KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Instrukcja współfinansowana przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
w projekcie

*„Innowacyjna dydaktyka bez ograniczeń
– zintegrowany rozwój Politechniki Łódzkiej – zarządzanie Uczelnią,
nowoczesna oferta edukacyjna i wzmacniania zdolności
do zatrudniania osób niepełnosprawnych”*

Instrukcja jest dystrybuowana bezpłatnie

Instrukcja do laboratorium, część 2

dr inż. Artur Klepaczko

Eksploracja danych

Zadanie nr 13 – Studia podyplomowe „Przetwarzanie i analiza obrazów biomedycznych”

Politechnika Łódzka
Instytut Elektroniki

90-924 Łódź, ul. Żeromskiego 116,
tel. 042 631 28 83
www.kapitalludzki.p.lodz.pl

Spis treści

1 Wstęp

Obecne zajęcia laboratoryjne są poświęcone zagadnieniom klasyfikacji wektorów danych w przypadku, gdy odrębne klasy można rozdzielić za pomocą funkcji liniowej. Procedura uczenia klasyfikatora sprowadza się tutaj do skonstruowania granicy decyzyjnej opisanej równaniem

$$y(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + w_0,$$

czyli do wyznaczenia parametrów $\mathbf{w}$ (wektor wag) oraz w_0 (odchylenie). Wektory danych przypisywane są do odpowiedniej klasy na podstawie znaku odpowiadających im wartości funkcji decyzyjnej. Pomimo swej prostoty, algorytmy klasyfikacji liniowej nadają się do rozwiązywania wielu, także skomplikowanych zadań eksploracji danych. Ponadto, dzięki istniejącym uogólnieniom funkcji liniowej, wykorzystywanym m.in. w algorytmie wektorów podpierających[1], możliwe jest zastosowanie tych samych metod również w przypadku klas nieliniowo separowanych.

W dalszym ciągu do rozwiązania poszczególnych zadań będzie wykorzystywany program Weka. Oprócz poznanego wcześniej modułu *Explorera*, wygodnym będzie użycie także narzędzia o nazwie *BoundaryVisualizer* (zob. pkt 3). Program ten pozwala na graficzne przedstawienie skonstruowanej granicy decyzyjnej.

Zbiory danych, których będą dotyczyły poszczególne ćwiczenia, pochodzą w części z repozytorium UCI, inne zaś zostały utworzone na podstawie obrazów tekstur naturalnych pochodzących z albumu Brodatza. Charakterystykę wybranych zbiorów zaprezentowano w pkt 2. Dodatkowo, jedno z zadań dotyczy zbioru danych syntetycznych, który nie posiada bezpośredniego odniesienia do rzeczywistego problemu klasyfikacji.

Oczekiwane efekty kształcenia:

→ Znajomość wybranych algorytmów klasyfikacji liniowej.

→ Rozumienie ogólnych zasad działania klasyfikatorów liniowych oraz pojęcia granicy decyzyjnej.

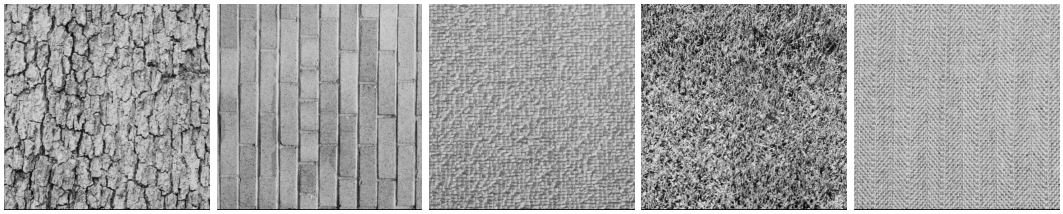
→ Umiejętność rozwiązywania nieliniowych problemów klasyfikacji przy użyciu algorytmu wektorów podpierających.

