

Algorytmy i metody optymalizacji - projekt 2-7

Bartosz Goławski

Zadanie 1: Zapoznać się z kwadratowym zadaniem optymalizacji w wersji primalnej i dualnej dla zadania SVM dostępnych pod tym samym adresem

W zadaniu klasyfikacji SVM poszukujemy optymalnej hiperpłaszczyzny decyzyjnej $w^T x - b = 0$, gdzie wektor w oraz parametr b są współczynnikami hiperpłaszczyzny. Klasa każdej próbki jest obliczana jako $y_i = \text{sgn}(w^T x_i - b)$, gdzie i - indeks próbki. Gdy danych nie da się łatwo odseparować liniowo konieczne jest wprowadzenie miękkiego marginesu λ - im większy margines, tym większa jest dopuszczalność błędnie zakwalifikowanych próbek.

Zadanie primalne można zdefiniować jako minimalizacja poniższej funkcji celu (liczba próbek = n):

$$\text{Min } \frac{1}{n} \sum_{i=1}^n \zeta_i + \lambda \|w\|^2$$

Przy ograniczeniach:

$$\begin{aligned} y_i (w^T x_i - b) &\geq 1 - \zeta_i \text{ dla } i \in \{1, 2, \dots, n\} \\ \zeta_i &\geq 0 \text{ dla } i \in \{1, 2, \dots, n\} \end{aligned}$$

Zadanie dualne można natomiast zdefiniować jako maksymalizacja poniższej funkcji celu:

$$\text{Max } \sum_{i=1}^n c_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i c_i (x_i^T x_j) y_j c_j$$

Przy ograniczeniach:

$$\begin{aligned} \sum_{i=1}^n c_i y_i &= 0 \text{ dla } i \in \{1, 2, \dots, n\} \\ c_i &\geq 0 \wedge c_i \leq \frac{1}{2n\lambda} \text{ dla } i \in \{1, 2, \dots, n\} \end{aligned}$$

Z rozwiązania zadania dualnego można wyliczyć rozwiązanie zadania primalnego za pomocą następujących wzorów:

$$\begin{aligned} w &= \sum_{i=1}^n c_i y_i x_i, \\ b &= w^T x_i - y_i, \text{ gdzie } i \text{ jest takim indeksem, dla którego } 0 < c_i < \frac{1}{2n\lambda}. \end{aligned}$$

Zadanie 2: Wygenerować losowo dwa zbiory punktów w przestrzeni 5-wymiarowej, które leżą po dwóch stronach 4-wymiarowej hiperpłaszczyzny liniowej zawartej w tej przestrzeni (jeden po jednej, drugi po drugiej stronie)

Można przyjąć następujący sposób rozwiązania:

- Wygeneruj losowo wektor kolumnowy w oraz parametr b hiperpłaszczyzny.
- Wygeneruj zbiór losowych punktów mających leżeć "nad" oraz "pod" hiperpłaszczyzną (znak klasyfikacji hiperpłaszczyzny jest odpowiednio dodatni bądź ujemny).
- Przejdź iteracyjnie po wszystkich punktach obu zbiorów - jeżeli punkt w zbiorze dodatnich leży pod hiperpłaszczyzną bądź punkt w zbiorze ujemnych leży pod hiperpłaszczyzną, to odbij go symetrycznie względem hiperpłaszczyzny na drugą stronę.

Po wykonaniu powyższego algorytmu jest pewność, że zbiory są odpowiednio odseparowane przez hiperpłaszczyznę. Odbicie symetryczne x_i^S punktu x_i można uzyskać dzięki następującemu wzorowi:

$$x_i^S = x_i - \frac{2w(w^T x_i - b)}{\|w\|^2} \text{ dla } i \in \{1, 2, \dots, n\}$$

Zadanie 3: Wybrać odpowiednią funkcję ze skrzynki narzędziowej OPTIMIZATION i rozwiązać zadanie znalezienia najlepszej płaszczyzny rozdzielającej rozwiązując: a) zadanie prymalne, b) zadanie dualne do prymalnego.

Oba problemy można rozwiązać w oparciu o generator problemów *optimproblem*. Przekazujemy do niego funkcję celu (odpowiednio sparametryzowaną o opcje maksymalizacji lub minimalizacji) oraz listę ograniczeń zdefiniowanych za pomocą polecenie *optimconstr*. Zmienne decyzyjne do ograniczeń oraz funkcji celu tworzymy za pomocą polecenia *optimvar*. Zadanie rozwiązujemy w oparciu o polecenie *solve*, do którego można przekazać dodatkowe opcje jak na przykład własne wyświetlanie stanu optymalizacji.

