Metoda najmniejszych kwadratów

Jan Sarba, Dariusz Rozmus

15.03.2025

1 Treść zadania

1.1 Zadanie 1

Celem zadania jest zastosowanie metody najmniejszych kwadratów do predykcji, czy nowotwór jest złośliwy (ang. *malignant*) czy łagodny (ang. *benign*). Nowotwory złośliwe i łagodne mają różne charakterystyki wzrostu. Istotne cechy to m. in. promień i tekstura. Charakterystyki te wyznaczane są poprzez diagnostykę obrazową i biopsje.
Do rozwiązania problemu wykorzystamy bibliotekę `pandas`, typ `DataFrame` oraz dwa zbiory danych:

- `breast-cancer-train.dat`
- `breast-cancer-validate.dat`.

Nazwy kolumn znajdują się w pliku `breast-cancer.labels`. Pierwsza kolumna to identyfikator pacjenta `patient ID`. Dla każdego pacjenta wartość w kolumnie `Malignant/Benign` wskazuje klasę, tj. czy jego nowotwór jest złośliwy czy łagodny. Pozostałe 30 kolumn zawiera cechy, tj. charakterystyki nowotworu.

a) Otwórz zbiory `breast-cancer-train.dat` i `breast-cancer-validate.dat` używając funkcji `pd.io.parsers.read_csv` z biblioteki `pandas`.

b) Stwórz histogram (z rozróżnieniem na typ nowotworu) i wykres wybranej kolumny danych przy pomocy funkcji `hist` oraz `plot`. W przypadku wykresu posortuj wartości kolumny od najmniejszej do największej. Pamiętaj o podpisaniu osi i wykresów.

c) Stwórz reprezentacje danych zawartych w obu zbiorach dla liniowej i kwadratowej metody najmniejszych kwadratów (łącznie 4 macierze). Dla reprezentacji kwadratowej użyj tylko podzbioru dostępnych danych, tj. danych z kolumn `radius (mean)`, `perimeter (mean)`, `area (mean)`, `symmetry (mean)`.

d) Stwórz wektor b dla obu zbiorów (tablicę numpy 1D-array o rozmiarze identycznym jak rozmiar kolumny `Malignant/Benign` odpowiedniego zbioru danych). Elementy wektora b to 1 jeśli nowotwór jest złośliwy, -1 w przeciwnym wypadku. Funkcja `np.where` umożliwi zwięzłe zakodowanie wektora b.

e) Znajdź wagi dla liniowej oraz kwadratowej reprezentacji najmniejszych kwadratów przy pomocy macierzy A zbudowanych na podstawie zbioru

`breast-cancer-train.dat`. Potrzebny będzie także wektor b zbudowany na podstawie zbioru `breast-cancer-train.dat`.

Uwaga. Problem najmniejszych kwadratów rozwiąż, stosując równanie normalne. Rozwiązując równanie normalne należy użyć funkcji `np.linalg.solve`, unikając obliczania odwrotności macierzy funkcją `scipy.linalg.pinv`.

f) Znajdź odrębny zbiór wag dla reprezentacji liniowej, używając funkcji `scipy.linalg.lstsq` (która stosuje rozkład SVD do rozwiązania problemu) oraz zbiór wag dla zregularyzowanej reprezentacji liniowej, rozwiązując równanie normalne oraz stosując $\lambda = 0.01$.

g) Oblicz współczynniki uwarunkowania macierzy, $(A^T A)$, dla liniowej i kwadratowej metody najmniejszych kwadratów, wyznaczone na zbiorze treningowym. Jaki wpływ ma współczynnik uwarunkowania na numeryczną interpretację wag?

h) Sprawdź jak dobrze otrzymane wagi przewidują typ nowotworu (łagodny czy złośliwy). W tym celu pomnóż liniową reprezentację zbioru `breast-cancer-validate.dat` oraz wyliczony wektor wag dla reprezentacji liniowej. Następnie powtórz odpowiednie mnożenie dla reprezentacji kwadratowej. Zarówno dla reprezentacji liniowej jak i kwadratowej otrzymamy wektor p. Zakładamy, że jeśli $p[i] > 0$, to i-ta osoba (prawdopodobnie) ma nowotwór złośliwy. Jeśli $p[i] \leq 0$ to i-ta osoba (prawdopodobnie) ma nowotwór łagodny.

Porównaj wektory p dla reprezentacji liniowej i kwadratowej z wektorem b (użyj reguł $p[i] > 0$ oraz $p[i] \leq 0$).

Dla wszystkich reprezentacji oblicz macierz pomyłek (ang. *confusion matrix*) oraz dokładność metody:

$acc = \frac{TP+TN}{TP+TN+FP+FN}$, gdzie
TP – liczba przypadków prawdziwie dodatnich
TN – liczba przypadków prawdziwie ujemnych
FP – liczba przypadków fałszywie dodatnich
FN – liczba przypadków fałszywie ujemnych.
Przypadek fałszywie dodatni zachodzi, kiedy model przewiduje nowotwór złośliwy, gdy w rzeczywistości nowotwór był łagodny. Przypadek fałszywie ujemny zachodzi, kiedy model przewiduje nowotwór łagodny, gdy w rzeczywistości nowotwór był złośliwy.