2 Zbiory danych

2.1 Tekstury naturalne

W ćwiczeniach wykorzystywane są dwa zbiory wektorów danych reprezentujące obrazy tekstur naturalnych z albumu Brodatza[2]. Pierwszy ze zbiorów został utworzony dla następujących obrazów (w nawiasach podano numer danego obrazu w albumie):

[1] *Support Vector Machines*
[2] http://sipi.usc.edu/database/database.cgi?volume=textures.



(a) (b) (c) (d) (e)

Rysunek 1: Obrazy tekstur naturalnych z albumu Brodatza

- *Bark* (D12 — rys. 1a),
- *Brick* wall (D94 — rys. 1b),
- *Raffia* (D84 — rys. 1c).

Drugi zbiór wykonano na podstawie obrazów:

- *Grass* (D9 — rys. 1d),
- *Herringbone weave* (D15 — rys. 1e).

W obrazach pierwszej grupy wyodrębniono po 128 mniejszych fragmentów w kształcie kwadratu o rozmiarach 20×20 piksele każdy. Dla każdego fragmentu wyznaczono zestaw cech tekstury za pomocą programu MaZda[3]. W programie tym, przy użyciu dostępnych w nim narzędzi do selekcji cech, wyznaczono podzbiór parametrów dających najlepsze rozróżnienie pomiędzy różnymi obrazami. Tak przygotowany zbiór danych wyeksportowano do formatu, dającego się odczytać przez program Weka.

Procedura utworzenia drugiego zbioru była podobna. Jednakże tu, zamiast kwadratu o boku 20 pikseli, użytu kwadratu o rozmiarach 16×16. Należy zaznaczyć, że zmniejszenie rozmiaru obszaru zainteresowania ma wpływ na dalsze przetwarzanie zbioru danych. Powoduje to bowiem ograniczenie ilości informacji o wzrocu tekstury zawartej w danym obrazie.

2.2 Australijski język migowy

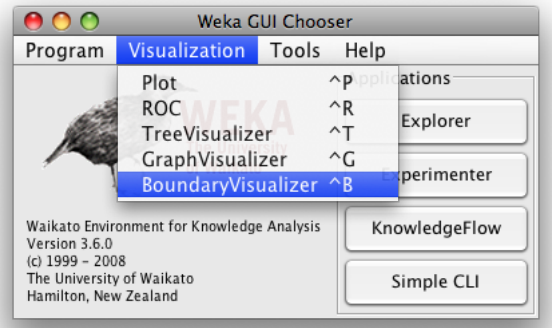
W repozyturium UCI znajduje się bardzo interesujący zbiór *Auslan*, w którym wektory danych odpowiadają wybranym znakom australijskiego języka migowego[4]. Cechy zastosowane tu do opisu poszczególnych znaków odpowiadają współrzędnym położenia dłoni oraz określonym ruchom palców w kolejnych chwilach wykonywania gestu.

Na potrzeby ćwiczenia laboratoryjnego wybrano do analizy tylko dwa spośród 95 znaków opisanych w zbiorze danych (zob.rys. 2. Obydwie klasy reprezentowane są ok. 150 przykładów podzielonych na część uczącą i testową.

[3] http://www.eletel.p.lodz.pl/mazda/.
[4] http://archive.ics.uci.edu/ml/datasets/Australian+Sign+Language+signs.

(*a*) (*b*)

Rysunek 2: Przykłady znaków w australijskim języku migowym: chłopiec (a) i dziewczynka (b)