Opis plików projektu:

- solution_random_points.m - skrypt rozwiązujący zadanie SVM w obu postaciach na losowo wygenerowanych punktach (zadanie 2)
- solution_monk.m - skrypt rozwiązujący zadanie SVM dla zbioru monk-2.dat (zadanie 4)
- performSVM.m - funkcja uruchamiająca rozwiązanie zadania dualnego oraz prymalnego oraz wyświetlająca wyniki każdego eksperymentu
- primalProblem.m - funkcja rozwiązująca zadanie prymalne - zwraca ona wynik klasyfikacji (liczbę próbek sklasyfikowanych poprawnie/błędnie) oraz parametry znalezionej płaszczyzny

- dualProblem.m - funkcja analogiczna do primalProblem.m, ale rozwiązująca zadanie dualne
- dispSolution.m - funkcja wyświetlająca wynik rozwiązanego zadania SVM

Eksperymenty przeprowadzono dla różnych wartości parametru λ , wyniki eksperymentu (wektor w oraz parametr b hiperpłaszczyzny znormalizowano względem w , żeby dało się porównywać wyniki w łatwy sposób):

zadanie	λ	Liczba poprawnych klasyfikacji	Liczba błędnych klasyfikacji	Znormalizowane w	Znormalizowane b
prymalne	1	93	7	[0.4630, 0.4880, 0.2453, 0.1599, 0.6795]	0.5001
	0.1	96	4	[0.4728, 0.4649, 0.1589, 0.1804, 0.7088]	0.3054
	0.01	97	3	[0.4704, 0.4829, 0.1863, 0.2863, 0.6549]	0.3395
	0.001	100	0	[0.3818, 0.4733, 0.1860, 0.3416, 0.6921]	0.2413
dualne	1	94	6	[0.4630, 0.4880, 0.2453, 0.1599, 0.6795]	0.4255
	0.1	96	4	[0.4728, 0.4649, 0.1589, 0.1804, 0.7088]	0.3055
	0.01	97	3	[0.4704, 0.4829, 0.1863, 0.2863, 0.6549]	0.3395

	0.001	100	0	[0.3818, 0.4733, 0.1860 0.3416, 0.6921]	0.2413
--	-------	-----	---	---	--------

Dane w tym zadaniu są odseparowane liniowo, więc można stosować bardzo wąskie marginesy - im węższy margines, tym dokładniejsza jest klasyfikacja. Oba zadania dają te same rozwiązania poza przypadkiem, gdy $\lambda = 0.001$ - wtedy rozwiązanie różni się delikatnie, ale mimo to klasyfikacja jest tak samo skuteczna.

Zadanie 4: Ściągnąć standardowe dane testowe dostępne pod adresem: <http://sci2s.ugr.es/keel/category.php?cat=clas#sub2> Proszę wybrać dane o nazwie: monk-2

Zbiór ten zawiera 432 próbki o sześciu atrybutach. Każda z próbek przypisana jest do klasy 0 lub 1. Wartości każdego z atrybutów są liczbami, więc w celu uruchomienia klasyfikacji SVM wystarczyło zamienić klasy z 0 na -1 (ponieważ klasyfikator w tym przypadku rozpoznaje znak klasyfikowanej próbki).

Zadanie 5: Proszę powtórzyć obliczenia z punktu 3 dla danych z punktu 4

Tabela wyników eksperymentu:

zadanie	λ	Liczba poprawnych klasyfikacji	Liczba błędnych klasyfikacji	Znormalizowane w	Znormalizowane b
prymalne	1	228	204	[-0.0001, -0.7061, -0.0000 -0.0520, -0.7061, -0.0000]	-5.1902
	0.1	348	84	[0.0006, -0.7724, 0.0008 -0.0328, -0.6342, 0.0008]	-3.2462
	0.01	348	84	[0.0000, -0.8944,	-2.9069

				0.0000 -0.0000, -0.4472, 0.0000]	
	0.001	348	84	[0.0000, -0.8944, 0.0000 -0.0000, -0.4472, 0.0000]	-2.9069
dualne	1	228	204	[-0.0001, -0.7055, -0.0000 -0.0660, -0.7056, -0.0000]	-5.2371
	0.1	348	84	[-0.0000, -0.7641, -0.0000 -0.0367, -0.6440, -0.0000]	-3.2552
	0.01	348	84	[0.0000, -0.8944, 0.0000 -0.0000, -0.4472, 0.0000]	-2.9069
	0.001	348	84	[0.0000, -0.8944, 0.0000 -0.0000, -0.4472, 0.0000]	-2.9069

Tym razem dane nie są liniowo separowalne, co od razu widać po wynikach eksperymentów - udało się uzyskać skuteczność klasyfikacji na poziomie 81% (348/432 poprawnych klasyfikacji). Widać również, że tym razem wyniki zadania prymalnego oraz dualnego różnią się dla wszystkich szerokości marginesu. Testowałem również bardzo szerokie marginesy ($\lambda > 10$ oraz $\lambda < 10^{-8}$), ale nie udało się uzyskać lepszej jakości klasyfikacji niż dla $\lambda = 0.1$.

Zadanie 6. Proszę porównać wyniki uzyskane różnymi metodami i sformułować wnioski.

Klasyczne liniowe zadanie klasyfikacji przy użyciu SVM jest zdecydowanie skuteczniejsze dla liniowo odseparowanych zbiorów danych. W przypadku zbiorów nie dających się

odseparować liniowo można użyć rozszerzeń tego zadania, na przykład można wykorzystać kernel trick. W przypadku danych ze zbioru monk-2 wyniki znajdowania parametrów hiperpłaszczyzny różniły się, ale dla każdej szerokości marginesu klasyfikator działał tak samo skutecznie porównując zadanie prymalne oraz dualne.