:

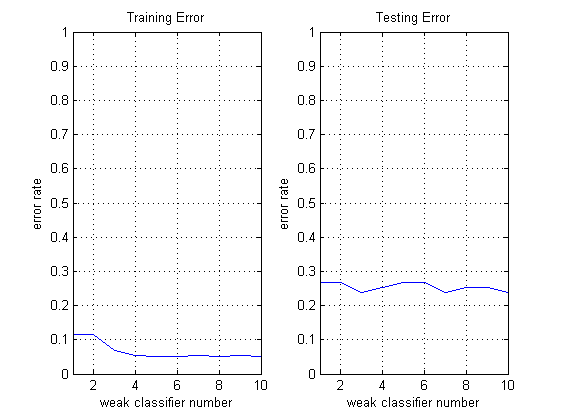
* Metoda „ward”: 905 poprawnie przyznanych klas na 1151 elementów
* Metoda „complete”: 1150 poprawnie przyznanych klas na 1151 elementów

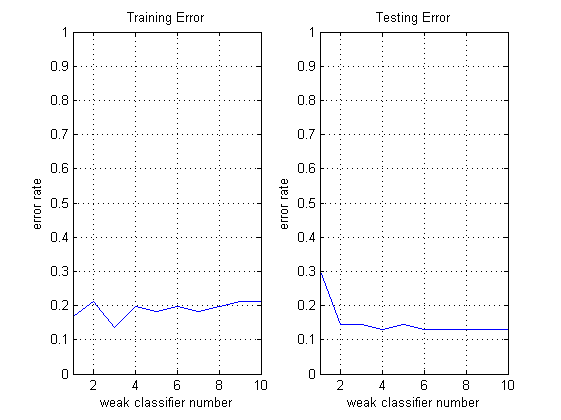
## Analiza kardiologiczna serca w poszukiwaniu odstępstw od normy

Algorytm AdaBoost dla zbioru TR25 (pole pod ROC = 0,2837):



Algorytm AdaBoost dla zbioru TR75 (pole pod ROC = 0,2371):



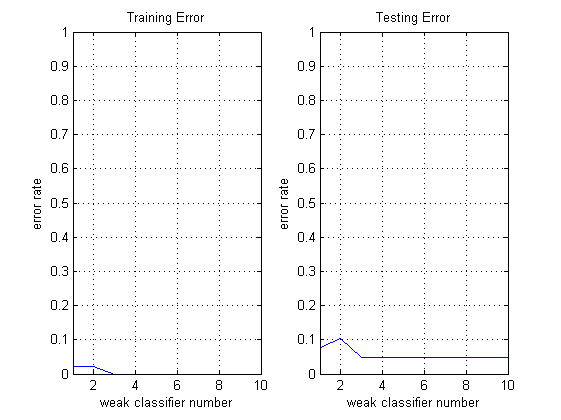
Algorytm SemiBoost nauczony na zbiorach TR25 i TR50.

Klasteryzacja aglomeratywna dla zbioru TR25:

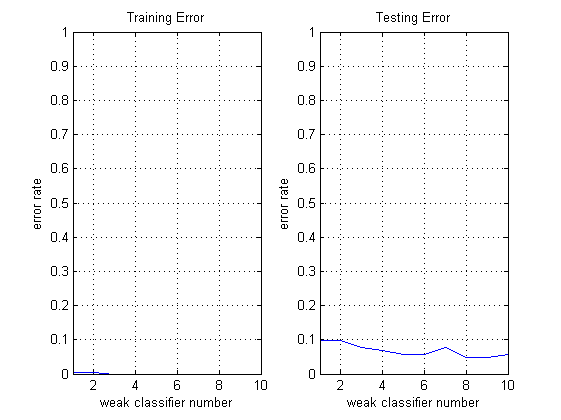
* Metoda „ward”: 143 poprawnie przyznanych klas na 143 elementy
* Metoda „complete”: 143 poprawnie przyznanych klas na 143elementy

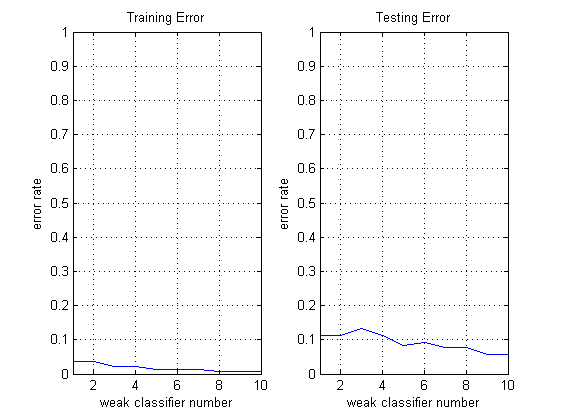
## Rozpoznawanie typu raka (łagodny/złośliwy)

Algorytm AdaBoost dla zbioru TR25 (pole pod ROC = 0,2086):



Algorytm AdaBoost dla zbioru TR75 (pole pod ROC = 0,3451):



Algorytm SemiBoost nauczony na zbiorach TR25 i TR50.