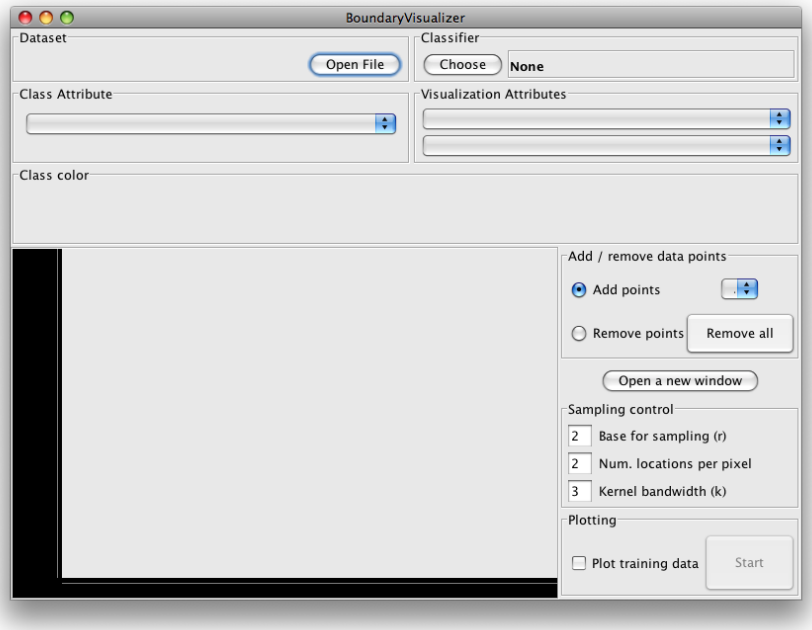
Rysunek 3: Główne okno programu *Weka GUI Chooser*

3 Narzędzie *BoundaryVisualizer*

3.1 Opis programu

Bardzo często oprócz ilościowej informacji o jakości wyuczonego klasyfikatora, istotne dla zrozumienia złożoności danego problemu może być graficzne przedstawienie skonstruowanej reguły decyzyjnej. Narzędzie *BoundaryVisualizer* służy właśnie do tego celu. Jego uruchomienie odbywa się poprzez wywołanie odpowiedniego polecenia z menu Visualize w głównym oknie programu Weka (rys. 3).

Po uruchomieniu programu należy wczytać plik zawierający dane (przycisk *Open file* — patrz rys. 4). Następnie należy wybrać algorytm klasyfikacji, przy pomocy którego mają zostać utworzone granice decyzyjne pomiędzy poszczególnymi klasami wczytanego zbioru wektorów danych (przycisk *Choose*). W dalszej kolejności należy wybrać dwuwymiarową przestrzeń cech (panel *Visulization Attributes*, w której prowadzone będą obliczenia. Do wczytanego zbioru można dodawać nowe lub usuwać istniejące punkty danych. W tym celu wystarczy naciskać dowolnym klawiszem myszy w obszarze wizualizacji znajdującym się po lewej stronie okna — rodzaj wykonanej akcji (dodanie/usunięcie wektora danych) zależy od zaznaczenia pola wyboru *Add points* albo *Remove points*.



Rysunek 4: Okno programu narzędziowego *BoundaryVisualizer*

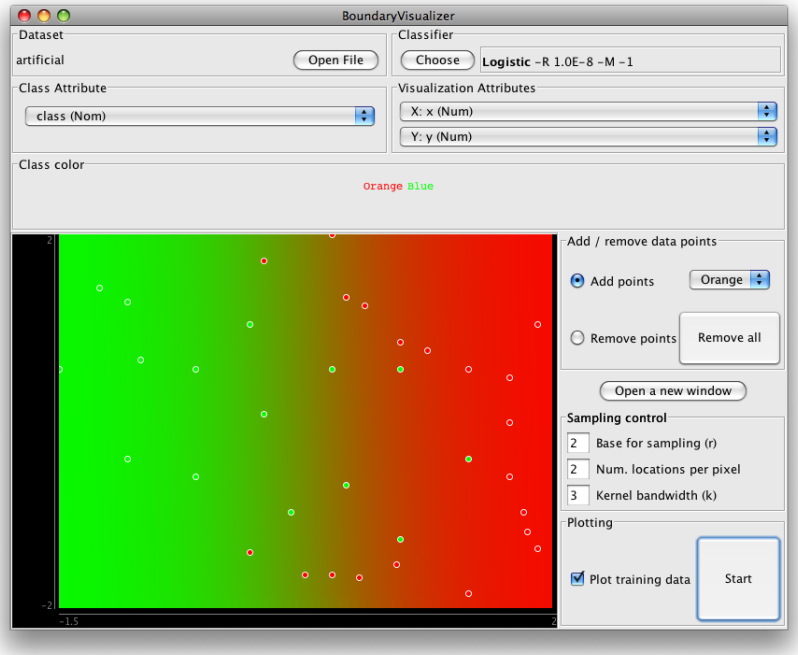
Wyznaczenie granic decyzyjnych nastąpi po naciśnięciu przycisku *Start*. Jeżeli wcześniej zaznaczono pole wyboru *Plot training data*, to po zakończeniu obliczeń, na wykonany rysunek obszarów decyzyjnych zostaną nałożone punkty danych.