2 Argumentacja

a) **Wczytanie danych** Dane wejściowe są przechowywane w plikach CSV. Wczytujemy je za pomocą biblioteki `pandas` i przypisujemy odpowiednie nazwy kolumn.

```
import pandas as pd

train_data = pd.read_csv("breast-cancer-train.dat", delimiter=",")
validate_data = pd.read_csv("breast-cancer-validate.dat", delimiter=",")

with open("breast-cancer.labels") as f:
    column_names = f.read().splitlines()

train_data.columns = column_names
validate_data.columns = column_names
```

b) **Liczba cech w reprezentacjach**

- **Reprezentacja liniowa**: Wykorzystuje wszystkie 30 cech (pomijając kolumny `patient ID` i `Malignant/Benign`)
- **Reprezentacja kwadratowa**: Używa 4 wybranych cech (`radius (mean)`, `perimeter (mean)`, `area (mean)`, `symmetry (mean)`), ich kwadratów oraz interakcji:

$$4_{\text{cechy}} + 4_{\text{kwadraty}} + 6_{\text{interakcje}} = 14_{\text{cech}}$$

c) **Tworzenie macierzy cech**

```
# Wszystkie cechy (30) dla modelu liniowego
all_features = [col for col in column_names
                if col not in ["patient ID", "Malignant/Benign"]]
A_lin_train = train_data[all_features].values

# 4 wybrane cechy dla modelu kwadratowego
quad_features = ["radius (mean)", "perimeter (mean)",
                 "area (mean)", "symmetry (mean)"]
A_quad_base = train_data[quad_features].values

# Rozszerzenie kwadratowe
A_quad_train = np.hstack([
    A_quad_base,
    A_quad_base ** 2,  # 4 kwadraty
    # 6 interakcji
    A_quad_base[:,0:1] * A_quad_base[:,1:2],
    A_quad_base[:,0:1] * A_quad_base[:,2:3],
    A_quad_base[:,0:1] * A_quad_base[:,3:4],
    A_quad_base[:,1:2] * A_quad_base[:,2:3],
    A_quad_base[:,1:2] * A_quad_base[:,3:4],
    A_quad_base[:,2:3] * A_quad_base[:,3:4]
])
```

d) **Tworzenie wektora klasyfikacji**

```
b_train = np.where(train_data["Malignant/Benign"] == "M", 1, -1)
```

e) **Rozwiązanie równań normalnych**

```
# Dla modelu liniowego (30 cech)
w_lin = np.linalg.solve(A_lin_train.T @ A_lin_train,
                        A_lin_train.T @ b_train)

# Dla modelu kwadratowego (14 cech)
w_quad = np.linalg.solve(A_quad_train.T @ A_quad_train,
                         A_quad_train.T @ b_train)
```

f) **Metody alternatywne**

```
# SVD (dla modelu liniowego)
w_lin_svd, _, _, _ = lstsq(A_lin_train, b_train)

# Ridge Regression (lambda=0.01)
lambda_reg = 0.01
I = np.eye(A_lin_train.shape[1])
w_ridge = np.linalg.solve(A_lin_train.T @ A_lin_train +
                          lambda_reg * I,
                          A_lin_train.T @ b_train)
```

g) **Analiza uwarunkowania**

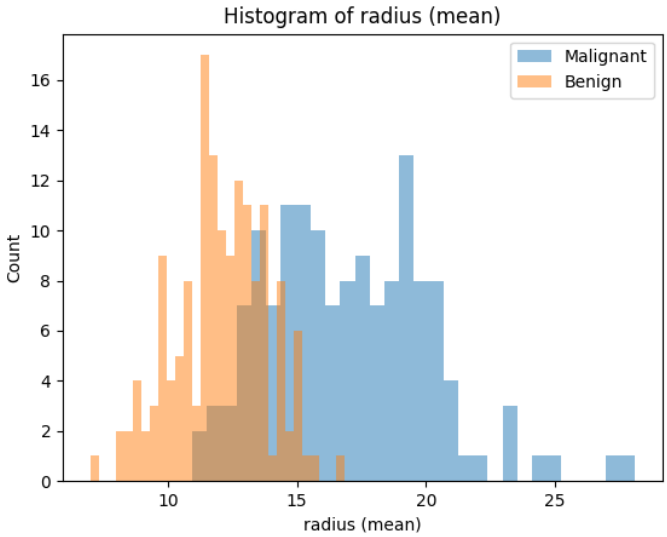
```
print(f"Liniowy: {np.linalg.cond(A_lin_train.T @ A_lin_train):.2e}")
print(f"Kwadratowy: {np.linalg.cond(A_quad_train.T @ A_quad_train):.2e}")
```

