Wydział Elektroniki i Technik Informacyjnych

Algorytmy i Metody Optymalizacji

Projekt 2

Optymalizacja z ograniczeniami

zestaw nr 16

Wykonał:

Paweł Gajewski, 269823

1. Cel projektu

Celem projektu jest rozwiązać zadanie znalezienia najlepszej płaszczyzny rozdzielającej zbiór danych, poprzez rozwiązanie zadania prymalnego oraz zadania do niego dualnego.

Aby zrealizować projekt, trzeba zapoznać się z zestawem narzędzi OPTIMALIZATION, programu Matlab oraz wybrać odpowiednią funkcję.

2. Przygotowania

a. Maszyna wektorów nośnych

Maszyna wektorów nośnych (ang. SVM - support vector machine) jest to abstrakcyjny koncept maszyny, która działa jak klasyfikator, a której nauka ma na celu wyznaczenie hiperpłaszczyzny rozdzielającej z maksymalnym marginesem przykłady należące do dwóch klas. Często wykorzystywany jest do wyznaczania trasy w robotyce moblinej (w tym, analizie obrazów i danych z sensorów).

Rozważamy uczący zbiór danych o *N* obserwacjach postaci $\{(x_1, y_2), ..., (x_N, y_N)\}$. Każdy punkt x_i jest *D*-wymiarowym wektorem. Zakładamy, ze $y_i \in \{-1, 1\}$. Zadaniem prymalnym nazywamy:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \xi_i$$

przy ograniczeniach:

$$\xi_i \geq 1 - y_i[w^T x_i + b]$$
$$\xi_i \geq 0$$

gdzie

- w - D-wymiarowy wektor wag
- b - przesuniecie
- ξ - N-wymiarowy margines błędu w przyporządkowaniu obserwacji do niewłaściwych klas, zdefiniowanych przez hiperpłaszczyznę wyznaczana przez w i b
- C - parametr minimalizujący margines błędu

Rozważamy zatem zadanie programowania kwadratowego o N + 1 + D niewiadomych oraz 2N ograniczeniach. Zadaniem dualnym do powyższego jest:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \left(x_i^T x_j\right)$$

przy ograniczeniach:

$$0 \leq \alpha_i \leq C$$
$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

gdzie

- α - N-wymiarowy wektor współczynników Lagrange'a

Problem ten także jest zadaniem programowania kwadratowego, w którym wyznaczamy *N* niewiadomych przy *N* + 1 ograniczeniach.

b. Wybór solvera

Problem maszyny wektorów nośnych jest zadaniem programowania kwadratowego, więc do obliczeń wybrano funkcję *quadprog* z algorytmem *interior-point-convex*. Funkcja ta, w zależności od

klasy obiektów matematycznych podawanych jako argumenty, to znaczy macierzy klasycznych lub macierzy rzadkich, automatycznie dobiera rodzaj solvera liniowego:

- Dla macierzy klasycznych – *dense*
- Dla macierzy rzadkich - *sparse*

3. Wyniki

a. Dane losowe

Rozwiązania zadania prymalnego i dualnego testowane było wielokrotnie, dla zbiorów danych losowych o różnych rozmiarach. Zauważono, że solver *sparse* nie radzi sobie ze zbiorami danych o dużym rozmiarze, ale nie będącymi macierzami rzadkimi (widać to zwłaszcza w zadaniu dualnym), więc obliczenia wykonano również dla danych w formie klasycznych macierzy programu Matlab.

Obliczenia powtórzono wielokrotnie. Otrzymane wyniki uśredniono i przedstawiono w poniższej tabeli. Pomiary dokonywane były na komputerze o przeciętnej konfiguracji sprzętowej, jednak jeżeli obliczenia trwały dłużej niż 2 godziny były przerywane.

	Zadanie prymalne			Zadanie dualne		
Rozmiar zbioru	Dokładność [%]	liczba iteracji	czas	Dokładność [%]	liczba iteracji	czas
100	100	19	0.099499	100	19	0.096941
500	99.33	41	6.171885	99.33	36	0.450702
1000	99.33	63	58.624240	100	62	3.921787
5000	-	-	>7200	99.8	132	432.4722
10000	-	-	>7200	99.97	173	3781.4743

Tabela 2. Wyniki obliczeń dla danych losowych dla solvera *dense*

	Zadanie prymalne			Zadanie dualne		
Rozmiar zbioru	Dokładność [%]	liczba iteracji	czas	Dokładność [%]	liczba iteracji	czas
100	96.67	14	0.036254	100	14	0.104275
500	100	19	0.191888	100	14	0.965239
1000	99.33	23	0.664463	99.33	18	4.703269
5000	99.93	27	2.718893	99.94	32	559.924326
10000	99.96	32	8.470985	99.96	39	5755.458404

Tabela 2. Wyniki obliczeń dla danych losowych dla solvera *sparse*

Wnioski:

- Wszystkie rozwiązania mają podobną dokładność,
- Ilość iteracji wykonywanych przez solver *sparse* była mniejsza niż w przypadku solvera *dense*,
- Zadanie dualne szybciej rozwiązywał solver *dense*, natomiast prymalne – *sparse*.
- Dla solvera dense:
    - Zadanie dualne liczone jest o wiele szybciej, niż zadanie prymalne. Dla dwóch największych zbiorów, rozwiązanie zadania prymalnego trwało więcej niż dwie godziny.
- Dla solvera sparse:
    - Zadanie prymalne liczone jest o wiele szybciej, niż zadanie dualne,

b. Dane rzeczywiste (spambase.dat)

Dane rzeczywiste do testów SVM wzięto z pliku *spambase.dat*. Zawiera on 7000 tysięcy rekordów, każdy będący wektorem o długości 58, więc o wiele dłuższy niż w danych losowych. Każdy z nich jednak posiada znaczącą ilość zer.

Zgodnie z przepuszczeniami, użycie solvera *dense* nie pozwoliło otrzymać rozwiązania, zarówno zadania prymalnego, jak i dualnego (obliczenia przerywano po trzech godzinach). Natomiast solver *sparse*, przez dużą ilość zer obecnych w zbiorze danych, poradził sobie lepiej niż dla danych losowych.

Rozwiązanie prymalne zostało znalezione w czasie 1.187 sekundy, przy użyciu 9 iteracji. Jednak skuteczność rozwiązania jest niepokojąco mała: tylko 13.189%. Rozwiązanie dualne natomiast wyznaczone zostało już w fazie *presolve* (0 iteracji), co trwało 0.404 sekundy, a skuteczność rozwiązania wynosiła już 99.967%.

4. Wnioski

- Nie ma jednoznacznej zasady, kiedy warto wykorzystywać zadanie prymalne, a kiedy dualne. Wszystko zależy od cech charakterystycznych danych (rozmiaru danych, ich „rzadkości", ilości próbek), na których się pracuje oraz dostępnego narzędzia.
- Macierze rzadkie w programie Matlab mogą być bardzo przydatne w rozwiązywaniu problemów pewnych klas na sprzęcie z małą ilością pamięci.

5. Bibliografia

- https://en.wikipedia.org/wiki/Support-vector_machine
- https://www.mathworks.com/help/optim/ug/quadprog.html
- https://sci2s.ugr.es/keel/dataset.php?cod=102
- https://www.mathworks.com/help/optim/ug/large-sparse-quadratic-program-with-interior-point-algorithm.html