3.2 Przykładowy wynik

Na rys. 5 zamieszczono wynik działania programu narzędziowego *BoundaryVisualizer* dla zbioru danych syntetycznych, o którym mowa była we wstępie. W przykładzie tym, granica decyzyjna została wyznaczona za pomocą algorytmu klasyfikacji liniowej, opartego na zasadzie regresji logistycznej. Jak widać, w tym przypadku klasyfikator liniowu nie był w stanie prawidłowo rozdzielić wektorów danych należących do przeciwnych klas.

4 Zadania do rozwiązania

Uwaga! O ile nie wskazano inaczej, zbiory danych, których dotyczą poniższe ćwiczenia umieszczone są na zdalnym serwerze. Jeśli w treści zadania mowa jest o wczytaniu określonego pliku, to należy dokonać tego poprzez naciśnięcie przycisku *Open URL* i wpisanie `http://eletel.p.lodz.pl/aklepaczko/open/nazwa_pliku`, przy czym ostatni człon ścieżki należy zastąpić nazwą odpowiedniego zbioru.



Rysunek 5: Przykładowy wynik działania programu *BoundaryVisualizer*

4.1 Problemy klasyfikacji liniowej

Zadanie 1

- Za pomocą przycisku *Open URL* wczytaj zbiór danych `BA_BR_RA_train.arff`.
- Przejdź do zakładki *Classify* i wybierz dowolny klasyfikator z grupy reguł lub drzew decyzyjnych poznany na poprzednich zajęciach. Wykonaj klasyfikację przy zaznaczonej opcji testowania metodą walidacji krzyżowej.
- Wybierz teraz klasyfikator *SVM* znajdujący się w grupie algorytmów funkcyjnych (*functions*). Klasyfikator ten występuje pod nazwą *SMO*, która odnosi się do jego szczególnej implementacji — *Sequential Minimal Optimization*. Przy domyślnych ustawieniach, algorytm ten realizuje funkcję liniową.
- Wykonaj ponownie klasyfikację i porównaj błąd klasyfikacji z wynikiem uzyskanym poprzednio. Który algorytm wydaje się lepszy?
- Przeprowadź klasyfikację jeszcze dwukrotnie. W pierwszym przypadku zaznacz opcję testowania na zbiorze treningowym. Za drugim razem, wybierz opcję testowania na odrębnym zbiorze (*Supplied test set*) i naciśnij przycisk *Set...* Wczytaj zbiór o nazwie `BA_BR_RA_test.arff`.

- Porównaj uzyskane wyniki. Co można powiedzieć o wiarygodności oszacowania rzeczywistego błędu klasyfikatora na podstawie testowania ze zbiorem treningowym oraz metodą walidacji krzyżowej?

Zadanie 2

- Uruchom narzędzie *BoundaryVisualizer*. Wczytaj plik *Iris.arff* znajdujący się w katalogu `C:\Program Files\Weka-3-6\Data`. Powiększ okno, jeżeli wczytany zbiór danych nie jest widoczny w całości.
- Za pomocą przycisku *Choose* wybierz klasyfikator oparty na regresji logistycznej — *SimpleLogistic* (także grupa *functions*).
- U dołu ekranu zaznacz opcję *Plot training data*, a następnie uruchom procedurę wyznacznania granic decyzyjnych między klasami za pomocą przycisku *Start*.
- Po zakończeniu procedury, naciśnij przycisk *Open a new Window*. Wybierz teraz klasyfikator *SMO* i naciśnij *Start*.
- Czy granice i obszary decyzyjne uzyskane w obu przypadkach pokrywają się? Który wynik wydaje się bardziej optymalny i dlaczego?