Klasteryzacja aglomeratywna dla zbioru TR25:

* Metoda „ward”: 19 poprawnie przyznanych klas na 67 elementów
* Metoda „complete”: 47 poprawnie przyznanych klas na 67 elementów

# Analiza wyników

## Analiza wiadomości e-mail pod kontem spamu

* AdaBoost:

Dla zbioru treningowego TR25 i testowego TE25 od pierwszej iteracji otrzymany wynik nie zawierał błędów. Dla zbioru treningowego TR75 i testowego jak wyżej, podczas uczenia po 3 iteracji uzyskujemy zero błędów natomiast dla zbioru testowego błąd wacha się w granicach 20% z tendencją spadkową. Wynik ten jest o tyle ciekawy, że zbiór treningowy TR25 zawiera się w zbiorze TR75 co by wskazywało na przeuczenie się algorytmu co skutkuje spadkiem ilości poprawnie sklasyfikowanym e-maili (należy dodać że zbiór TE25 nie zawiera się w zbiorze TR75 ani tym bardziej TR25 – tyczy się to także punktów poniżej).

* SemiBoost:

Można zauważyć, że błąd treningowy jest od 3 iteracji praktycznie cały czas na podobnym poziomie. Natomiast błąd testowy maleje( z pewnymi skokami) od wartości ok. 0.11 do 0.9.

## Analiza kardiologiczna serca w poszukiwaniu odstępstw od normy

AdaBoost:

Dla zbioru treningowego TR25 i testowego TE25 ilość błędów przy uczeniu się spada z poziomu 15% (pierwsza iteracja) do ok. 8% już przy 6 iteracji. Ilość błędów przy testach wacha się pomiędzy 30 a 20% z tendencją spadkową. Dla zbioru treningowego TR75 i testowego jak wyżej ilość błędów przy uczeniu się spada z poziomu 11% (pierwsza iteracja) do ok. 5% już przy 4 iteracji. Ilość błędów przy testach utrzymuje się w granicy 25%.

* SemiBoost:

Można zayważyć, że błąd treningowy utrzymuje się na mniej więcej stałym poziomie (ok. 0.2), natomiast błąd testowy zmalał ponad 2-krotnie(z ok. 0.3 do 0.12). Zatem błąd testowy jest niższy od błędu treningowego.

## Rozpoznawanie typu raka (łagodny/złośliwy)

* AdaBoost:

Dla zbioru treningowego TR25 i testowego TE25 ilość błędów podczas uczenia się od 3 iteracji wynosi 0. Ilość błędów przy testach utrzymuje się na poziomie 5% od 3 iteracji. Dla zbioru treningowego TR75 i testowego jak wyżej ilość błędów przy uczeniu od 3 iteracji wynosi 0. Ilość błędów przy testach utrzymuje się w granicy 10 a 5% z tendencją spadkową.

* SemiBoost:

W tym przypadku błąd treningowy maleje liniowo od wartości 0.03 do 0.01. Z kolei błąd testowy wzrósł po 3 iteracjach z 0.11 do 0.13, aby potem spadną do 0.6. Można powiedzieć, że algorytm bardzo dobrze poradził sobie z tym problemem.

Przed przeprowadzeniem eksperymentów sądziliśmy, że wyniki uczenia algorytmu Ada na zbiorze TR75 będę lepsze od wyników uczenia na zbiorze TR25 (gdyż w TR75 mamy do dyspozycji więcej przykładów). Okazało się jednak, że do rozpoznania 25% przykładów wystarczyło się nauczyć na 25% innych przykładów.

Algorytm SemiBoost posiadał informację o klasach 25% przypadków oraz posiadał same atrybutu 50% przypadków testowych. Taki dobór danych okazał się całkiem sensowny, gdyż dane nieetykietowane nie pozwalały się „przeuczać” algorytmowi. Algorytm SemiBoost okazał się lepszy od AdaBoost trenowanym na zbiorze TR75 oraz minimalnie gorszy AdaBoost trenowanym na zbiorze TR25.

## Klasteryzacja aglomeratywna

Na załączonych wynikach widać, że klasteryzacja daje bardzo dobre wyniki, można powiedzieć, że są one porównywalne z wynikami uzyskanymi algorytmem AdaBoost. Zdecydowanie lepsze wyniki (porównywalne z AdaBoost) zapewnia metoda „complete”. Metoda „ward” dla niektórych zbiorów (analiza serca i spamu) działa porównywalnie dobrze, a dla niektórych gorzej (analiza raka). Wynik taki był do przewidzenia, ponieważ metoda klasyfikacji użyta w wariancie „ward” jest mniej dokładna i bardziej podatna na błędy wynikające z danych odbiegających od średniej dla danej klasy. Najgorzej klasteryzacja sprawdza się dla analizy raka, ale wynika to z najmniejszego zbioru danych i algorytm klastrujący nie był wstanie się dostatecznie dobrać parametrów dla poprawnego rozróżnienia klas. Ogólny zaobserwowany trend jest taki, że im mamy więcej danych treningowych tym później dla danych testowych klasteryzacja jest dokładniejsza.

# Wnioski

Można zauważyć, że algorytm AdaBoost zdecydowanie lepiej się sprawdza, gdy mamy mniejszy zbiór treningowy, dla większego następuje najprawdopodobniej przeuczenie się algorytmu. Dokładnie odwrotny wniosek natomiast można wyciągnąć po analizie algorytmów, klastrujących gdzie zwiększenie zbioru danych powodowało poprawienie klastrowania. Wynika to także z tego, że algorytmy klastrujące zaimplementowane w Matlabie są bardzo zaawansowane i dopracowane.

Algorytm SemiBoost okazał się lepszy dla danych „nowych”, czyli niedostępnych w trakcie nauki. Wynika to z tego, iż do nauczenia tego algorytmu wykorzystano także próbkę danych nieetykietowanych, które niepozwalany algorytmowi za bardzo dopasować się do zbioru treningowego.