h) **Ewaluacja modeli**

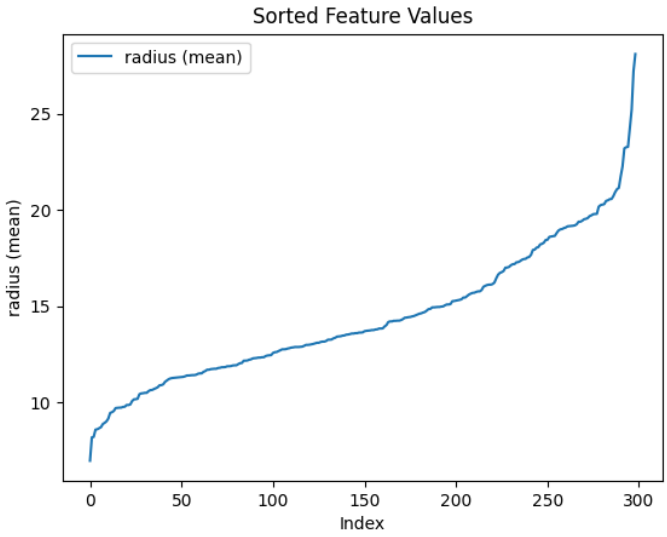
```
# Przyklad dla modelu liniowego
pred_lin = (A_lin_validate @ w_lin) > 0
print(confusion_matrix(b_validate, pred_lin))
print(f"Dokladnosc: {accuracy_score(b_validate, pred_lin):.4f}")
```

3 Wyniki

Oto wykresy z podpunktu *a*:



Rysunek 1: Rozkład średniego promienia



Rysunek 2: Wykres dla danych posortowanych

Współczynnik uwarunkowania macierzy ($A^T A$) wpływa na interpretację wag w następujący sposób:

- **Duży współczynnik uwarunkowania** oznacza, że macierz ($A^T A$) jest bliska osobliwości, co prowadzi do:
  - Niestabilności numerycznej w rozwiązaniach równań normalnych.
  - Dużej czułości wag na niewielkie zmiany w danych wejściowych.
  - Przesadnie dużych wartości niektórych wag, co może sugerować nadmierne dopasowanie do danych treningowych.
- **Mały współczynnik uwarunkowania** oznacza lepiej uwarunkowany problem, co skutkuje:
  - Stabilniejszymi i bardziej interpretowalnymi wagami.
  - Mniejszym ryzykiem nadmiernego dopasowania.

W przypadku metody kwadratowej współczynnik uwarunkowania jest zwykle większy niż dla metody liniowej, ponieważ dodanie cech kwadratowych i interakcyjnych zwiększa kolinearność danych. Dlatego regularyzacja (np. Ridge Regression) może pomóc w stabilizacji wag i poprawie generalizacji modelu.
Program zwraca następujące wyniki:

```
Condition number (linear): 824319661.4021404
Condition number (quadratic): 9.02853580416064e+17

Confusion Matrix (Linear):
[[192   8]
 [  7  52]]
Accuracy (Linear): 0.9421

Confusion Matrix (Quadratic):
[[185  15]
 [  5  54]]
Accuracy (Quadratic): 0.9228

Confusion Matrix (SVD - Linear):
[[192   8]
 [  7  52]]
Accuracy (SVD - Linear): 0.9421

Confusion Matrix (Ridge Regression - Linear):
[[192   8]
 [  7  52]]
Accuracy (Ridge Regression - Linear): 0.9421

Condition number (Ridge Regression - Linear): 788143664.6745715
```

4 Wnioski

Liczba uwarunkowania dla macierzy normalnej ($A^T A$)**:** Model liniowy ma bardzo wysoką liczbę uwarunkowania($\approx 8.24 \times 10^8$), co wskazuje na silną niestabilność obliczeniową. Model kwadratowy ma jeszcze gorszy współczynnik uwarunkowania ($\approx 9.03 \times 10^{17}$), co sugeruje ekstremalne problemy numeryczne. Ridge Regression znacząco poprawia współczynnik uwarunkowania ($\approx 7.88 \times 10^8$), choć ten wciąż pozostaje niepokojąco wysoki.
Dokładność modeli: Model liniowy, jego wersja SVD oraz Ridge Regression osiągają tę samą dokładność (94.21%), co sugeruje, że regularizacja Ridge nie pogarsza predykcji. Model kwadratowy jest mniej dokładny (92.28%) i dodatkowo bardziej niestabilny.
Macierz pomyłek: Model kwadratowy ma więcej fałszywie pozytywnych wyników (15 vs. 8 w modelu liniowym), co wskazuje na większą tendencję do błędnych klasyfikacji zdrowych próbek jako chorych.
Wniosek: Model liniowy wydaje się najlepszym wyborem – jest stabilny (zwłaszcza po SVD/Ridge), osiąga najwyższą dokładność i ma lepszy balans błędów niż model kwadratowy.

5 Bibliografia

- **lstsq - dokumentacja**
- **Wykład 2 - interpolacja (MOWNiT)**