4.2 Problemy klasyfikacji nieliniowej

Zadanie 3

- Zamknij narzędzie *BoundaryVisualizer* i powróć do modułu *Explorera*. Wczytaj zbiór danych `auslan-bg-train.csv`, zawierający charakterystyki znaków australijskiego języka miganego.
- W zakładce *Classify* wybierz klasyfikator *SMO*. Zaznacz opcję *Supplied test set* i wskaż zbiór testowy znajdujący się w pliku `auslan-bg-test.csv`. Przeprowadź klasyfikację.
- W zakładce *Preprocess*, w panelu atrybutów zaznacz cechy o numerach 3, 22 i 23 (etykieta klasy). Za pomocą przycisku *Invert* odwróć zaznaczenia i naciśnij przycisk *Remove*. Po wykonaniu tej operacji wektory danych powinny składać się tylko z dwóch cech oraz etykiety kategorii. Przeprowadź klasyfikację za pomocą algorytmu SVM z użyciem metody 10-krotnej walidacji krzyżowej. Jaki jest błąd klasyfikacji?
- Przejdź do zakładki *Visualize* i przeanalizuj rozkład wektorów danych w zredukowanej przestrzeni cech. Czy jakikolwiek algorytm liniowy nadaje się do klasyfikacji tego zbioru?

- Powróć do zakładki *Classify* i otwórz okno dialogowe ustawień opcji algorytmu *SMO*. Naciśnij przycisk *Choose* służący tu do wyboru funkcji jądra i wybierz funkcję *RBFKernel*. Następnie wyedytuj opcje funkcji jądra. Zmień wartość współczynnika *gamma* na 1.5. Zamknij otwarte okna dialogowe potwierdzając dokanane zmiany. Wykonaj klasyfikację i porównaj otrzymane wyniki.

Zadanie 4

- Wczytaj plik `Sonar.arff`. Wykonaj klasyfikację tego zbioru za pomocą dowolnie wybranego drzewa lub reguły decyzyjnej.
- Ponownie wybierz klasyfikator *SMO* i wskaż wielomianową funkcję jądra. Przeprowadź klasyfikację dla tego algorytmu dwukrotnie przy następujących ustawieniach opcji funkcji jądra:
  - *exponent*: 1, *useLowerOrder*: *False*
  - *exponent*: 1, *useLowerOrder*: *True*

  W pierwszym przypadku realizowana jest funkja liniowa, w drugim zaś funkcja wielomianowa 3-ego stopnia. Czy w przypadku zbioru *Sonar*, zysk z użycia nieliniowej funkcji jądra jest równie istotny, jak miało to miejsce w przypadku zbioru *Auslan*?
- Sprawdź, czy ewentualne zwiększenie nieliniowości klasyfikatora (poprzez zwiększenie współczynnika wykładniczego) spowoduje zmniejszenie błędu klasyfikacji.

Zadanie 5

- Powtórz zadania z poprzedniego ćwiczenia dla zbioru tekstur `GR_HW_16.csv`.
- Czy nielinowa funkcja jądra pozwoliła zwiększyć dokładność klasyfikacji?

Zadanie 6

- Uruchom ponownie narzędzie *BoundaryVisualizer*. Wczytaj plik `artificial.arff` (z lokalizacji wskazanej przez prowadzącego zajęcia) zawierający wektory należące do hipotetycznego zbioru danych.
- Wybierz klasyfikator SVM z wielomianową funkcją jądra 2. stopnia (opcja *useLowerOrder* ustawiona na *True*). Zaznacz pole wyboru *Plot training data* i naciśnij *Start*.
- Czy po zakończeniu obliczeń wszystkie punkty danych znalazły się we właściwym obszarze? Powtórz operację dla 5. i 8. stopnia funkcji wielomianowej.

Łódź 2009

Zredagowane w LaTeX-u