WSI - Regresja i klasyfikacja

Marcin Szymczak 314910

1 Opis badanej metody

SVM jest metoda klasyfikacji, która polega na znalezieniu hiperpłaszczyzny odzielającej klasy marignesem. Pseudokod dla przypadku liniowo separowalnego:

- Powtarzaj przez zadaną maksymalną liczbę iteracji:
  - Dla każdej próbki treningowej:
    - Oblicz warunek marginesu: $y_i(w^T x_i + b)$.
    - Jeśli próbka nie narusza marginesu:
      - Zaktualizuj wagi, minimalizując wpływ regularyzacji.
    - Jeśli próbka narusza margines:
      - Zaktualizuj wagi, biorąc pod uwagę wpływ tej próbki na granicę decyzyjną.
      - Zaktualizuj przesunięcie marginesu.

Natomiast gdy przypadek nie jest liniowo separowalny należy zastosować SVM z funkcjami jądrowymi. Dzięki funkcją jądrowym, dane są przekształcane do wyższej wymiarowości, gdzie mogą być liniowo separowalne. Pseudokod:

- Powtarzaj przez zadaną maksymalną liczbę iteracji:
  - Dla każdej próbki treningowej:
    - Oblicz wartość funkcji decyzyjnej z użyciem funkcji jądrowej: $f(x_i) = \sum_{j=1}^{N} \alpha_j y_j k(x_j, x_i) + b$, gdzie $k(x_j, x_i)$ to funkcja jądrowa.
    - Oblicz warunek marginesu: $y_i f(x_i)$.
    - Jeśli próbka nie narusza marginesu:
      - Nie aktualizuj mnożników α ani przesunięcia b.
    - Jeśli próbka narusza margines:
      - Zaktualizuj mnożnik Lagrange'a α_i, uwzględniając błędy klasyfikacji i regularyzację.
      - Zaktualizuj przesunięcie marginesu b.
- **Liniowe**: $k(u, v) = u^T v$

- **Wielomianowe**: $k(u, v) = (1 + u^T v)^d,\ d > 0$
- **Gaussowskie/Radialne (RBF)**: $k(u, v) = \exp\left(-\frac{\|u-v\|^2}{2\sigma^2}\right)$

2 Planowane eksperymenty numeryczne

2.1 Cel eksperymentów

Celem jest zbadanie wpływu wybranych hiperparametrów na dokładność klasyfikacji modeli SVM.

1. Klasyczny SVM - badanie wpływu zmiany hiperparametrów.
2. SVM z funkcjami jądrowymi (RBF i wielomianowe, liniowe) - sprawdzenie jedynie jednej konfiguracji, ponieważ zaimplementowane przezemnie modele trenują się bardzooo długo.

3 Warunki eksperymentów numerycznych

3.1 Data set

- Zbiór danych: Wine Quality Data set (red wine),
- Zmienna objaśniająca - dodanie nowej kolumny "quality_binary" na podstawie kolumny "quality", gdzie wartości większe bądź równe 6 dają wartość 1, a pozostałe -1.
- Podział danych - 80% dane treningowe, 20% dane testowe

3.2 Eksperymenty dla klasycznego SVM

Hiperparametry:

- learning rate: {0.0001, 0.001, 0.1},
- Liczba iteracji : {10, 100, 1000},
- Parametr regularyzacji (λ): {0.01, 0.1, 1}.
- Porównanie dokładności.

3.3 Eksperymenty dla SVM z różnymi funkcjami jądrowymi

- Porównanie dokładności.

4 Otrzymane Wyniki

Learning Rate	Accuracy (%)
0.0001	44.06
0.001	44.06
0.1	55.94

Tabela 1: Wpływ wartości learning rate na dokładność.

Iterations	Accuracy (%)
10	55.62
100	44.06
1000	44.06

Tabela 2: Wpływ liczby iteracji na dokładność.

Lambda	Accuracy (%)
0.01	44.06
0.1	44.06
1	55.94

Tabela 3: Wpływ wartości parametru lambda na dokładność.

Kernel	Accuracy (%)
Liniowy	44.06
RBF	63.75

Tabela 4: Dokładność modeli SVM dla różnych jąder

Kernel	Accuracy (%)
Liniowy	73.12
RBF	68.44

Tabela 5: Dokładność modeli SVM dla różnych jąder w scikit-learn.

5 Wnioski

- Wpływ learning rate - większy learning rate spowodował poprawę dokładności, natomiast jeszcze większy ten współczynnik mógłby spowodować niestabilność modelu.

- Liczba iteracji - Zwiększenie iteracji nie poprawiło dokładności a nawet przyniosło odwrotny efekt.
- Parametr regularyzacji - zwiększenie go dało zwiększenie dokładności.
- Z podpunktów powyżej można wywnioskować, że data set zastosowany do treningu nie jest liniowo separowalny.
- W data set jak zmieniłem warunek "quality_binary" na większe równe 7 to nieważne jakie parametry miał model, zawsze dawał dokładność 85,31%
- Zaimplementowane przeze mnie modele z funkcjami jądrowymi trenują się bardzo długi czas przez co nie byłem w stanie ich zbadać dla sensownej ilości iteracji. Dlatego posłużyłem się gotową implementacją z scikit-learn do sprawdzenia jaką dokładność rzeczywiście jesteśmy w stanie otrzymać dla takich SVM dla takiego data setu.
- Nawet dla małej liczby iteracji SVM z RBF dał dużo lepszą dokładność od zwykłego SVM.
- Wyniki są dużo lepsze co daje wniosek taki że przygotowany data set jest liniowo nieseparowalny.